

Planiranje robotskog djelovanja zasnovano na tumačenju prostornih struktura

Švaco, Marko

Doctoral thesis / Disertacija

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:161283>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-01-31**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)





Sveučilište u Zagrebu

Fakultet strojarstva i brodogradnje

Marko Švaco

**PLANIRANJE ROBOTSKOG
DJELOVANJA ZASNOVANO NA
TUMAČENJU PROSTORNIH
STRUKTURA**

DOKTORSKI RAD

Zagreb, 2015.



University of Zagreb

Faculty of Mechanical Engineering and Naval Architecture

Marko Švaco

**ROBOT BEHAVIOUR PLANNING BASED
ON INTERPRETATION OF SPATIAL
STRUCTURES**

DOCTORAL THESIS

Zagreb, 2015.



Sveučilište u Zagrebu

Fakultet strojarstva i brodogradnje

Marko Švaco

**PLANIRANJE ROBOTSKOG
DJELOVANJA ZASNOVANO NA
TUMAČENJU PROSTORNIH
STRUKTURA**

DOKTORSKI RAD

Mentor: prof. dr. sc. Bojan Jerbić

Zagreb, 2015.



University of Zagreb

Faculty of Mechanical Engineering and Naval Architecture

Marko Švaco

**ROBOT BEHAVIOUR PLANNING BASED
ON INTERPRETATION OF SPATIAL
STRUCTURES**

DOCTORAL THESIS

Supervisor: Bojan Jerbić, PhD Full professor

Zagreb, 2015.

Podaci za bibliografsku karticu

| | |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UDK: | 004.89:681.5 |
| Ključne riječi: | Robotika, autonomni sustavi, teorija adaptivne rezonancije, genetski algoritmi |
| Znanstveno područje: | Tehničke znanosti |
| Znanstveno polje: | Strojarstvo |
| Institucija: | Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje |
| Mentor: | prof. dr. sc. Bojan Jerbić |
| Broj stranica: | 116 |
| Broj slika: | 65 |
| Broj tablica: | 19 |
| Broj korištenih bibliografskih jedinica: | 106 |
| Datum obrane: | 30.4.2015. |
| Povjerenstvo: | prof. dr. sc. Dubravko Majetić, predsjednik (Fakultet strojarstva i brodogradnje, Zagreb) prof. dr. sc. Bojan Jerbić, mentor (Fakultet strojarstva i brodogradnje, Zagreb) prof. dr. sc. Stjepan Bogdan, član (Fakultet elektrotehnike i računarstva, Zagreb) |
| Institucija u kojoj je rad pohranjen: | Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje Nacionalna i sveučilišna knjižnica u Zagrebu |

Podaci o mentoru

Bojan Jerbić rođen je 13. rujna 1957. godine u Zagrebu. Osnovnu školu i Matematičku gimnaziju pohađao je u Zagrebu. Na Fakultet strojarstva i brodogradnje, Sveučilišta u Zagrebu upisao se 1976 godine. Diplomirao je 25. veljače 1983. godine diplomskim radom: "Kinematička struktura Stanford manipulatora".

Poslijediplomski studij pohađao je u razdoblju od 1984. do 1987. godine na FSB-u, smjer Tehnologija u strojarskoj proizvodnji, Projektiranje proizvodnih procesa. Magistarski rad pod naslovom "Istraživanje optimalnog redoslijeda sredstava za proizvodnju", obranio je 1987. godine te stekao naslov magistra tehničkih znanosti.

Kao stipendista Florida State University boravi tijekom 1989. godine šest mjeseci na Department of Industrial Engineering, provodeći istraživanja u okviru izrade disertacije i znanstveno-stručnog usavršavanja. Disertaciju pod naslovom: "Interpretacija geometrije CAD modela u projektiranju automatske montaže ekspertnim sustavom", obranio je 1993. godine na Fakultetu strojarstva i brodogradnje, Sveučilišta u Zagrebu, te stekao naslov doktora tehničkih znanosti iz znanstvenog područja Strojlarstvo.

Od 1984. godine radi na Katedri za projektiranje proizvodnih procesa, Zavoda za tehnologiju, FSB-a, prvo kao pripravnik, a zatim kao stručni suradnik sve do 1986. godine, kada je izabran za asistenta. U zvanje znanstvenog asistenta izabran je 1988. godine. Zvanje docenta stekao je 1995. godine, izvanrednog profesora 2000. godine, a zvanje redovitog profesora 2005. godine. Od 1993. do 2006. godine voditelj je Laboratorija za projektiranje izradbenih i montažnih sustava. Dužnost predstojnika Zavoda za robotiku i automatizaciju proizvodnih sustava obnašao je od 2005. do 2008. godine. Na mjestu voditelja Katedre za projektiranje izradbenih i montažnih sustava je od 2007. godine do danas.

Sudjeluje u izvođenju nastave iz predmeta: Automati za montažu, Projektiranje automatskih montažnih sustava, Umjetna inteligencija, Računalne mreže, Virtualno oblikovanje mehatroničkih sustava, Istodobno inženjerstvo, Programiranje automata za montažu, Vizijski sustavi, Inteligentni montažni sustavi i Integrirano inženjerstvo. Uveo je u nastavu brojne nove sadržaje, od primjene inženjerskih računalnih metoda u projektiranju do primijenjene robotike i umjetne inteligencije.

U znanstvenom radu posvetio se projektiranju izradbenih i montažnih sustava, poglavito razvoju računalnih metoda u projektiranju i metoda umjetne inteligencije u robotskoj montaži. U organiziranom znanstvenoistraživačkom radu sudjeluje od 1987. godine i autor je više od stotinu znanstvenih i stručnih radova. Bio je suradnik na sedam znanstvenih projekata te voditelj osam domaćih i međunarodnih znanstvenih i tehnologijskih projekata.

Osim u nastavnim i istraživačkim, aktivno sudjeluje u ostalim aktivnostima Fakulteta doprinoseći ukupnom razvoju znanstveno nastavnih djelatnosti. Od 1990. do 2006. godine član je Odbora za informatizaciju i kompjutorizaciju Fakulteta, a od 1997. godine djelovođa i potom član Povjerenstva za diplomske ispite Proizvodnog smjera, sve do 2010. Od 2009. do 2010. godine član je Odbora za poslijediplomske studije. Funkciju predsjednika Odbora za strategiju Fakulteta obnaša od 2010. do 2012. godine.

Tijekom svoje karijere kontinuirano je radio na popularizaciji struke putem brojnih javnih nastupa u medijima te objavljujući mnoge popularizacijske članke. Održao je 28 pozvanih predavanja u zemlji i inozemstvu.

Zahvale

Zahvaljujem mentoru prof. dr. sc. Bojanu Jerbiću prije svega na tome što mi je omogućio dolazak na Fakultet te na pruženoj podršci, velikoj upornosti i vjerovanju u moje sposobnosti, te brojnim savjetima dobivenim tijekom izrade ovog rada.

Članovima komisije prof. dr. sc. Dubravku Majetiću i prof. dr. sc. Stjepanu Bogdanu na korisnim savjetima i kritikama te pruženoj pomoći tijekom izrade ovog rada.

Prof. dr. sc. Zoranu Kunici, dr. sc. Petru Ćurkoviću, dr. sc. Tomislavu Stipančiću i Denisu Bašiću na potpori i savjetima.

Svojoj obitelji želim zahvaliti na dugogodišnjoj podršci.

Velika hvala prijateljima Bojanu i Filipu što su mi pomagali uvijek kada je bilo potrebno.

Zahvaljujem se svojoj supruzi Danijeli, koja je uvijek bila puna razumijevanja i strpljenja te mi pružila neizmjernu podršku. Jedna velika hvala i mom sinu Noi što me uvijek uspijevao nasmijati te što je bio moja inspiracija.

Zagreb, travanj 2015.

Marko Švaco

Sažetak

Robot je programabilan mehanizam čije se djelovanje temelji na upravljačkim algoritmima. Prilikom rada u nestrukturiranoj okolini upravljački algoritmi postaju eksplicitne funkcije položaja i vremena u povratnoj vezi sa stanjem okoline. Obradu podataka iz okoline te zaključivanje o odgovarajućem djelovanju robota moguće je temeljiti na principima strojnog učenja.

Predloženo istraživanje bavi se razvojem modela učenja i planiranja djelovanja robota. Proces učenja temelji se na novoj umjetnoj neuronskoj mreži klasifikacijom prostornih struktura. Pod pojmom prostorne strukture podrazumijeva se interpretacija rasporeda poznatih objekata u ravnini koje robot percipira vizijskim sustavom. Umjetna neuronska mreža za klasifikaciju i prepoznavanje prostornih struktura zasniva se na teoriji adaptivne rezonancije. Razvijena je nova dvorazinska klasifikacija po obliku i rasporedu objekata te funkcija preslikavanja neuređenog skupa objekata na poznate klasificirane strukture. Planiranje djelovanja robota temeljno je na usporednoj evoluciji rješenja razvojem novoga genetskog algoritma. Novi genetski algoritam kao osnovni cilj ima prostornu pretvorbu neuređenog stanja objekata u uređeno.

Ključne riječi: robotika, autonomni sustavi, teorija adaptivne rezonancije, genetski algoritmi.

Extended summary

Chapter 1 “Introduction” gives an overview of the doctoral thesis, the current state of the art and the main goal and hypotheses. Today the development of technical system is characterized by the development of autonomous technical systems which are able to interpret their environment and learn based on experience. An autonomous robot [1-3] is a device with certain motor skills, sensors which provide feedback from the dynamic environment, with which the robot is in continuous interaction. It is an intelligent, and to a certain extent self-sufficient machine that can operate in real world environments without explicit human control, based on control algorithms. In nondeterministic environments control algorithms become explicit functions of position with respect to time and feedback of the current environment state. An autonomous robot is able to act in a real environment without direct human impact.

The motivation scenario of the proposed research assumes that the application of robots is still extremely complex and that it requires highly educated and specialised experts. The main assumption in this research is the process of learning that is a result of visual interpretation of the environment. Human interpretation of the environment (workspace) is based on context and more precisely on structural arrangement of objects and their respective meaning. As an example, the industrial process of product assembly can be considered. A person can intuitively connect arbitrarily spaced mounting elements in the workspace, bringing them into mutual spatial and geometrical relationship. The process is done by precisely executing a defined sequence of actions required to build the final product. Another example could be taken from medical practice, i.e. from a surgical procedure. A defined collection of surgical instruments, objects and tools in the operating theatre indicates a specific surgery procedure. It also specifies a series of actions and manipulation of individual instruments that the appropriate surgery procedure requires. These quite different examples clearly demonstrate that by observing a limited working domain one can make certain assumptions about the set of actions i.e. about the behaviour associated with the current arrangement and meaning of objects in the workspace. A robotic system should be capable of observing, mapping and finally learning object spatial layout within the work environment. The most important step in

this complex process is the learning ability which will be attributed by the developed ARTgrid classification architecture.

The proposed research is dealing with two main issues. The first problem is learning and classification of spatial structures by visual interpretation of robot workspace. The initial knowledge of the working domain can be coded in form of a knowledge base [5] or expert systems [6]. For working domains which are not static this is done by developing robot control models that can learn new information based on experience [7]. In the process of developing autonomous robotic systems, learning models can be based on statistical methods [8], reinforcement learning [9], neural networks [10] and other principles. Moreover these principles can be implemented as part of a single robot system or for multiagent robotic systems [11]. The second problem which this research addresses is robot behaviour planning [12] by which the robot will be able to build recognized spatial structures from an unordered set of known objects.

The development of robot control models which have the ability to adopt to dynamic changes in the environment is a very actual topic [26-28]. One group of research is focusing on developing new neural networks based on Adaptive resonance theory which were initially designed by Grossberg, Carpenter and associates. Some of the most widely used neural networks based on ART are: ART 2 [34], ART 2A [35], ART 3 [36], Fuzzy ART [37], SF-ART [38], Fusion ART [39]. ART neural networks are used within control models providing functionality such as: one shot learning and incremental learning. In the robotic community the implementation of ART neural networks can be found in [39-46]. Fuzzy ARTMAP is used for learning a set of robot motion primitives in assembly and manipulation scenarios [39]. FusionART is used for parallel information processing from robot sensory systems [40]. Unsupervised learning of ART-2 network based on cross-modality learning is described in [41]. For classification of robot motions an ARTMAP neural network was used [42]. Other control architecture based on incremental learning strategies are described in [30,31,51,52]. These approaches use spectral clustering [30], functional visual processing by dividing short term and long term memory [31], crowdsourcing [51] and sub-symbolic interpretation of the environment [52]. A description of architectures and algorithms for robot behaviour planning can be found in [54-64]. In [55] a robot planning model was developed based on first order logic and verified on an actual robot system. A multichannel ART neural network was used

in [56] for planning in a virtual environment using software agents. Behaviour planning based on sound and gestures is described in [59,60]. Ontologies are used in [62] for the purpose of behaviour planning in a specific industrial domain. A genetic algorithm for task scheduling is developed in [64].

The main goal of this research is development and verification of a control model that will enable a robot to recognize and build ordered spatial structures, based on two dimensional visual interpretation of the environment. Based on the main goal of the research our hypotheses are: I) recognition of spatial structures through two dimensional interpretation of object arrangement in robot workspace can be based on an ART neural network which associates structures and given objects by shape and position, II) the construction of spatial structures can be planned by a new genetic algorithm.

Chapter 2 “Learning and classification of spatial structures” describes the developed Adaptive resonance theory neural network. A robot collects information from the environment, it processes the information, reasons and makes decisions upon which it acts on the environment. Interaction with the environment is imperative for reasoning about a certain object or event. The main hypothesis is that similar stimuli activate a related set of actions. This scenario is used as the backbone for the development of the ARTgrid architecture used for classification of spatial structures. Objects in the robot workspace form a certain spatial structure. A spatiale structure can be defined at two levels of granularity. First the morphology i.e. the general shape can be recorded without regarding individual objects in the structure. At a more detailed level, for providing finer details, individual objects and their respective information is obtained (position, orientation). With the developed architecture it is possible to obtain a generalized concept for learning these sequences and for creating categories of similar activities. The generalization capability of the proposed NN is important for providing the learning agent information about solving new but similar tasks. The ARTgrid neural network is organized in two hierarchical levels - the Ga2 and Ga3. The Ga2 level records only certain features of the input space structures providing a more generalized method for category recognition. Within each Ga2 level, there exist a certain (dynamically increasing) number of related categories coded in the Ga3 level respectively. Each Ga2 category consists of a minimum one Ga3 category. Ga3 categories represent more specific forms of inputs passed into the parent Ga2 category. Furthermore, each Ga3 category has an

associated object matrix (MTO). The space structure categories within the Ga3 level are also generalized, but only to a certain degree of the Ga2 level. Purpose of the "dual level" architecture is to enable rapid search through the match tracking process where the best category match needs to be found for a given input structure. An input structure is first recognized at Ga2 level, and the output neuron that has the highest category choice function is activated. If the resonance value is higher than the vigilance parameter, the category is updated with the new input. The same process is repeated for the Ga3 level. If there is no output neuron that yields a resonance level higher than expected by the vigilance parameter a new category is formed by storing the input class in a new output neuron, either in Ga2 and Ga3 or only at Ga3 level. ARTgrid can be illustrated as a dual network system which passes registered categories from Ga2 toward the Ga3 level. Both modules have an additional Resonance adaptation subsystem (RAS) that is functioning within each step of the category choice and match tracking. RAS is used for additional control of category choices. RAS takes into account the object matrix (MTO) and controls an additional resonance gain that can either increase or decrease the resonance value based on object matrix matching. This process ensures that stable output categories both in Ga2 and Ga3 are created. The classification of ARTgrid is prone to produce a slight variety in output categories when different sequences of input structures are applied. As the input structures are not known in advance, neither their morphology nor their number and other parameters, the network can produce slightly varying outputs. The number of created categories varies both at the Ga2 and Ga3 level. This output is expected as the network does not have structured inputs but it learns and updates its long-term connection weights incrementally. In the case where the robot i.e. its learning model, doesn't have an a priori view of the world, the sequence of input patterns is arbitrary.

Chapter 3 "Associating unordered with ordered spatial states" describes how the assignment of an unordered spatial state is made to previously classified spatial structures. For the purpose of behaviour planning a new function was developed which is used to associate two object sets. For an unordered set of object (unordered spatial state) the best matching classified structure K should be found giving a final ordered set of objects. For connecting two object states the similarity function developed within the ART neural network are used combined with the Kuhn-Munkres assignment algorithm. When the assignment of individual objects from the robot workspace is made to a specific output class an optimal assignment of repetitive objects should be made. A cost matrix C is generated with values of Euclidian

distances of all object combinations. An ordered set of objects \mathcal{P} and \mathcal{K} is generated that will be used in the process of robot behaviour planning. The ordered sets \mathcal{P} and \mathcal{K} are used for unique object mapping.

Chapter 4 “Robot behaviour planning” describes the developed genetic algorithm and its verification. Robot behaviour planning can be defined as the process through which an agent organizes its actions. In this research robot behaviour is described as an order set of actions, which carried out sequentially transform a given initial state P to a final state K. The task for the planning algorithm is to find a feasible set of robot actions through which the initial state of the environment, given a finite number of discrete transformations, will be rearranged into a given final state. The environment is interpreted through position and orientation of objects. The solution to the planning problem is proposed as a two-step method. First a constructive heuristic is developed for making an initial set of “good” solutions. Then a genetic algorithm is developed that optimizes the initial set of solutions. The constructive heuristic uses only mutations for making an initial population of state transitions from P to K. It generates 30 solutions, and after three iterations a comparison of all solutions is made. One of the solutions with the lowest fitness is replaced by one of the fittest individuals through the process of cloning. Only mutations that yield a better fitness are used where all other mutations are not used. The fitness function is evaluated as the total time needed for the robot to rearrange the objects from the initial into the final set. In the second step the genetic algorithm is developed for optimizing the initial population. The solution from the constructive heuristic can be observed as a permutation of objects where each object has an associated position and orientation. Three mutation operators M1-M3 “remove and reinsert” [100], “swap” [101] and “invert” [102] are used. New rules are applied for each of the former mutation operators because certain solutions are not feasible. A new mutation operator M4 is developed which can change the position of a randomly selected object in P. Each solution is coded in parallel in two forms: permutation and position. Mutation operators M1-M3 only affect the permutation part of the chromosome where the mutation operator M4 changes only the position of an object. The crossover of individuals is also utilized but only the crossover of the permutations. A comparison of generated solution by the constructive heuristic, genetic algorithm and greedy search is made. The solutions generated by the constructive heuristic have the lowest fitness and are unstable while the solution from the genetic algorithm has an average fitness of 18% higher than by using greedy search.

Chapter 5 “Validation of the control model” - the complete control model consisting of the ART network, genetic algorithm for behaviour planning and developed robot control programs is verified. The experimental setup consists of an articulated robot arm (Fanuc M10-iA), an industrial vision system and a control computer which communicates with the robot controller through TCP/IP protocol. The experimental setup was verified at the Laboratory for Manufacturing and Assembly Systems Planning, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb. The first step of experimental validation is interpretation and recognition of spatial structures using the developed ART neural network and robot behaviour planning using the constructive heuristic and the new genetic algorithm. Two main functional components and associated robot control programs for I) learning and classification and II) spatial structure recognition and behaviour planning, were developed and implemented.

Chapter 6 “Conclusion” discusses the summary and the main achievements of the doctoral thesis. Original scientific contributions of this work can be summarised as: I) a self-organizing artificial neural network for classification of spatial structures based on Adaptive resonance theory characterized by a new two-level classification of object shape and arrangement and by a new mechanism for associating an unordered object set with an ordered object set, II) a new genetic algorithm for robot behaviour planning in an unstructured environment III) the robot control model comprised of the ART neural network, genetic algorithm and developed robot control programs. Future research will incorporate the possibility for classification of spatial structures through three-dimensional workspace interpretation. Furthermore, the developed robot control programs should be generalised for implementation on a broader range of industrial robots in the Laboratory.

Keywords: robotics, autonomous systems, Adaptive resonance theory, genetic algorithms.

Sadržaj

| | | |
|----------|----------------------------------------------------------------------|----|
| 1. | Uvod..... | 1 |
| 1.1. | Struktura doktorskog rada | 6 |
| 1.2. | Pregled dosadašnjih istraživanja..... | 7 |
| 1.2.1. | ART arhitekture..... | 8 |
| 1.2.2. | Ostale arhitekture | 10 |
| 1.2.3. | Planiranje robotskog djelovanja | 12 |
| 1.3. | Cilj i hipoteza rada..... | 15 |
| 1.4. | Metodologija i plan istraživanja | 16 |
| 1.4.1. | Faza 1. Razvoj i izrada upravljačkog modela | 16 |
| 1.4.2. | Faza 2. Implementacija i verifikacija na laboratorijskoj opremi..... | 18 |
| 2. | Učenje i klasifikacija prostornih struktura | 20 |
| 2.1. | Djelovanje robota u radnoj okolini..... | 20 |
| 2.2. | Teorija adaptivne rezonancije..... | 22 |
| 2.2.1. | AFS neuronska mreža | 23 |
| 2.3. | Razvoj nove ART neuronske mreže..... | 25 |
| 2.3.1. | Arhitektura ART mreže..... | 25 |
| 2.3.2. | Postupak učenja ART mreže | 27 |
| 2.3.3. | Prepoznavanje struktura prema obliku i rasporedu | 28 |
| 2.3.4. | Određivanje ulaznog redoslijeda struktura..... | 31 |
| 2.3.5. | Karakteristika učenja ART mreže | 34 |
| 2.3.6. | Pouzdanost klasifikacije – verifikacija..... | 37 |
| 2.3.6.1. | Opis validacije klasifikacije..... | 37 |
| 2.3.6.2. | Evaluacija klasifikacije – testna skupina I. | 40 |
| 2.3.6.3. | Evaluacija klasifikacije – testna skupina II..... | 47 |

| | | |
|----------|----------------------------------------------------------------------|-----|
| 3. | Povezivanje neuređenog prostornog stanja s uređenim | 49 |
| 3.1. | Funkcija povezivanja | 49 |
| 3.2. | Funkcija preslikavanja | 56 |
| 3.3. | Primjer povezivanja neuređenog stanja objekata s uređenim..... | 58 |
| 4. | Planiranje robotskog djelovanja | 60 |
| 4.1. | Neformalni opis problema planiranja | 61 |
| 4.2. | Definicija problema planiranja | 63 |
| 4.3. | Prijedlog rješenja planiranja djelovanja | 64 |
| 4.3.1. | Kreiranje inicijalnog skupa rješenja – konstruktivni algoritam | 64 |
| 4.3.2. | Genetski algoritam | 71 |
| 4.3.2.1. | Operatori mutacije | 72 |
| 4.3.2.2. | Operator križanja | 75 |
| 4.3.2.3. | Implementacija genetskog algoritma planiranja | 77 |
| 4.3.3. | Generiranje plana pohlepnim pretraživanjem | 84 |
| 5. | Validacija upravljačkog modela | 88 |
| 5.1. | Učenje i klasifikacija | 90 |
| 5.2. | Planiranje robotskog djelovanja | 92 |
| 5.2.1. | Generiranje inicijalnog plana djelovanja..... | 93 |
| 5.2.2. | Primjena genetskog algoritma | 95 |
| 5.3. | Sažeti pregled postupka validacije | 97 |
| 6. | Zaključak | 99 |
| | Literatura | 103 |
| | Prilog A | 112 |
| | Životopis..... | 113 |
| | Curriculum vitae..... | 116 |

Popis slika

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Slika 1-1. Lijevo: Proizvoljno raspoređeni ugradbeni elementi; desno: ugradbeni elementi tvore funkcionalne sklopove. | 2 |
| Slika 1-2. Prikaz robotskog sustava. | 4 |
| Slika 1-3. Grupiranje simuliranih struktura upotrebom principa ART-a (Tscherepanow i suradnici [50]). | 9 |
| Slika 1-4. Klasifikacija objekata primjenom principa ART-a (Kirstein i suradnici [30]). | 10 |
| Slika 1-5. Učenje i izgradnja dvodimenzionalnih struktura unutar procesa generalizacije prikupljenih informacija (Chung i suradnici [51]). | 11 |
| Slika 1-6. Planiranje djelovanja robota temeljeno na simboličkom prikazu radnog prostora (Dearden i Burbridge [54]). | 12 |
| Slika 1-7. Generiranje plana temeljenog na logici prvog reda za izgradnju prostornih struktura (Ekvall i Kragić [55]). | 13 |
| Slika 1-8. Generiranje plana za softverske agente korištenjem iFALCON arhitekture zasnovane na ART-u (Subagdja i Tan [56]). | 13 |
| Slika 1-9. Planiranje izgradnje prostornih struktura u interakciji s čovjekom (Rickert i suradnici [57]). | 14 |
| Slika 2-1. a) Slijed toka informacija i djelovanja prilikom interakcije robota s okolinom b) Predloženi model sastavljen od triju osnovnih komponenata. Crvene strelice označavaju djelovanja, dok zelene označavaju prikupljanje informacija iz okoline. (AA: autonoman agent). | 21 |
| Slika 2-2. Arhitektura Fuzzy ART neuronske mreže. | 22 |
| Slika 2-3. Arhitektura AFS neuronske mreže. | 23 |
| Slika 2-4. Prikaz svih podsustava unutar razvijene ART mreže. | 26 |
| Slika 2-5. Dvorazinska arhitektura razvijene ART mreže. | 26 |
| Slika 2-6. Primjer 14 slučajno generiranih prostornih struktura. | 33 |
| Slika 2-7. a) Primjer izlazne klase karakterizirane matricom 200 x 200 b) 12 prostornih struktura koje su primijenjene na klasu prikazanu pod a). | 33 |
| Slika 2-8. Primjer klasifikacije u 4 izlazne Gal klase (prikazane u 1. stupcu s lijeva). | 34 |
| Slika 2-9. Arhitektura predložene ART mreže. | 34 |
| Slika 2-10. Brzina pretraživanja postojećih klasa razvijene ART mreže za $\varphi=0,8$ $\varphi=0,85$ i $\varphi=0,9$ | 36 |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Slika 2-11. Brzina pretraživanja postojećih klasa razvijene ART mreže za $\varphi=0,9$ $\varphi=0,95$ i $\varphi=0,98$ | 36 |
| Slika 2-12. Učenje prostornih struktura. | 41 |
| Slika 2-13. Primjer 70 prostornih struktura generiranih pomoću algoritma (Algoritam 2-3.). | 42 |
| Slika 2-14. Različite prostorne strukture sastavljene od poznatih objekata. | 42 |
| Slika 2-15. Primjer 70 prostornih struktura generiranih pomoću algoritma (Algoritam 2-3.) te označenih prema pripadnosti jednoj od 4 moguće klase I, II, III ili IV. | 43 |
| Slika 2-16. Prikaz Kappa-koeficijenta za 20 različitih slučajeva učenja ART mreže sa skupom za učenje koji je prikazan na slici (Slika 2-15.). | 46 |
| Slika 2-17. Skup od 47 prostornih struktura za testiranje točnosti klasifikacije. | 47 |
| Slika 2-18. Prikaz parametara točnosti klasifikacije za klasifikaciju prostornih struktura prikazanih na slici (Slika 2-17.). | 48 |
| Slika 3-1. Neuređeno prostorno stanje objekata. | 49 |
| Slika 3-2. Trivijalan slučaj preslikavanja neuređenog stanja a) na postojeću, prethodno klasificiranu strukturu b). | 50 |
| Slika 3-3. a) neuređeno prostorno stanje objekata, b) postojeće izlazne klase. | 50 |
| Slika 3-4. Osam vrsta objekata. | 51 |
| Slika 3-5. Skica prostornog preslikavanja neuređenog stanja objekata u uređeno. | 54 |
| Slika 3-6. Skica prostornog preslikavanja neuređenog stanja objekata u uređeno. | 56 |
| Slika 3-7. Primjer početnog (lijevo) i konačnog (desno) stanja za koje treba odrediti preslikavanje pojedinačnih predmeta iz početne u konačnu strukturu. | 56 |
| Slika 3-8. Povezivanje neuređenog prostornog stanja s poznatim klasificiranim strukturama. | 58 |
| Slika 3-9. Dijagram toka procesa povezivanja neuređenog prostornog stanja s postojećim klasificiranim strukturama. | 59 |
| Slika 4-1. Primjer početnog (lijevo) i konačnog (desno) stanja za koje je potrebno generirati plan robotskog djelovanja. | 61 |
| Slika 4-2. Lijevo: početno stanje objekata \mathcal{P} , desno: konačno stanje objekata \mathcal{K} | 62 |
| Slika 4-3. Vjerojatnost mutacije oko točke (12,14). Lijevo – $\sigma=2a$, desno $\sigma=a$ | 66 |
| Slika 4-4. Primjer generiranog rješenja koje se sastoji od 12 koraka. | 68 |
| Slika 4-5. Usporedba dobrote 5 različitih rješenja dobivenih konstruktivnim genetskim algoritmom evaluacijom funkcije dobrote iz jednadžbe (4.3). | 69 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Slika 4-6. Prikaz 4 diskretna koraka pri generiranju rješenja pomoću konstruktivnog genetskog algoritma. Prikazani su koraci 3, 7, 14 i završni korak 18, kada je generirano posljednje rješenje. | 70 |
| Slika 4-7. Lijevo: početno stanje objekata \mathcal{P} , desno: konačno stanje objekata \mathcal{K} | 77 |
| Slika 4-8. Prikaz početnog i konačnog stanja u radnom prostoru s vidljivim preklapanjem svih objekata. | 78 |
| Slika 4-9. Jednostavan slučaj planiranja – izravno postavljanje svih objekata na konačni položaj. | 78 |
| Slika 4-10. Slučaj kada nije moguće izravno postaviti sve objekte u konačni položaj zbog kolizija. | 79 |
| Slika 4-11. Konvergencija rješenja za vjerojatnost križanja $P(c) = 0,1$ te različite vrijednosti vjerojatnosti mutacija $P(m)$ | 80 |
| Slika 4-12. Konvergencija rješenja za vjerojatnost mutacije $P(m) = 0,1$ te različite vrijednosti vjerojatnosti križanja $P(c)$ | 80 |
| Slika 4-13. Konvergencija rješenja za vjerojatnost mutacije $p_m = 0,1$ te različite vrijednosti vjerojatnosti križanja p_c | 81 |
| Slika 4-14. Utjecaj odnosa p_m i p_c na dobrotu rješenja Φ . Raspon $p_m = [0 - 1]$, $p_c = [0 - 1]$. Na grafikonu je prikazan medijan od 10 simulacija za svaki par: ukupno 1210 simulacija. | 81 |
| Slika 4-15. Utjecaj odnosa p_m i p_c na dobrotu rješenja Φ . Raspon $p_m = [0 - 1]$, $p_c = [0 - 1]$. Na grafikonu je prikazana srednja vrijednost od 10 simulacija za svaki par: ukupno 1210 simulacija. | 82 |
| Slika 4-16. Utjecaj odnosa p_m i p_c na dobrotu rješenja Φ . Raspon $p_m = [0 - 1]$, $p_c = [0 - 1]$. Dijagram prikazuje jedno od 8 rješenja korištenih u generiranju dijagrama na slici (Slika 4-14.) i (Slika 4-15.). | 82 |
| Slika 4-17. Najbolje rješenje za problem prikazan na slici (Slika 4-8.). Dobrota rješenja iznosi 2497. | 83 |
| Slika 4-18. Tri rješenja za problem prikazan na slici (Slika 4-8.). Dobrota prvog rješenja: 2544, dobrota drugog rješenja: 2562, Dobrota trećeg rješenja: 2539. | 84 |
| Slika 4-19. Prikaz najboljeg rješenja dobivenog pohlepnim pretraživanjem, $\Phi = 2889$ | 86 |
| Slika 4-20. Usporedba najboljih rješenja dobivenih pohlepnim pretraživanjem s konstruktivnim i genetskim algoritmom. | 87 |
| Slika 5-1. Prikaz laboratorijskog postava. I) robot, II) robotska upravljačka jedinica, III) računalo, IV) vizijski sustav (kamera), V) radni prostor robota. | 88 |
| Slika 5-2. Osnovne cjeline upravljačkog modela. | 89 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Slika 5-3. Dijagram toka validacije upravljačkog modela na laboratorijskoj opremi. | 90 |
| Slika 5-4. Učenje prostornih struktura. Inicijalizacija robotskog sustava započinje demonstracijom različitih prostornih struktura koje se klasificiraju pomoću algoritma (Algoritam 2-2.). | 91 |
| Slika 5-5. Povezivanje neuređenog prostornog stanja s prethodno klasificiranim prostornim strukturama: a) neuređeno stanje objekata u radnom prostoru robota i prikaz objekata lokaliziranih pomoću vizijskog sustava, b) postavljanje hipoteza o izgradnji poznatih klasa, c) izbor pobjedničke klase. | 92 |
| Slika 5-6. a) početno stanje, b) generirano konačno stanje, c) izgrađeno konačno stanje. | 93 |
| Slika 5-7. Najbolje rješenje (11 koraka) dobiveno konstruktivnim algoritmom – $\Phi_0=3163$... | 94 |
| Slika 5-8. Najbolje rješenje (9 koraka) dobiveno genetskim algoritmom – $\Phi_K=2517$ | 96 |
| Slika 5-9. Izvršavanje plana na realnom robotskom sustavu: a) prikaz robota u početnoj točki, b) neuređeno početno stanje objekata, c) – g) izuzimanje i odlaganje prvog objekta, h) – k) izuzimanje i odlaganje ostalih objekata, l) izgrađena uređena prostorna struktura. | 97 |

Popis tablica

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tablica 1-1. <i>Usporedba upravljačkih arhitektura</i> | 11 |
| Tablica 1-2. <i>Usporedba modela i arhitektura planiranja robotskog djelovanja</i> | 15 |
| Tablica 2-1. <i>Konfuzijska matrica dvaju razreda (dviju klasa)</i> | 37 |
| Tablica 2-2. <i>Primjer konfuzijske matrice za klasifikaciju u četiri razreda</i> | 39 |
| Tablica 2-3. <i>Primjer konfuzijske matrice za slučajni klasifikator koji raspoređuje ulazne uzorke u omjeru kao i promatrani klasifikator iz tablice (Tablica 2-2.)</i> | 40 |
| Tablica 2-4. <i>Konfuzijska matrica između označivača 1 i označivača 2</i> | 44 |
| Tablica 2-5. <i>Konfuzijska matrica između označivača 1 i označivača 3</i> | 44 |
| Tablica 2-6. <i>Konfuzijska matrica između označivača 2 i označivača 3</i> | 44 |
| Tablica 2-7. <i>Ukupna mjera sklonosti slaganju</i> | 45 |
| Tablica 2-8. <i>Uprosječna konfuzijska matrica za slučaj klasifikacije 70 prostornih struktura prikazanih na slici (Slika 2-15.). U zgradama je naznačena standardna devijacija</i> | 46 |
| Tablica 3-1. <i>Inicijalno pridruživanje objekata iz početnog stanja u konačno</i> | 57 |
| Tablica 3-2. <i>Optimalno pridruživanje objekata iz početnog stanja u konačno prema F</i> | 58 |
| Tablica 4-1. <i>Niz instrukcija generiranog plana prikazanog na slici (Slika 4-4.)</i> | 68 |
| Tablica 4-2. <i>Izmjerena vremena T_1 i T_2 linearnih gibanja robotske ruke</i> | 71 |
| Tablica 4-3. <i>Najbolje rješenje (10 koraka) dobiveno pohlepnim pretraživanjem, $\{x_1, y_1, \delta_1\}$: pozicija izuzimanja, $\{x_2, y_2, \delta_2\}$: pozicija odlaganja, $\Phi_K = 2889$</i> | 86 |
| Tablica 5-1. <i>Raspodjela dobrote rješenja dobivene konstruktivnim algoritmom</i> | 94 |
| Tablica 5-2. <i>Najbolje rješenje (11 koraka) dobiveno konstruktivnim algoritmom, $\{x_1, y_1, \delta_1\}$ – pozicija izuzimanja, $\{x_2, y_2, \delta_2\}$: pozicija odlaganja, $\Phi_0 = 3163$</i> | 95 |
| Tablica 5-3. <i>Raspodjela dobrote rješenja dobivene pomoću genetskog algoritma</i> | 95 |
| Tablica 5-4. <i>Najbolje rješenje (9 koraka) dobiveno genetskim algoritmom, $\{x_1, y_1, \delta_1\}$: pozicija izuzimanja, $\{x_2, y_2, \delta_2\}$: pozicija odlaganja, $\Phi_K = 2517$</i> | 96 |

Popis algoritama

| | |
|-----------------------------------------------------------------------------------------------------|----|
| Algoritam 2-1. <i>ARTgrid inicijalizacija</i> | 27 |
| Algoritam 2-2. <i>Dvorazinski ART s podsustavom prilagodbe vrijednosti rezonancije</i> | 30 |
| Algoritam 2-3. <i>Generator slučajnih struktura</i> | 32 |
| Algoritam 2-4. <i>Pretraživanje klasa i učenje mreže – AFS</i> | 35 |
| Algoritam 3-1. <i>Preslikavanje neuređenog prostornog stanja na klasificirane strukture</i> | 55 |
| Algoritam 4-1. <i>Konstruktivni algoritam</i> | 67 |
| Algoritam 4-2. <i>Optimizacijski genetski algoritam</i> | 74 |
| Algoritam 4-3. <i>Verifikacija plana djelovanja</i> | 75 |
| Algoritam 4-4. <i>Usporedi permutacije</i> | 76 |
| Algoritam 5-1. <i>Robotski upravljački program</i> | 98 |

Popis oznaka i kratica

| Popis oznaka | | |
|-----------------------|-------------------------------------------------------|-----------------|
| Oznaka | Značenje | Mjerna jedinica |
| \mathcal{P} | Skup objekata u početnom stanju | – |
| \mathcal{K} | Skup objekata u konačnom stanju | – |
| \mathcal{A} | Uređeni skup robotskih akcija | – |
| \mathbf{P} | Matrica početnog stanja objekata | – |
| \mathbf{K} | Matrica konačnog stanja objekata | – |
| \mathbf{A} | Matrica trenutnog stanja objekata | – |
| β_1 | Parametar učenja – stupanj 1. | – |
| β_2 | Parametar učenja – stupanj 2. | – |
| φ_1 | Parametar budnosti – stupanj 1. | – |
| φ_2 | Parametar budnosti – stupanj 2. | – |
| \mathbf{I}_g | Ulazna struktura | – |
| N_p | Broj izlaznih klasa | – |
| N | Broj ulaznih struktura | – |
| \mathbf{W}_j | Matrica težinskih vrijednosti | – |
| $Ga2$ | Polje klasa | – |
| T_{i1} | Funkcija odabira | – |
| m_{iJ} | Funkcija preklapanja | – |
| \mathcal{W} | Skup svih težinskih vrijednosti | – |
| C_{\max} | Maksimalan broj centara | – |
| \mathcal{U} | Skup parametara za generiranje prostornih struktura | – |
| b | Ukupan broj struktura | – |
| p_x, p_y, p_φ | Vjerojatnosti promjene vrste, položaja i kuta objekta | – |
| μ_x | Očekivanje koordinate x | – |
| μ_y | Očekivanje koordinate y | – |
| μ_z | Očekivanje koordinate φ | – |
| σ_x | Standardna devijacija koordinate x | – |
| σ_y | Standardna devijacija koordinate y | – |

| | | |
|--------------------|--------------------------------------------------------|------|
| σ_φ | Standardna devijacija koordinate φ | – |
| v | Brzina | mm/s |
| ω | Kutna brzina | °/s |
| d_1 | Ulazna struktura najveće gustoće | – |
| d_2 | Ulazna struktura najmanje gustoće | – |
| U_i | Lokalizirani objekt | – |
| S_j^k | Objekt iz postojeće klasi | – |
| H, Q | Matrica kvantifikacije sličnosti oblika | – |
| F | Matrica optimalnog pridruživanja | – |
| C | matrica cijena | – |
| Φ | Funkcija dobrote | – |
| \mathcal{O} | Skup svih generiranih planova iz GA | – |
| O_i | Jedinični plan djelovanja | – |
| T_i | Vrijeme izuzimanja objekta | s |
| $M1^1, M2^1, M3^1$ | Modificirani operatori mutacije | – |
| $M4$ | Operatori mutacije položaja | – |
| p_c | Vjerojatnost križanja | – |
| p_m | Vjerojatnost mutacije | – |
| O_{opt} | Najbolji plan djelovanja | – |
| p | Permutacija redoslijeda | – |
| x_i, y_i | Položaj u referentnom koordinatnom sustavu robota | mm |
| δ_i | Orijentacija u referentnom koordinatnom sustavu robota | ° |

Popis kratica

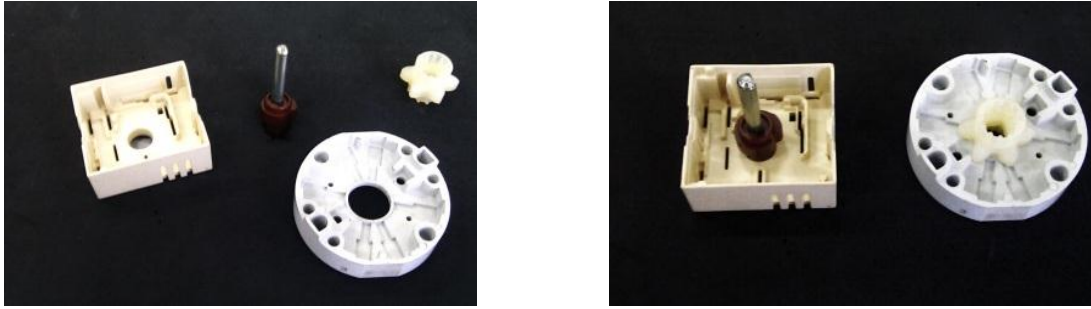
| | |
|-----|-------------------------------------------------------------------|
| ART | <i>Adaptive resonance theory</i> (Teorija adaptivne rezonancije) |
| AFS | <i>Adaptive fuzzy shadow network</i> (Neizrazita adaptivna mreža) |
| MTO | Matrica sličnosti objekata |
| GA | Genetski algoritam |
| SSG | Stupanj slobode gibanja |
| AI | <i>Artificial Intelligence</i> (Umjetna inteligencija) |
| MAS | <i>Multiagent systems</i> (Višeagentski sustavi) |

1.UVOD

Danas razvoj tehnike ponajviše karakterizira razvoj autonomnih tehničkih sustava koji su sposobni tumačiti djelovanje okoline te učiti na temelju iskustva. Autonoman robot [1-3] je uređaj s motoričkim sposobnostima i sensorima za prikupljanje povratnih informacija iz promjenjive okoline. Prikupljene informacije obrađuje upravljački program odgovoran za rad izvršnih elemenata ostvarujući aktivnu interakciju s okolinom u realnom vremenu. U određenoj mjeri to je samodostatan uređaj koji može djelovati u realnom svijetu bez izravnog ljudskog utjecaja.

Motivacijski scenarij predloženog istraživanja polazi od pretpostavke da je primjena robota još uvijek iznimno složena te traži visokoobrazovane i specijalizirane stručnjake za puštanje u rad i održavanje. Ovim istraživanjem želi se pridonijeti razvoju robotskih sustava koji će u budućnosti biti jednostavniji za korištenje, kako u industrijskim tako i u uslužnim djelatnostima.

Osnovna pretpostavka od koje se polazi u ovom istraživanju je proces učenja koji proizlazi kao odgovor na vizualnu spoznaju okoline. Ljudska se interpretacija okoline pretežno zasniva na tumačenju objekata i njihova prostornog odnosa. Kao primjer moguće je uzeti proces sklapanja industrijskih proizvoda. Čovjek je sposoban razumjeti veze između proizvoljno raspoređenih ugradbenih elementa u radnom prostoru te osmisliti slijed aktivnosti koje ih dovode u odgovarajući geometrijski odnos tvoreći funkcionalni sklop (Slika 1-1.). Kao drugi primjer moguće je uzeti rad kirurga tijekom operativnog zahvata [4]. Određen položaj kirurških pomagala i alata na stolu s instrumentima ukazuje na vrstu operacije koja se obavlja, ali i na niz radnji te rukovanja s pojedinim instrumentima koje odgovarajući operativni zahvat podrazumijeva. Navedeni primjeri ukazuju na to kako je promatranjem ograničene radne domene moguće donijeti određene zaključke o složenom djelovanju povezanom s trenutnim položajem i značenjem objekata u radnom prostoru.



Slika 1-1. Lijevo: Proizvoljno raspoređeni ugradbeni elementi; desno: ugradbeni elementi tvore funkcionalne sklopove.

Predloženo istraživanje u svojoj osnovi bavi se dvama osnovnim problemima koje je moguće identificirati iz prethodnih tvrdnji.

Prvi je problem učenja i klasifikacije prostornih struktura vizualnom interpretacijom radnog prostora robota. Kako bi robot mogao donositi određene odluke i na temelju njih djelovati na svoju okolinu, potrebno je inicijalno znanje o domeni problema u kojoj će robot djelovati. Znanje te informacije o okolini moguće je unaprijed kodirati u obliku baza znanja [5] ili ekspertnih sustava [6] te se njima koristiti kao integriranim komponentama robotskih upravljačkih modela. Takav je pristup efikasan u onim okruženjima koja su predvidiva te tamo gdje su unaprijed poznate moguće situacije u kojima se robotski sustav može zateći. Uzevši u obzir činjenicu da većina realnih radnih okolina nije u potpunosti statička, postoji mogućnost nepredvidivih situacija tj. stanja okoline koja nisu bila predviđena u procesu inicijalnog razvoja upravljačkog modela. Zbog toga se sve više razvijaju i implementiraju algoritmi i modeli učenja [7] koji omogućavaju prilagodbu novim stanjima okoline. Ti se algoritmi i modeli najčešće upotrebljavaju kao dijelovi sveobuhvatnih upravljačkih modela robota. Pri razvoju autonomnih robotskih sustava modeli učenja mogu se temeljiti na statističkim metodama [8], učenju pojačavanjem (eng. *reinforcement learning*) [9], neuronskim mrežama [10] te drugim principima. Navedeni pristupi mogu biti implementirani na pojedinačnim robotima, kao i u okviru grupe robota [11]. Modeli i algoritmi koji imaju mogućnost inkrementalnog tj. kontinuiranog ažuriranja i dopunjavanja omogućavaju fleksibilniji rad.

Kao drugi problem ističe se planiranje djelovanja robota pomoću kojeg će robot biti sposoban izgraditi prepoznate prostorne strukture iz neuređenog skupa poznatih objekata. Planiranje djelovanja [12] (engl. *behavior planning, task planning*) razlikuje se od problema planiranja putanje (engl. *motion planning, path planning*) [13,14]. Cilj planiranja putanje je pronalaženje izvedive ili najkraće putanje između dviju ili više točaka u prostoru. Zadatak planiranja

putanje je pronaći skup vrijednosti upravljačkih varijabli koje će omogućiti transformaciju položaja iz početnog kroz konačan broj međustanja u konačan položaj. S druge strane, cilj planiranja djelovanja robota je odrediti uređeni skup elemenata rada kojima će se početno stanje okoline i objekata koji se u njoj nalaze kroz konačan broj koraka preurediti u traženo konačno stanje. Problem određivanja uređenog skupa elemenata rada je od osnovnog interesa, dok se planiranje putanje može riješiti heurističkim metodama koje daju određeno, ali ne optimalno rješenje. Kao kriterij optimalnosti planiranja putanje moguće je uzeti ukupno vrijeme kretanja robotske kroz određen skup točaka u prostoru. U procesu planiranja putanje često je dovoljno pronaći izvediv plan koji nužno nije globalno optimalan. Planiranje putanje u tom smislu može biti integralni dio planiranja djelovanja [15,16] ili mogu biti uključena određena pojednostavljena. Planiranje djelovanja, kao i planiranje putanje, izrazito su kompleksni problemi te se istraživačka zajednica često njima bavi odvojeno.

Algoritmi učenja i klasifikacije te planiranje djelovanja sastavni su dio upravljačkog modela razvijenog u sklopu ovog istraživanja koji će osposobiti robota za prepoznavanje i izgradnju uređenih prostornih struktura. Problem planiranja djelovanja riješit će se za domenu problema izgradnje uređenih prostornih struktura pomoću poznatih objekata. Objekt u osnovi karakterizira njegov vanjski oblik, ali i druge značajke, kao što su: boja, materijal, unutrašnja geometrija, itd. Istraživanje će se ograničiti na interpretaciju značenja objekta putem prepoznavanja njegova vanjskog oblika. Vanjski oblik objekata, primjerice u procesima strojne montaže, određuje glavne plohe prihvata, mogućnosti manipulacije te druge bitne parametre procesa. S druge strane, ljudska percepcija objekata velikim se dijelom zasniva na vizualnoj percepciji oblika [17] kojem čovjek pridaje određeno značenje. Zadaća prepoznavanja, klasifikacije i međusobnog povezivanja neuređenih prostornih struktura s uređenima je snalaženje robota u novim uvjetima radne okoline, kao i mogućnost korištenja prethodnih spoznaja u novim, ali sličnim problemima. Pod pojmom prostorne strukture podrazumijeva se dvodimenzionalna interpretacija rasporeda poznatih objekata u ravnini koje robot percipira pomoću vizijskog sustava. Primjer jednog takvog robotskog sustava te prostorne strukture prikazan je na slici (Slika 1-2.).



Slika 1-2. Prikaz robotskog sustava.

U ovom istraživanju pretpostavlja se da slična stanja okoline u kojoj se robot nalazi impliciraju slična složena robotska djelovanja. Zbog toga je moguće klasificirati prostor percepcije, kao i prostor djelovanja. Klasifikacija prostora percepcije je svojstvo da se prema određenim uvjetima sličnosti dvaju (ili više) lokaliziranih stanja robotske okoline svrstavaju u istu domenu problema tj. u istu klasu. Na taj je način moguće smanjiti količinu informacija koje robot pohranjuje, a s druge strane moguće je potpuno nove ulazne informacije svrstati u postojeće klase te primijeniti slične obrasce ponašanja. Inicijalne informacije koje upravljački model sadrži mogu biti u potpunosti predefinirane te nepromjenjive [5] ili je informacije o domeni u kojoj robot djeluje moguće prenijeti s čovjeka na robota u konačnom broju demonstracija [18]. Takva se inicijalna demonstracija najčešće odvija kao svojevrsna faza inicijalizacije robotskog sustava. Pristup koji je primijenjen u ovom istraživanju uključuje mogućnost kontinuirane te inkrementalne prilagodbe okolini stalnim ugađanjem parametara upravljačkog modela zaduženog za klasifikaciju stanja okoline. Osnovne karakteristike takvog pristupa uključuju uspješnu prilagodbu promjenljivim uvjetima okoline te pohranjivanje novih ulaznih uzoraka bez zaboravljanja prethodno klasificiranih.

Potrebno je napraviti kompromis između stabilnosti tj. očuvanja prethodno naučenih uzoraka, i plastičnosti tj. spremnosti da se uče novi uzorci. To je moguće ako sustav za klasifikaciju uspijeva razlikovati poznate ulazne uzorke tj. one uzorke koje je već upamtio i nepoznate uzorke, tj. one uzorke kojima još nije bio izložen [19].

Te karakteristike posjeduje teorija adaptivne rezonancije (engl. *adaptive resonance theory* – ART). ART je kognitivna (spoznajna) neuronska mreža koja pokušava objasniti kako ljudski mozak autonomno uči, klasificira, raspoznaje i predviđa događaje u promjenjivoj okolini. U svojoj osnovi ART [20-23] čini skup samoorganizirajućih natjecateljskih neuronskih mreža čija je svrha klasifikacija i grupiranje ulaznih uzoraka. Kako bi ART mreža mogla riješiti dvojbu stabilnosti i plastičnosti koju je postavio Grossberg, sustav za prepoznavanje i klasifikaciju uzoraka treba imati dva podsustava: podsustav pažnje i podsustav orijentacije. Podsustav pažnje obrađuje poznate uzorke, dok se podsustav orijentacije aktivira kada je pred neuronsku mrežu postavljen novi uzorak. Sa svrhom prepoznavanja prostornih struktura i njihove klasifikacije primijenjeni su osnovni principi teorije adaptivne rezonancije. Razvijena je nova konfiguracija ART neuronske mreže koja omogućava povezivanje struktura i pripadajućih objekata prema obliku i položaju u radnom prostoru.

Zadatak planiranja je pronaći mogući slijed radnji robota kojima će se početno stanje okoline kroz konačan broj koraka preurediti u traženo konačno stanje uz najmanji utrošak robotskog rada. Robotski rad interpretira se kroz utrošeno vrijeme i prijeđeni put za obavljanje potrebnih kretnji [24]. Za planiranje djelovanja robota koje kao ulazni skup parametara uzima početno i konačno stanje objekata u radnom prostoru razvijen je novi genetski algoritam. Genetski algoritam temelji se na dvjema osnovnim komponentama: konstruktivnoj i optimizacijskoj. Konstruktivni dio algoritma zadužen je za generiranje inicijalnog skupa rješenja kroz usporednu evoluciju većeg broja rješenja, njihovim međusobnim uspoređivanjem te ukidanjem loših i kloniranjem dobrih parcijalnih rješenja. Nakon generiranja inicijalnog skupa rješenja provodi se postupak optimizacije modificiranim operatorima mutacije te križanja kako bi se postigla konvergencija dobrote rješenja.

1.1. Struktura doktorskog rada

Doktorski rad sastoji se od pet poglavlja. U nastavku se sažeto opisuju pojedini dijelovi rada.

- U prvom, uvodnom, poglavlju predstavljena je motivacija istraživanja, pregled područja te dosadašnjih istraživanja. Postavljen je osnovni cilj istraživanja, nakon kojeg slijede hipoteze. Prikazan je plan i metodologija istraživačkog rada.
- Drugo poglavlje opisuje razvoj nove neuronske mreže temeljene na teoriji adaptivne rezonancije (ART) koja služi za učenje i klasifikaciju prostornih struktura. Opisane su nove značajke predloženog ART-a, koje uključuju dvorazinsku konfiguraciju i klasifikaciju prema obliku i rasporedu. Na kraju poglavlja prikazani su rezultati statističke analize klasifikacije prostornih struktura.
- U trećem je poglavlju opisana funkcija preslikavanja i povezivanja neuređenog prostornog stanja objekata s uređenim. Funkcija preslikavanja korištenjem razvijene ART mreže omogućava da se za određeno neuređeno stanje objekata u radnom prostoru robota odredi najbližnja postojeća prostorna struktura za koju se potom planira izgradnja.
- Genetski algoritam planiranja djelovanja opisan je u trećem poglavlju. Izložen je problem planiranja robotskog djelovanja te je u skladu s njime predložen novi genetski algoritam koji se sastoji od dviju osnovnih komponenata: konstruktivne i optimizacijske. Dobiveni rezultati uspoređeni su s heurističkim pohlepnim pretraživanjem te je ujedno prikazana analiza utjecaja različitih kombinacija vjerojatnosti mutacije i križanja.
- U četvrtom poglavlju detaljno je opisana implementacija razvijenog upravljačkog modela temeljena na novoj ART mreži te genetskom algoritmu planiranja robotskog djelovanja. Rezultati dobiveni na realnoj istraživačkoj opremi uspoređeni su s onima dobivenim simulacijama te je potvrđena primjenjivost razvijenih algoritama na stvarnom robotskom sustavu.
- U zadnjem poglavlju nalazi se rekapitulacija postignutih rezultata te je dan kritički osvrt. Navedeni su postignuti izvorni znanstveni doprinosi istraživanja. Dodatno su navedena ograničenja razvijenog modela i algoritama te opisana moguća područja primjene. Dane su preporuke i smjernice za buduća istraživanja.

1.2. Pregled dosadašnjih istraživanja

Razvoj robotskih upravljačkih modela koji imaju mogućnost prilagodbe promjenama u radnoj okolini aktualna je tema koja privlači intenzivnu pozornost istraživačke zajednice [25-27]. Istraživanje [28] govori o mogućnostima i prednostima primjene algoritama kao što su umjetne neuronske mreže te genetsko programiranje, koji su uspoređeni s tradicionalnim metodama. Stalna prilagodba omogućava upravljačkim modelima ažuriranje parametara radi lokalizacije te manipulacije objektima u nepoznatoj okolini bez poznavanja kalibracije vizijskog sustava.

Pohranjivanje i učenje ulaznih uzoraka u jednoj iteraciji (engl. *one shot learning*) [29,30] ili u manjem broju iteracija (engl. *k-shot learning*) omogućava prepoznavanje ulaznih uzoraka koji su lokalizirani svega nekoliko puta. Ovaj je proces bitan za stabilno funkcioniranje i djelovanje robotskih sustava u promjenljivoj okolini jer je potrebno moći pohraniti, a kasnije i prepoznati uzorke ili stanja okoline koje je robotski upravljački model susreo svega nekoliko puta ili tek jedanput. Uzorci mogu biti grupirani prema određenoj mjeri sličnosti, što omogućava kreiranje generaliziranih tj. prototipnih skupina uzoraka kojima se pridaje određeno značenje.

Nadalje, primjena navedenog pristupa u robotici sve je intenzivnija i češća. Autori u [31] razvijaju robotski upravljački model kojemu je cilj korištenje radnji, akcija i djelovanja koje je prethodno demonstrirao čovjek u sličnim stanjima okoline. Istraživanje se bavi problemom generalizacije postojećeg znanja kroz prilagodbu parametara kretanja robotske ruke te manipulacije poznatim predmetima.

Jedan od problema prilikom razvoja novih neuronskih arhitektura s kojima se istraživači vrlo često susreću je katastrofalno zaboravljanje [32]. Prilikom inkrementalnog ažuriranja postojećih klasa promjenama težinskih vrijednosti u neuronskoj mreži vrlo često dolazi do retroaktivne interferencije. U takvim slučajevima prilikom lokalizacije novih ulaznih uzoraka dolazi do neželjenih promjena na postojećim pohranjenim uzorcima te njihove klase mogu izgubiti svojstvenu prepoznatljivost. To znači da postojeća klasa više ne odgovara ulaznom uzorku koji je u nju bio klasificiran u prethodnim iteracijama. Predlažu se modeli koji smanjuju utjecaj retroaktivne interferencije u neuronskim arhitekturama [33], no taj je problem još uvijek sveprisutan u domeni praktične primjene neuronskih mreža koje koriste inkrementalno ažuriranje postojećih klasa [21].

1.2.1. ART arhitekture

Osnovne ART neuronske mreže koje su razvili Grossberg, Carpenter i suradnici kao što su ART 2 [34], ART 2A [35], ART 3 [36], Fuzzy ART [37], ali i druge ART arhitekture kao što su SF-ART [38], Fusion ART [39] poslužile su kao osnova za razvoj novih neuronskih arhitektura za primjenu u autonomnim robotskim sustavima.

ART neuronske mreže upotrebljavaju se u okviru upravljačkih modela koji imaju mogućnost učenja u jednoj iteraciji te inkrementalnog učenja sa svrhom klasifikacije ulaznih uzoraka. Ujedno, ART je vrlo interesantan istraživačima zbog svojih mogućnosti dobrog odnosa između karakteristika stabilnosti i plastičnosti [21]. Grossberg opisuje dilemu kako napraviti kompromis između stabilnosti tj. očuvanja prethodno naučenih ulaznih uzoraka, i plastičnosti tj. spremnosti učenja novih ulaznih uzoraka [19]. Istraživanja koja upotrebljavaju ili razvijaju umjetne neuronske mreže zasnovane na ART-u unutar robotske domene opisana su u [39-46], dok pregledni rad [47] govori o primjenama u ostalim inženjerskim problemima.

Fuzzy ARTMAP arhitektura [41] primijenjena je prilikom učenja niza osnovnih kretnji robota u procesima rukovanja i sklapanja objektima. Model je implementiran unutar upravljačke jedinice robota kao neuronski upravljački sklop (engl. *neural network controller* – NNC). Kao jedini modalitet za prikupljanje informacija iz okoline upotrijebljen je senzor sile i momenata na prirubnici robota. Upravljački model robota koristi se početnom bazom znanja koja se inkrementalno popunjava novim uzorcima prilikom sklapanja raznovrsnih objekata. Rezultati pokazuju visoku razinu generalizacije nakon učenja procesa sklapanja raznovrsnih objekata korištenjem iste početne baze znanja.

Arhitektura Fusion ART [39] upotrebljava se za paralelno procesiranje većeg broja ulaznih kanala sa senzorskih sustava robota pomoću Fuzzy ART neuronske mreže. Fusion ART sadrži niz međusobno povezanih ulaznih kanala, pri čemu se svaki kanal preslikava na zasebnu Fuzzy ART neuronsku mrežu. U predloženom modelu ART je upotrijebljen kao univerzalni mehanizam za klasifikaciju. Fusion ART obuhvaća i integrira tradicionalno različite koncepte učenja kao što su učenje bez učitelja i učenje s učiteljem, integracija znanja zasnovana na pravilima te učenje pojačavanjem (engl. *reinforcement learning*).

Nenadgledano učenje ART-2 neuronske mreže temeljeno na unakrsnim modalitetima za prepoznavanje i klasifikaciju gestikulacija opisano je u [42]. Implementacija istraživanja napravljena je na humanoidnoj robotskoj platformi. Hipoteza autora govori kako se robotsko

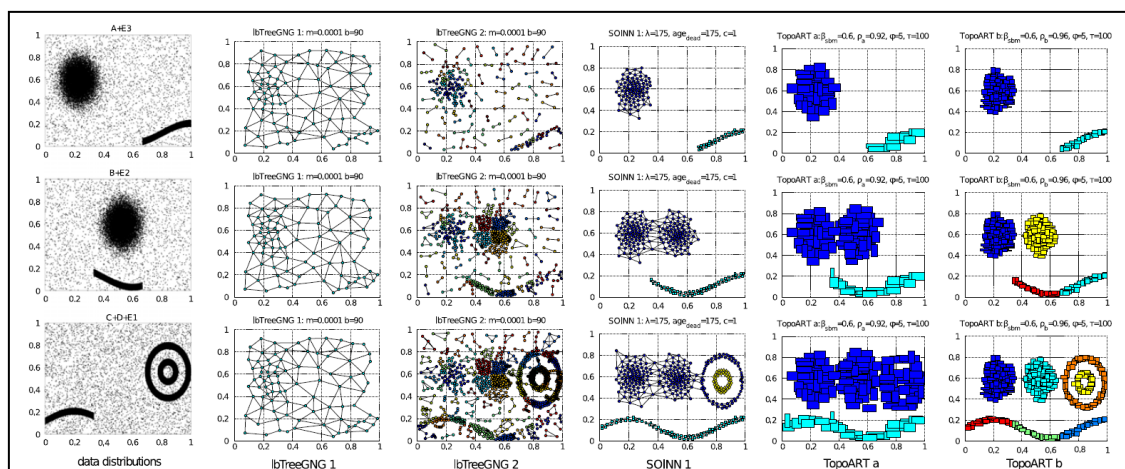
djelovanje može predstaviti kao klasifikacijski problem koji povezuje različite prostorno-vremenske strukture tj. gestikulacije. Model se koristi većim brojem ulaznih modaliteta: zvukom, gestama i vizualnom interpretacijom okoline.

U srodnom istraživanju nadgledano učenje ARTMAP neuronske mreže [43] upotrijebljeno je za klasifikaciju kretnji koje percipira robot. Kretnje predstavljaju prostorno-vremenski odnos položaja ruke čovjeka te se povezuju s određenim značenjem.

Fung i Liu [48] pokušavaju sprežanjem metoda iz teorije igara u svrhu adaptivne klasifikacije ART umjetnih neuronskih mreža optimirati veličine i broj izlaznih klasa izravnim utjecanjem na parametar budnosti mreže prilikom robotskog učenja.

Autori u [44] razvili su novi upravljački model CobART za klasifikaciju robotskog djelovanja temeljen na hijerarhijskom povezivanju ART neuronskih mreža. CobART omogućava klasifikaciju elementarnih i složenih robotskih djelovanja. Hijerarhijska povezanost upravljačkog modela ima svojstvo proširivosti te sadrži dijelove koji se mogu upotrebljavati pri formiranju novih klasa. Implementacija modela izvršena je u simulacijskom okruženju [49] te na realnom mobilnom robotu [45].

Arhitektura SiFuAM [46] prikazana na slici (Slika 1-3.) integrira ulazne podatke s tri robotska senzorska sustava sa svrhom grupiranja sličnih objekata. U [50] daje se prijedlog primjene nove neuronske mreže TopoART temeljene na Fuzzy ART-u. Osnovna značajka predložene arhitekture je mala osjetljivost na šum te hijerarhijska izrada klasa sa svrhom klasifikacije s više stupnjeva detalja. TopoART nije verificiran na realnom robotskom sustavu.



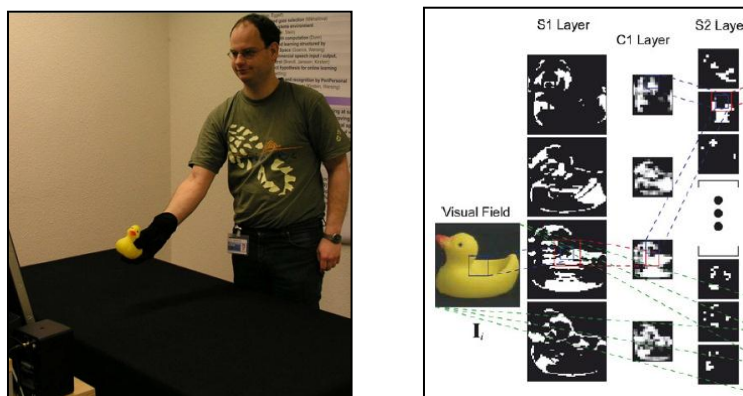
Slika 1-3. Grupiranje simuliranih struktura upotrebom principa ART-a (Tscherepanow i suradnici [50]).

1.2.2. Ostale arhitekture

Uz primjenu ART neuronskih mreža razvijaju se raznovrsne robotske upravljačke arhitekture koje imaju komponente inkrementalnog učenja [30,31,51,52] novih obrazaca djelovanja te različitih vizualnih uzoraka prikupljenih iz okoline.

Primjena prethodno grupiranih obrazaca robotskog djelovanja u budućim sličnim zadacima upotrebom spektralnog grupiranja (engl. *spectral clustering*) opisana je u [31]. Autori predlažu novi model pohranjivanja i grupiranja informacija kontinuiranim praćenjem promjena u radnoj okolini. Model je primijenjen na zadacima oponašanja kretanja čovjeka kao i manipulacije poznatim predmetima.

Učenje nepoznatih objekata hijerarhijskim vizualnim procesiranjem te funkcionalnim odvajanjem kratkoročne i dugoročne memorije po uzoru na ART prikazano je u [30]. Prepoznavanje objekata temelji se na obliku i boji objekata, što je prikazano na slici (Slika 1-4.). Prvi nivo modela služi za grubo prepoznavanje lokalne orijentacije primjenom principa „pobjednik uzima najviše“ (engl. *winner takes most mechanism*). Sljedeći nivo modela koristi se samo četvrtinom informacija iz prethodnog nivoa, ali je osjetljiv na kombinacije značajki iz prvog nivoa. Taj se proces ponavlja u tri nivoa kako bi se smanjila kompleksnost prikaza objekata. Ovo istraživanje predstavlja prvi uspješan pristup za pohranjivanje, prepoznavanje te klasifikaciju velikog broja objekata složene geometrije.



Slika 1-4. Klasifikacija objekata primjenom principa ART-a (Kirstein i suradnici [30]).

U istraživanju [51] razvijen je robotski sustav koji ima mogućnosti učenja od većeg broja ljudi koristeći se servisima pružanja usluga preko interneta (engl. *crowdsourcing*). Osnovna ideja sastoji se od sklapanja dvodimenzionalnih objekata pomoću kocaka različitih boja, gdje se kroz veći broj demonstracija obavlja generalizacija prikupljenih informacija sa svrhom kasnijeg autonomnog rekonstruiranja sličnih struktura (Slika 1-5.).



Slika 1-5. Učenje i izgradnja dvodimenzionalnih struktura unutar procesa generalizacije prikupljenih informacija (Chung i suradnici [51]).

U istraživanju [52] robotsko pohranjivanje i klasifikacija novih obrazaca djelovanja temelji se na oponašanju rada čovjeka. Osnovni cilj istraživanja očituje se u pronalasku veze između simboličkih informacija visokog nivoa apstrakcije te podsimboličkih informacija. Na simboličkom nivou generalizacija djelovanja u različitim okolinama i s različitim objektima napravljena je na temelju značajki i međusobnih odnosa objekata u prostoru. Osnovni problem navedenog istraživanja istaknut je u pronalasku podsimboličkih informacija koje će poslužiti za generiranje robotskih trajektorija koje su konzistentne s traženim simboličkim ciljevima.

U tablici (Tablica 1-1.) prikazana je usporedba postojećih arhitektura primijenjenih na robotskim sustavima.

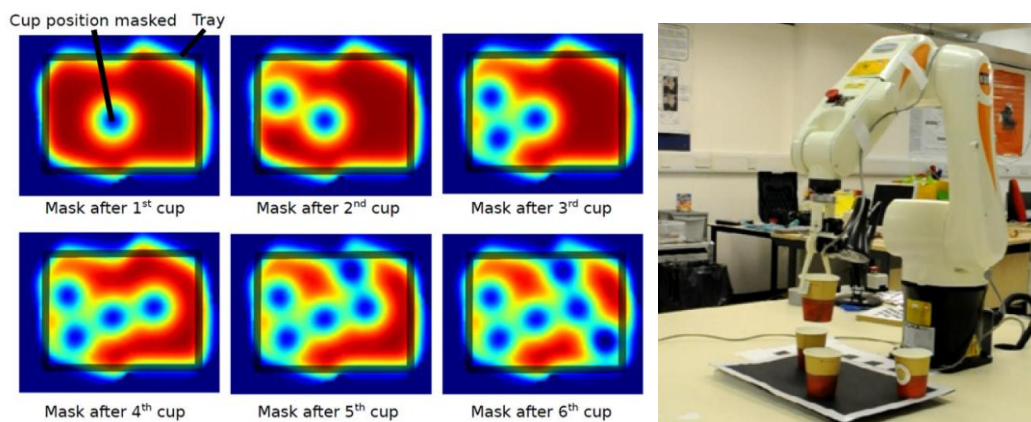
Tablica 1-1. Usporedba upravljačkih arhitektura.

| Arhitektura | Primjena | Validacija |
|-----------------------------------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| FAM [41,53] | Učenje obrazaca sklapanja. | Klasifikacija robotskih kretnji u odnosu na strukturu okoline. |
| fusionART [39] | Paralelno procesiranje većeg broja ulaznih kanala korištenjem više povezanih Fuzzy ART-a . | Klasifikacija vizualnih informacija sa senzorskih sustava robota. |
| ART-2 [42] | Klasifikacija robotskog djelovanja. | Klasifikacija šest različitih vrsta ponašanja na humanoidnom robotu. |
| ARTMAP [40] | Učenje obrazaca gibanja. | Sklapanje zasnovano na klasifikaciji oblika objekata. |
| CobART [45,49] | Učenje jednostavnih i složenih robotskih ponašanja. | Inkrementalno učenje novih obrazaca ponašanja temeljeno na strukturi radne okoline. |
| SiFuAM [46], TopoART [50] | Grupiranje uzoraka. | Hijerarhijska klasifikacija skupina objekata s više stupnjeva detalja. |
| Spektralno grupiranje [31] | Učenje značajki objekata. | Robotski sustav klasificira 50 raznovrsnih objekata. |
| Klasifikacija prostornih struktura [52] | Korištenje teorije grafova u svrhu modeliranja različitih prostornih struktura. | Klasifikacija prostornih struktura sastavljenih od objekata istovrsnog oblika. |

1.2.3. Planiranje robotskog djelovanja

Modeli, arhitekture te algoritmi planiranja robotskog djelovanja opisani su u [54 - 64]. Većina istraživanja koja se bavi planiranjem djelovanja ne bavi se izravno problemom planiranja putanja, već se koristi postojećim algoritmima planiranja putanja ili se koristi određenim heuristikama kako bi se pronašlo bilo koje rješenje za kretanje od početne točke do konačne. Djelovanje robota i izvršavanje određenog niza zadataka predstavljeno je kao apstrakcija visoke razine (engl. *high level of abstraction*) koja se rješava za određene domene problema, uvjete okoline ili specifične robotske kinematike.

Istraživanje opisano u [54], a prikazano slikom (Slika 1-6.), razvija robotski sustav planiranja djelovanja u procesima manipulacije predmetima. Zadaci koje robot treba riješiti predstavljeni su na simboličkoj razini kako bi bilo moguće pronaći veći broj geometrijskih rješenja. Razvijeni sustav planiranja djelovanja ima mogućnost dvosmjernog preslikavanja simboličkih stanja u geometrijska stanja radnog prostora i pripadajućih objekata. Verifikacija istraživanja izvršena je na industrijskom robotu sa šest stupnjeva slobode gibanja (SSG) koristeći se trima poznatim predmetima.



Slika 1-6. Planiranje djelovanja robota temeljeno na simboličkom prikazu radnog prostora (Dearden i Burbridge [54]).

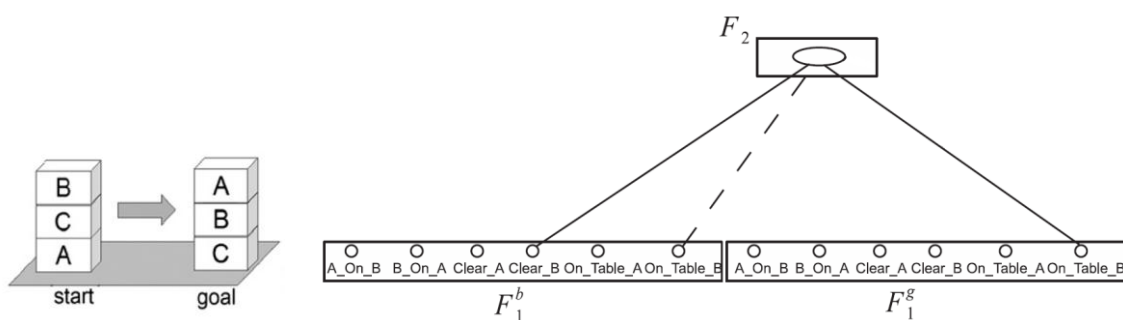
U istraživanju [55] razvijen je model planiranja robotskog djelovanja za zadaće manipulacije predmetima sa svrhom pretvorbe početnog položaja objekata u zadano konačno. Parcijalni planovi temelje se na logici prvog reda (engl. *first order logic*). Radi smanjenja kompleksnosti problema predefiniiran je konačan broj pomoćnih lokacija tj. međuspremnik (engl. *buffer location*) na koje je predmete moguće odložiti u slučaju da predmet nije moguće izravno postaviti na traženu konačnu poziciju. Sustav je preliminarno evaluiran u virtualnom okruženju te potom na realnoj robotskoj ruci sa šest SSG manipulacijom s tri objekta kao što

je prikazano na slici (Slika 1-7.). Podrazumijeva se da su svi objekti poznati te da je poznat način njihova izuzimanja te manipulacije.



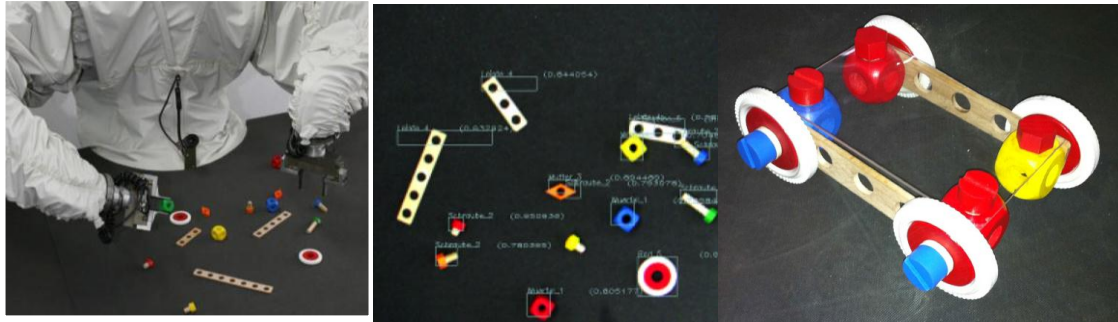
Slika 1-7. Generiranje plana temeljenog na logici prvog reda za izgradnju prostornih struktura (Ekvall i Kragić [55]).

Određivanje slijeda poznatih planova koji svojim izvršavanjem dovode do željenog konačnog stanja zasnovanog na ART-u opisano je u [56]. Pristup se zasniva na ekspertnom znanju promatrane domene koja sadržava cjelokupno semantičko i strukturalno znanje. Predložen je model hijerarhijskog planiranja u pojednostavljenom svijetu kocaka (engl. *simplified blocks world*) prikazanom na slici (Slika 1-8.). Model planiranja kao osnovne gradivne jedinice koristi višekanalnu ART neuronsku mrežu i novu gradijentnu klasifikaciju koja mijenja standardnu metodu aktivacije izlaznih klasa „pobjednik uzima sve” (engl. *winner-take-all*). Višekanalna ART mreža koristi veći broj fuzzy ART mreža u procesu planiranja. Implementacija te verifikacija predloženog modela izvršena je za virtualne agente.



Slika 1-8. Generiranje plana za softverske agente korištenjem iFALCON arhitekture zasnovane na ART-u (Subagdja i Tan [56]).

Suradnički rad čovjeka i robota za izgradnju pojednostavljenih trodimenzionalnih prostornih struktura prikazan je u [57], a razvoj navedenog sustava opisan je u [58]. Razvijeni robotski sustav ima predodređen cjelokupan plan procesa sklapanja koji se mijenja u interaktivnom djelovanju i pri sklapanju objekata zajedno s čovjekom.



Slika 1-9. Planiranje izgradnje prostornih struktura u interakciji s čovjekom (Rickert i suradnici [57]).

Planiranje djelovanja robota zasnovano na dodiru, zvuku i gestama prikazano je u [59,60]. Cilj razvijenog robotskog sustava je ostvariti intuitivno djelovanje robota u prostoru dijeljenom s čovjekom tijekom procesa industrijskog sklapanja te manipulacije poznatim predmetima. Planiranje robotskog djelovanja temelji se na dekomponiranju zadaća u jednostavnije strukture.

Za novu generaciju industrijskih robota [61] čiji je cilj raditi zajedno s ljudima potrebni su „inteligentni” sustavi upravljanja kako bi se smanjilo zahtijevano vrijeme programiranja te potrebna ekspertiza za integraciju takvih sustava. U istraživanju [62] implementiran je model planiranja koji uključuje ontologije koje opisuju određenu domenu djelovanja proizvodnih industrijskih robota. Osnovni cilj rada je opis mogućnosti robotskih sustava pomoću ontologija kako bi čovjek što jednostavnije mogao unijeti zadatke takvom sustavu. U procesu planiranja skupa djelovanja robota slijedno se dodaju akcije koje ispunjavaju prethodno zadane preduvjete. U koraku verifikacije upotrebljava se isti model planiranja kako bi se verificiralo udovoljavaju li određene sekvence prethodno postavljenim zahtjevima.

Problem usmjeravanja vozila koji je moguće riješiti uporabom genetskih algoritama [63] sadrži određene komponente koje su slične planiranju robotske manipulacije predmetima u nestrukturiranoj okolini. Analogija je vidljiva u optimiranju sume ukupnog prijeđenog puta robota od položaja izuzimanja do položaja odlaganja predmeta pronalaženjem optimalne ili vrlo dobre permutacije redoslijeda robotskih radnji, što je vrlo slično problemu posjećivanja određenog broja stajališta s autonomnim vozilom te optimiranjem ukupnog prijeđenog puta vozila.

Autori u [64] predlažu novi genetski algoritam za raspoređivanje zadataka industrijskom robotu. Osnovni cilj istraživanja očituje se u minimiziranju utrošenog vremena za izvršavanje svih kretanja robota između prethodno poznatih položaja na kojima se pojedini zadatak obavlja.

U tablici (Tablica 1-2.) prikazana je usporedba arhitektura planiranja.

Tablica 1-2. *Usporedba modela i arhitektura planiranja robotskog djelovanja.*

| Arhitektura | Primjena | Validacija |
|----------------------------|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Simboličko planiranje [54] | Povezivanje simboličkih i geometrijskih informacija u svijetu. | Planiranje manipulacije poznatim objektima u ravnini pomoću industrijskog robota sa šest SSG-a. |
| Logika prvog reda [55] | Planiranje izgradnje poznatih prostornih struktura. | Izgradnja struktura uporabom poznatih objekata pomoću industrijskog robota sa šest SSG-a. |
| ART iFALCON [56] | Modul planiranja softverskih agenata zasnovan na ART-u. | Planiranje sklapanja struktura u virtualnom svijetu. |
| Baza znanja [57,58] | Generiranje plana i sklapanje objekta na dvoručnom robotskom sustavu. | Predodređeni plan procesa sklapanja mijenja se tijekom interaktivnog djelovanja i sklapanja objekata zajedno s čovjekom. |
| Ontologije [62] | Planiranje djelovanja primjenom znanja pohranjenog u ontologijama. | Pojednostavljen rad sa standardnim industrijskim robotskim sustavima. |
| Genetski algoritmi [64] | Planiranje redoslijeda izvršavanja zadataka. | Za poznate lokacije na kojima robot izvršava zadatke određuje se optimalan redoslijed izvršavanja. |

1.3. Cilj i hipoteza rada

U okviru disertacije predlaže se razvoj nove neuronske mreže koja će omogućiti klasifikaciju rasporeda objekata razlikujući objekte po njihovom obliku i položaju u radnom prostoru, te novoga genetskog algoritma koji će omogućiti planiranje robotskog djelovanja za izgradnju prepoznatih struktura zasnovanog na asocijativnom povezivanju neuređenog skupa objekata s prethodno klasificiranim strukturama. Raspored objekata određen je njihovim položajem i međusobnim odnosom u radnom prostoru.

Kao cilj istraživanja postavljena je sljedeća tvrdnja:

- Cilj istraživanja je razvoj i verifikacija upravljačkog modela koji će osposobiti robota za prepoznavanje i izgradnju uređenih prostornih struktura, temeljenog na dvodimenzionalnoj interpretaciji radnog prostora.

Na temelju postavljenog cilja mogu se postaviti sljedeće dvije hipoteze:

- Prepoznavanje prostornih struktura dvodimenzionalnom interpretacijom rasporeda objekata u radnom prostoru moguće je temeljiti na ART klasifikacijskoj neuronskoj mreži, koja povezuje strukture i pripadajuće objekte prema obliku i položaju.
- Izgradnju prostornih struktura moguće je planirati novim genetskim algoritmom zasnovanim na usporednom kreiranju i vrednovanju mogućih strategija robotskog djelovanja.

1.4. Metodologija i plan istraživanja

Cjelokupno istraživanje podijeljeno je na dvije osnovne faze. Kao prvi korak nameće se teorijski dio istraživanja unutar kojeg su postavljeni osnovni zahtjevi tj. u skladu s ciljem istraživanja postavljene su hipoteze na koji će način biti moguće riješiti pojedini problem. U toj fazi istraživanja razvijene su osnovne komponente upravljačkog modela robota koje se očituju u novoj ART mreži te novom genetskom algoritmu planiranja robotskog djelovanja. Te su dvije komponente upravljačkog modela testirane u okviru računalnih simulacija korištenjem virtualnih prostornih struktura te virtualnog okruženja za planiranje robotskog djelovanja. Izvršena je statistička analiza točnosti klasifikacije.

U drugoj je fazi izvršena implementacija razvijenih komponenata te njihovo povezivanje s realnim robotskim sustavom. U tu su svrhu razvijeni robotski upravljački programi koji imaju mogućnost kontinuirane komunikacije s ART mrežom te algoritmom planiranja djelovanja kako bi se za realne prostorne strukture mogao obavljati proces učenja, klasifikacije, prepoznavanja te povezivanja i planiranja. Nadalje, izvršena je verifikacija cjelokupnog upravljačkog modela te je potvrđena njegova primjenjivost u realnom radnom okruženju.

1.4.1. Faza 1. Razvoj i izrada upravljačkog modela

U ovoj fazi razvoj je usmjeren na rješavanje dvaju problema. Prvi je problem učenja i klasifikacije prostornih struktura na temelju dvodimenzionalne interpretacije radnog prostora robota. Drugi problem usmjeren je na razvoj algoritma za planiranje robotskog djelovanja pomoću kojeg će robot biti sposoban izgraditi prepoznate prostorne strukture iz neuređenog skupa poznatih objekata.

Razvoj nove neuronske mreže zasnovane na dvodimenzionalnoj interpretaciji prostora za prepoznavanje i klasifikaciju prostornih struktura temelji se na inkrementalnom učenju. Za robotsko djelovanje u promjenjivoj okolini nije moguće unaprijed odrediti sve djelotvorne strategije. Zbog toga je potrebna mogućnost stalnog prilagođavanja slobodnih parametara neuronske mreže, odnosno odgovarajućih težinskih vrijednosti uslijed procesa interakcije robota s okolinom. Razvijena je nova umjetna neuronska mreža koja se koristi osnovnim principima ART-a. Implementirana je natjecateljska strategija učenja i metoda inicijalne obrade ulaznih podataka pomoću skrivenog sloja predloženog u [65,66]. Razvijena je nova dvorazinska klasifikacija ulaznih struktura pomoću dva skupa međusobno povezanih neuronskih mreža. Takav pristup omogućava pohranjivanje većeg broja prototipnih struktura unutar pojedine klase. Ujedno je na taj način moguće efikasnije pretraživanje postojećih klasa neuronske mreže kod većih vrijednosti parametra budnosti mreže, kao što je objašnjeno u poglavlju 2.3.5. Sa svrhom klasifikacije prostornih struktura prema obliku, ali i prema rasporedu pojedinačnih objekata razvijen je novi podsustav prilagodbe vrijednosti rezonancije [67,68] unutar ART neuronske mreže, opisan u poglavlju 2.3.3. Sličnost dviju prostornih struktura moguće je promatrati na više razina zrnitosti (engl. *level of granularity*) te su u ovom istraživanju predloženi oblik i raspored kao dva osnovna parametra. Razvijeni podsustav određuje mjeru sličnosti dviju prostornih struktura zasnovanu na obliku i rasporedu. Radi povezivanja neuređenog prostornog stanja objekta s postojećim klasificiranim strukturama predložena je nova funkcija preslikavanja. Funkcija preslikavanja kao osnovnu mjeru sličnosti uzima ukupnu kumulativnu sumu svih parcijalnih sličnosti pojedinačnih objekata unutar dviju struktura. Izabire se ono preslikavanje s najvećom kumulativnom sličnosti korištenjem Mađarske metode tj. Kuhn-Munkresova algoritma. Testirana je točnost klasifikacije ART neuronske mreže te je napravljena statistička obrada dobivenih rezultata.

Povezivanje prostornih struktura, osim logičkog, podrazumijeva i realno prostorno preslikavanje stanja. To znači da za svaki povezani par stanja treba generirati odgovarajući plan djelovanja robota kreiranjem uređenog skupa elemenata rada. Zadatak planiranja je pronaći mogući slijed radnji robota kojima će se početno stanje kroz konačan broj koraka preurediti u traženo konačno stanje uz najmanji utrošak robotskog rada. Robotski rad interpretira se kroz utrošeno vrijeme i prijedeni put za obavljanje potrebnih kretnji. Planiranje djelovanja robota kao ulazne parametre uzima dvodimenzionalnu interpretaciju rasporeda objekata u početnom i konačnom stanju. Razvoj algoritma planiranja robotskog djelovanja zasniva se na principima genetskih algoritama i lokalnog pretraživanja (engl. *local search*).

Planiranje djelovanja robota odvija se u dvije slijedne faze. Za generiranje inicijalnog skupa rješenja koristi se novi konstruktivni genetski algoritam. Osnovne značajke koje karakteriziraju razvijeni konstruktivni genetski algoritam su usporedno razvijanje većeg broja rješenja koja tvore inicijalnu populaciju te međusobno uspoređivanje i vrednovanje usporednih strategija djelovanja. Tako najbolje strategije robotskog djelovanja tj. one s najvećom vrijednošću funkcije dobrote imaju najveću vjerojatnost daljnjeg razvoja, dok se najlošije strategije ukidaju. U procesu generiranja inicijalnog skupa rješenja koristi se isključivo operator mutacije položaja objekata. Funkcija dobrote određena je kao funkcija minimizacije ukupnog vremena potrebnog za izvršavanje pojedinog plana. Ukupni prijedeni put robota koreliran je s vremenom te su na taj način sva gibanja izražena u vidu utrošenog vremena. Za drugu fazu planiranja djelovanja razvijen je te primijenjen novi genetski algoritam za optimiranje inicijalne populacije rješenja. Problem optimiranja postavljen je kao problem optimiranja redoslijeda (tj. permutacija) i položaja. Predložene su dvije vrste operatora mutacije i križanja: a) oni koji mijenjaju permutaciju tj. redoslijed objekata u planu, te b) operatori koji mijenjaju prostorni položaj objekata. Nadalje, definirane su određene strategije mutacija i križanja koje omogućavaju konvergenciju rješenja. Napravljena je analiza parametara vjerojatnosti mutacije i križanja sa svrhom pronalaska kombinacije koja daje rješenja najveće dobrote.

1.4.2. Faza 2. Implementacija i verifikacija na laboratorijskoj opremi

U ovoj fazi napravljen je eksperimentalni dio istraživanja implementacijom razvijenog upravljačkog modela na realnom robotskom sustavu. Osnovne cjeline upravljačkog modela očituju se u razvijenoj ART neuronskoj mreži te u genetskom algoritmu planiranja robotskog djelovanja. Eksperimentalni dio istraživanja proveden je u Laboratoriju za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje pri Sveučilištu u Zagrebu. Radi verifikacije upravljačkog modela tj. razvijene ART neuronske mreže te genetskog algoritma planiranja robotskog djelovanja napravljena je implementacija na robotskoj ruci sa šest stupnjeva slobode gibanja te vizijskom sustavu.

Razvijene su te implementirane dvije osnovne funkcionalne cjeline : I. učenje i klasifikacija, II. prepoznavanje prostornih struktura i planiranje odgovarajućeg djelovanja robota.

- I. U ovom dijelu eksperimentalnog istraživanja priređen je niz različitih prostornih struktura sastavljenih od poznatih objekata. Pomoću vizijskog sustava i razvijene

umjetne neuronske mreže omogućeno je učenje te klasifikacija predstavljenih struktura. Početni skup poznatih objekata moguće je dopunjavati novim objektima u kasnijim fazama rada sustava.

- II. Planiranje djelovanja temeljeno je na prepoznavanju, odnosno asocijativnom povezivanju zadanog neuređenog skupa objekata s poznatim naučenim i klasificiranim prostornim strukturama. Implementirana je asocijativna metrika, pri čemu su uzete u obzir najsličnije strukture iz izlaznog sloja neuronske mreže temeljene na vrstama pojedinačnih objekata i njihovih oblika. Najsličnija poznata klasificirana struktura s ulaznim neuređenim skupom objekata iz radnog prostora robota određena je kao konačno stanje. Pomoću razvijenog genetskog algoritma planiranja djelovanja te korištenjem povratne informacije s vizijskog sustava robota generira se plan djelovanja robota.

Razvijen je robotski program koji je u realnim radnim uvjetima sposoban izvesti zadani plan izgradnje prostorne strukture te koji ima mogućnost kontinuirane dvosmjerne komunikacije s centralnim upravljačkim računalom zaduženim za zadatke klasifikacije i povezivanja te generiranja plana djelovanja. Iz preliminarnih istraživanja [69,70] modificirani su te dijelom implementirani adaptivni upravljački algoritmi za djelovanje robota u nestrukturiranoj radnoj okolini.

2.UČENJE I KLASIFIKACIJA PROSTORNIH STRUKTURA

Svijet u kojem živimo je strukturiran [71]. Svaki tehnički sustav ima detaljno određenu strukturu komponenata, pojedini postupak ima strukturu pripadajućih radnji, dok svaki proizvod ili sklop ima zadanu strukturu ugradbenih elemenata. Prema određenim teorijama iz područja neuroznanosti [72,73] objekte koje percipiramo i njihove međusobne odnose moguće je opisati pomoću simbolične interpretacije svijeta, o kojem zapravo nemamo izravno znanje.

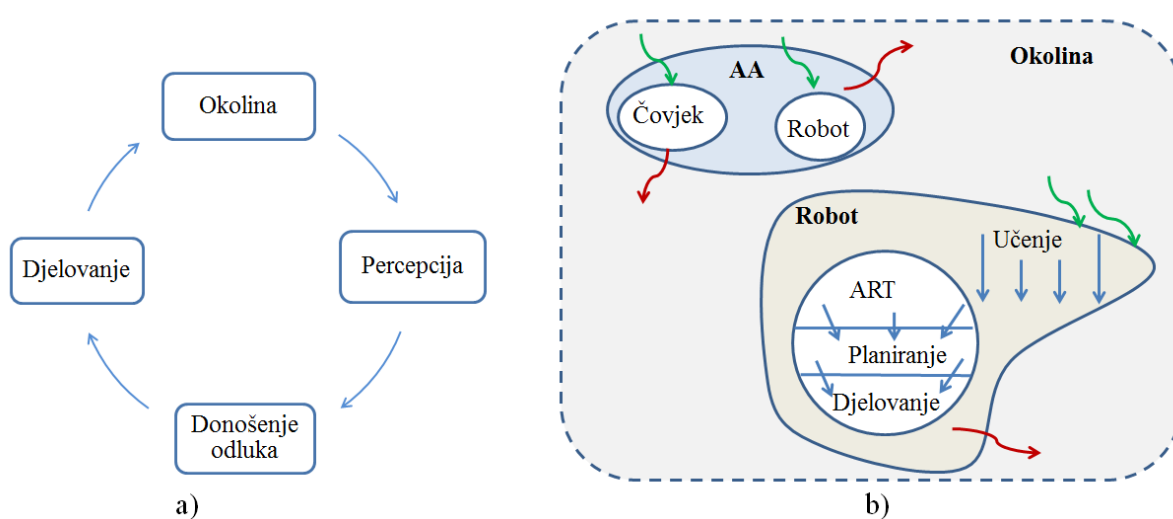
Nadalje, objekti u prostoru najčešće tvore uređene odnose koji imaju pripadajuće značenje. Često je potrebno moći razumjeti veze između proizvoljno raspoređenih ugradbenih elementa u radnom prostoru te osmisliti slijed aktivnosti koje ih dovode u odgovarajući geometrijski odnos. U okviru razmatranja odnosa objekata u radnom prostoru robota pojam prostorne strukture označava dvodimenzionalnu interpretaciju rasporeda poznatih objekata u ravnini koje robot percipira vizijskim sustavom. Učenje prostornih struktura podrazumijeva bilježenje simboličkih odnosa objekata te povezivanje sličnih stanja radnog prostora na temelju stečenih spoznaja. Jedan od mogućih pristupa ovom problemu je proces klasifikacije, grupiranja te razvrstavanja sličnih prostornih stanja odnosno prostornih struktura.

U ovom su poglavlju prikazani osnovni principi na temelju kojih je izgrađena kognitivna komponenta upravljačkog modela robota zadužena za učenje, prepoznavanje i klasifikaciju prostornih struktura te povezivanje neuređenog prostornog stanja objekata s uređenim. Primijenjeni su osnovni principi teorije adaptivne rezonancije (ART) te je razvijena nova konfiguracija ART neuronske mreže koja omogućava povezivanje struktura i pripadajućih objekata prema obliku i položaju u radnom prostoru robota.

2.1. Djelovanje robota u radnoj okolini

Kao što se vidi na slici (Slika 2-1.a), robot skuplja informacije iz okoline, procesira ih, zaključuje te donosi odluke na temelju kojih djeluje na okolinu. U ovom se ciklusu okolina

mijenja te se ponavljaju navedene radnje. Spomenuta interakcija s okolinom nužna je za zaključivanje o određenoj situaciji u kojoj se robot nalazi. Pretpostavlja se da slični ulazni podražaji aktiviraju slična složena djelovanja robota. Robot iz okoline percipira različite objekte u proizvoljnim prostornim odnosima s kojima je potrebno obaviti radnje manipulacije te sklapanja. Prikupljanje informacija iz okoline ograničeno je na vizualnu percepciju objekata odnosno njihovih prostornih struktura. Sličan ulazni uzorak tj. prostorna struktura određena je kada pojedinačni objekti tvore slične prostorne odnose ili kada su oblici dviju struktura srodni. Na slici (Slika 2-1.b) prikazan je cjelokupni pogled motivacijskog scenarija tj. scenarija koji će se ispitivati (engl. *use case*).



Slika 2-1. a) Slijed toka informacija i djelovanja prilikom interakcije robota s okolinom b) Predloženi model sastavljen od triju osnovnih komponenata. Crvene strelice označavaju djelovanja, dok zelene označavaju prikupljanje informacija iz okoline. (AA: autonoman agent).

Scenarij za validaciju (poglavlje 5.) uključuje pojednostavljene objekte s kojima robot može obavljati manipulaciju u radnom prostoru. Objekti mogu tvoriti proizvoljne prostorne strukture, gdje svaki od objekata ima tri SSG-a, dvije translacije te jednu rotaciju, čime je njegov položaj određen u zajedničkoj ravnini.

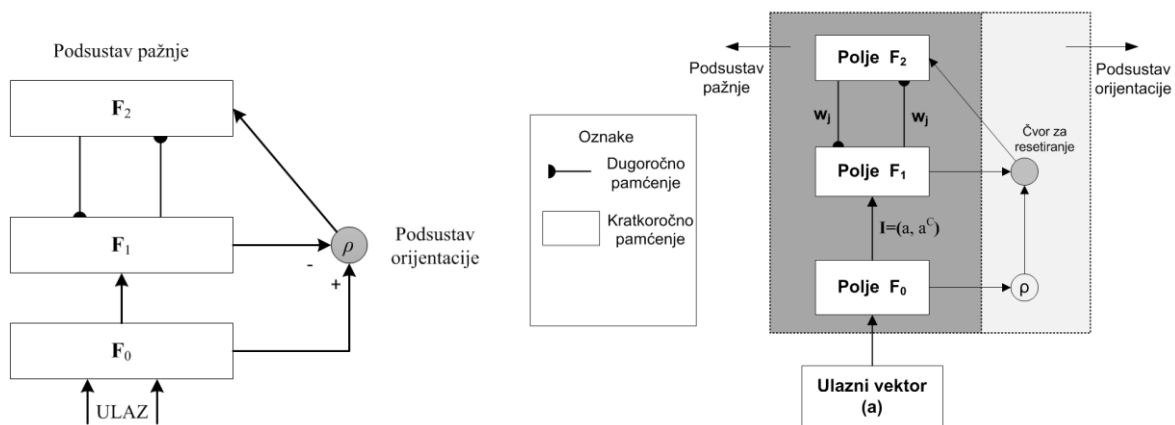
Definicija1. Prostorna struktura je uređeni skup koji se sastoji od najmanje dva objekta koji se nalaze u određenom geometrijskom odnosu. U nastavku teksta upotrijebljen je naziv struktura.

Sa svrhom prepoznavanja, klasifikacije i pohranjivanja novih prostornih struktura, robotski upravljački model koristit će se osnovnim principima teorije adaptivne rezonancije (ART).

Razvoj nove ART mreže omogućit će prepoznavanje i klasifikaciju prostornih struktura. U sljedećem poglavlju slijedi kratki opis ART neuronskih mreža.

2.2. Teorija adaptivne rezonancije

Osnovna karakteristika svih ART neuronskih mreža je mogućnost razlikovanja poznatih uzoraka tj. onih uzoraka koje je mreža već upamtila i nepoznatih uzoraka, tj. onih uzoraka kojima još nije bila izložena. Da bi to bilo moguće, Grossberg [74] je predložio dva podsustava: podsustav pažnje i podsustav orijentacije (Slika 2-2.).



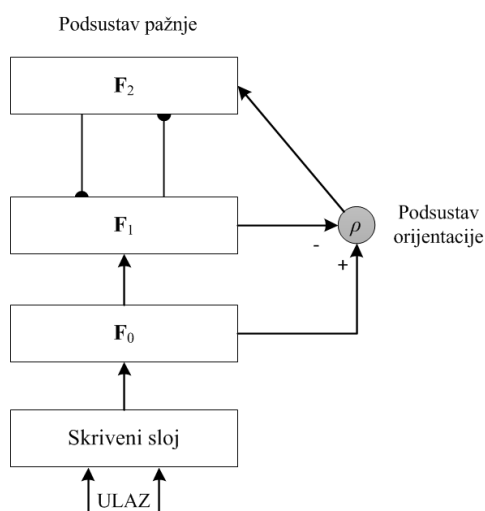
Slika 2-2. Arhitektura Fuzzy ART neuronske mreže.

Podsustav pažnje obrađuje poznate uzorke te se sastoji od dva sloja neurona koji su međusobno potpuno recipročno povezani. Prvi sloj (F1) registrira uzorak koji dolazi iz osjetnih organa ili neke druge neuronske mreže te ga šalje u sloj F2. U sloju F2 pomoću funkcije sličnosti, koja je različita u različitim varijantama ART algoritama [37,75,76] računa se preklapanje ulazne strukture s postojećim klasama tj. njihovim težinskim vrijednostima. Ulazna struktura označava lokalizirani uređeni skup objekata u radnom prostoru robota. U procesu učenja mreže svaka se struktura klasificira te pojedina klasa u ART mreži tj. njezine težinske vrijednosti sadrže informacije o većem broju struktura, što će biti opisano u poglavlju 2.3. U sloju F2 postoje inhibitorne veze između neurona te nakon određenog vremena samo jedan neuron u sloju F2 ostaje aktivan predstavljajući klasu za predstavljani uzorak. Nadalje, računa se vrijednost funkcije preklapanja, pa ako je ta vrijednost dovoljno velika, znači da je mreža prepoznala uzorak. U tom slučaju, neuroni iz oba sloja međusobno pojačavaju svoju aktivnost i mreža ulazi u stanje rezonancije. U suprotnom slučaju, kada je preklapanje malo, mreža se nalazi pred novim uzorkom, pa se aktivira sustav za orijentaciju. On resetira aktivne F2 neurone, odnosno onemogućuje njihovo ponovno aktiviranje nakon resetiranja. Putem

lateralne inhibicije (odnosno principa „pobjednik uzima sve”) traži se neuron koji ima najveću aktivnost. To ne može biti isti neuron koji je prethodno resetiran jer ga sustav za orijentaciju inhibira i onemogućuje mu ponovnu aktivaciju. Novi F2 neuron šalje signale u sloj F1 i cijeli se postupak ponavlja. Na taj način mreža pretražuje postojeće klase kako bi pronašla onaj uzorak koji se najviše preklapa s uzorkom registriranim u sloju F1. U slučaju kada nijedan F2 neuron ne udovoljava kriteriju inicijalizira se novi F2 neuron koji izravno pohranjuje uzorak iz okoline. Prilikom mijenjanja sinaptičkih efikasnosti F1 i F2 neuroni postaju bolje povezani te dolazi do međusobnog pojačavanja aktivnosti, tj. rezonancije [19]. Budući da je rezonancija posljedica adaptivnih promjena na sinapsama, otud dolazi i naziv Teorija adaptivne rezonancije. Kriterij koji određuje koliko se registrirani uzorak u sloju F1 mora poklapati s očitanim uzorkom iz sloja F2 određuje parametar budnosti mreže.

2.2.1. AFS neuronska mreža

Radi lakšeg razumijevanja svih aspekata teorije i njezine evaluacije prikazan je primjer rada AFS (engl. *adaptive fuzzy shadow network*) [66] neuronske mreže koja se temelji na Fuzzy ART algoritmu te ART1 algoritmu, a poslužit će kao osnova za kreiranje nove neuronske mreže i nove arhitekture u okviru ovog doktorskog rada. Na slici (Slika 2-3.) prikazana je arhitektura AFS neuronske mreže koja sadrži jedan dodatni skriveni sloj u odnosu na standardni FuzzyART algoritam.



Slika 2-3. Arhitektura AFS neuronske mreže.

Svrha skrivenog sloja je dodavanje „sjena” tj. proširenje rubnih piksela tako da vrijednost piksela proporcionalno opada s udaljavanjem od ruba objekta. Osnovni parametri AFS neuronske mreže su: parametar učenja β , parametar budnosti mreže φ , broj izlaznih neurona

N , funkcija sličnosti $T(\mathbf{X}, \mathbf{W}_k)$, matrice težinskih vrijednosti \mathbf{W}_k , dok su ulazni uzorci označeni sa \mathbf{X} .

Algoritam 1. AFS algoritam

Korak 1. Inicijalizacija

$\beta \in [0,1]$, postavi vrijednost parametra učenja

$\varphi \in [0,1]$, postavi vrijednost parametra budnosti mreže (parametar sličnosti)

$N=0$, postavi broj izlaznih neurona (klasa).

Korak 2. Registracija ulaznog uzorka

Registracija ulaznog uzorka \mathbf{X} iz okoline. \mathbf{X} predstavlja binarnu matricu.

Korak 3. Skriveni sloj

Izmijeni ulazni uzorak \mathbf{X} u \mathbf{X}^* koristeći se izrazom:

$$\mathbf{X}_{ij}^* = \begin{cases} \mathbf{X}_{ij} & \forall \mathbf{X}_{ij} \neq 0 \\ \mathbf{X}_{ij} \frac{\alpha}{l_1 + l_2}, & \forall l_1 + l_2 < l_{\max} \end{cases} \quad \begin{matrix} l_1 = |i^* - i| \\ l_2 = |j^* - j| \end{matrix} \quad (2.1)$$

Korak 4. Funkcija sličnosti

Za svaki izlazni neuron računaj funkciju sličnosti prema izrazu:

$$T_k = \frac{\|\mathbf{X}^* \wedge \mathbf{W}_k\|}{\|\mathbf{X}^* \vee \mathbf{W}_k\|} \quad (2.2)$$

Norma $\|\cdot\|$ određena je izrazom $\|\mathbf{A}\| = \sum_i \sum_j \mathbf{A}_{ij}$.

Korak 5. Najsličnija klasa

Izaberi izlaznu klasu K koja ima najveću vrijednost sličnosti $T_K = \max_k \{T_k\}$.

Korak 6. Rezonancija

Ako je $T_K > \varphi$, nastupa rezonancija te dolazi do promjene težinskih vrijednosti prema izrazu:

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \beta(\mathbf{W}^{(t-1)} - \mathbf{X}^*) \quad (2.3)$$

Ako je $T_K \leq \varphi$, idi na sljedeći korak

Korak 7. Inicijalizacija nove izlazne klase

Povećaj broj izlaznih neurona na $N = N+1$. Postavi težinske vrijednosti novog neurona na $\mathbf{W}_N = \mathbf{X}^*$. Idi na korak 2.

2.3. Razvoj nove ART neuronske mreže

Prostorna struktura može se odrediti na dva stupnja detalja (zrnatosti). Prvi stupanj označava oblik cjelokupne strukture bez utjecaja individualnih objekata unutar strukture. Drugi, detaljniji stupanj prepoznavanja uključuje prikupljanje informacija o individualnim objektima unutar prostorne strukture kao što su pozicija i orijentacija u referentnom koordinatnom sustavu. Prostorna struktura prepoznaje se u dvodimenzionalnom radnom prostoru, gdje se poznati objekti mogu nalaziti u različitim prostornim odnosima.

Neka $\mathbf{I}_g = (I_{g_{11}}, I_{g_{12}}, \dots, I_{g_{mm}})$, $g \in [1, Np]$ označava lokaliziranu ulaznu strukturu, gdje je $I_{g_{ij}} \in [0,1]$. Svaki piksel na slici (dimenzija slike je 300×300 piksela) koji odgovara određenom objektu poprima vrijednost 1, dok svi ostali pikseli poprimaju vrijednost 0 tvoreći na taj način inicijalnu binarnu matricu. Neka \mathbf{W}_j označava matrice težinskih vrijednosti. Osnovni parametri razvijene ART mreže uključuju stopu učenja $\beta_i \in [0,1]$ te parametar budnosti mreže $\varphi_i \in [0,1]$, za $i = [1, 2]$. Postavljanjem parametra budnosti mreže na visoke vrijednosti ($\varphi_i > 0,9$), mreža generira vrlo specifične klase. Postavljajući parametar budnosti mreže na vrijednost 1, svaka se ulazna struktura kategorizira unutar nove klase. Analogno tome, postavljajući φ na malu vrijednost ($< 0,1$) mreža generira generalne klase unutar kojih se nalazi velik broj različitih struktura. Sličan je ekstrem kada je $\varphi = 0$, tada se sve ulazne matrice registriraju unutar iste klase mreže te mreža može imati maksimalno jednu izlaznu klasu.

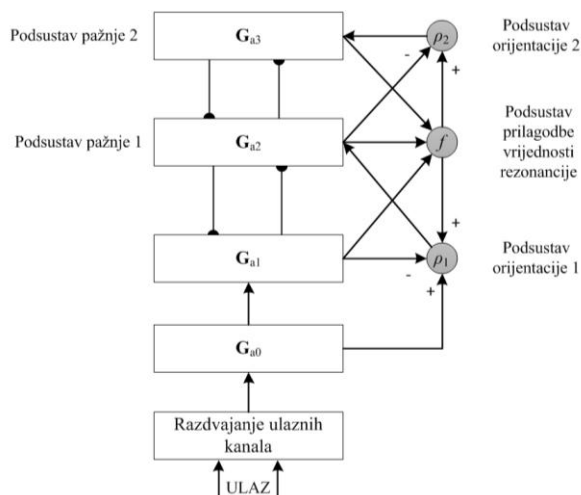
Generalno gledajući, kada je budnost mreže mala, rijetko će dolaziti do aktiviranja podsustava za orijentaciju, te će mreža moći klasificirati u zasebne klase samo one uzorke među kojima postoji velika razlika (sposobnost razlučivanja mreže), dok će slične uzorke tretirati kao primjere iste generalne klase. S druge strane, ako je budnost visoka, mreža će postati osjetljiva na male razlike u uzorcima [19].

2.3.1. Arhitektura ART mreže

Na slici (Slika 2-4.) prikazana je arhitektura razvijene ART mreže. Uz standardne podsustave pažnje i orijentacije dodatno su implementirani podsustavi pažnje i orijentacije te je razvijen

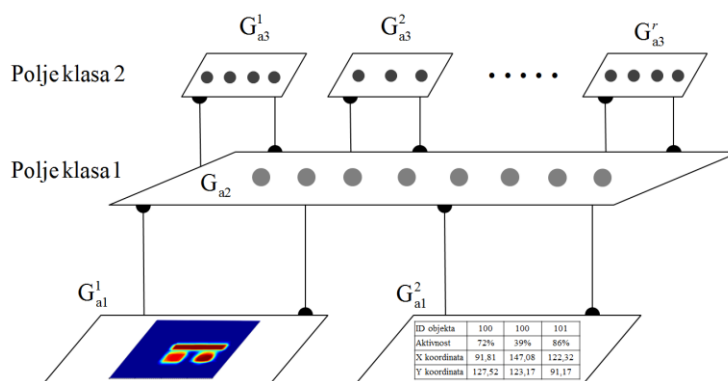
2. Učenje i klasifikacija prostornih struktura

novi podsustav prilagodbe vrijednosti rezonancije. Zbog dvorazinske konfiguracije ART mreže potrebna su dva sustava pažnje i dva podsustava orijentacije. Nadalje, sa svrhom klasifikacije prostornih struktura prema obliku, ali i prema rasporedu pojedinačnih objekata, razvijen je novi podsustav prilagodbe vrijednosti rezonancije.



Slika 2-4. Prikaz svih podsustava unutar razvijene ART mreže.

Na slici (Slika 2-5.) prikazani su ulazni kanali u mrežu. S lijeve strane nalazi se lokalizirana prostorna struktura označena s G_{a1}^1 , dok se s desne strane nalazi matrica objekata označena s G_{a1}^2 . Matrica objekata ima pet redaka dok broj stupaca odgovara broju objekata. Za svaki od objekata zapisan je položaj (x, y) , orijentacija i aktivnost. Pretragom postojećih klasa, prvo u polju G_{a2} te potom u pripadajućem polju G_{a3} izabire se klasa koja ima najveću sličnost sa ulaznom strukturom prema izrazu (2.6). Ako takve klase nema, generiraju se nove izlazne klase u oba polja ili samo u G_{a2} polju.



Slika 2-5. Dvorazinska arhitektura razvijene ART mreže.

Zbog svoje mrežaste i razgranate strukture, razvijena ART neuronska mreža nazvana je ARTgrid [68].

2.3.2. Postupak učenja ART mreže

Funkcija učenja koja se primjenjuje u sklopu razvijene ART mreže prikazana je u (2.4). Jednadžba (2.5) označava originalnu funkciju učenja iz Fuzzy ART algoritma [37], gdje težinske vrijednosti mogu isključivo smanjivati svoju aktivnost kako bi se mreža nakon određenog broja iteracija u potpunosti stabilizirala. Povećanje težinskih vrijednosti nije moguće, odnosno određene karakteristike ulaznih uzoraka koje se u nekoliko iteracija predstavljaju mreži ne mogu biti naučene ako nisu postojale unutar inicijalne strukture. Funkcija iz (2.4) omogućava da težinske vrijednosti povećavaju ili smanjuju svoju aktivnost proporcionalno postavljenom koeficijentu brzine učenja. Ako je ulazna struktura \mathbf{I} identična težinskim vrijednostima, tada ne dolazi do promjene. Kada mreža nema inicijalizirane izlazne klase, početna se struktura izravno kopira u mrežu postavljanjem stope učenja na vrijednost 1 (engl. *fast learning*). Inicijalizacija mreže prikazana je algoritmom (Algoritam 2-1.).

$$\mathbf{W}_j^{(t)} = \beta(\mathbf{I}) + (1-\beta)\mathbf{W}_j^{(t-1)} \quad (2.4)$$

$$\mathbf{W}_j^{(t)} = \beta(\mathbf{I} \wedge \mathbf{W}_j^{(t-1)}) + (1-\beta)\mathbf{W}_j^{(t-1)} \quad (2.5)$$

Algoritam 2-1. ARTgrid inicijalizacija.

Algoritam : ART_inicijalizacija

Ulaz: \mathbf{I}

Izlaz: \mathbf{W}

- 1 \Za ulaznu strukturu \mathbf{I} generiraj inicijalnu matricu težinskih vrijednosti \mathbf{W}
 - 2 $\mathbf{W}\{1\}(1) \leftarrow \mathbf{I}$
 - 3 $\mathbf{W}\{1\}\{1\}(1) \leftarrow \mathbf{I}$
 - 4 MTO \leftarrow konvertiraj_I(\mathbf{I})
 - 5 $\mathbf{W}\{1\}(2) \leftarrow$ MTO
 - 6 $\mathbf{W}\{1\}\{1\}(2) \leftarrow$ MTO
-

Nakon inicijalizacije mreže za svaki novi ulazni uzorak računa se funkcija odabira (engl. *choice function*) za svaku G_{a2} izlaznu klasu prema izrazu:

$$T_{i1} = \frac{\|\mathbf{I} \wedge \mathbf{W}_{a2}^i\|}{\|\mathbf{I} \vee \mathbf{W}_{a2}^i\|}, \quad i = 1, 2, \dots, N_1 \quad (2.6)$$

gdje N_1 označava trenutni broj G_{a2} klasa. Za matrice \mathbf{A} i \mathbf{B} neizraziti operator i određen je s $(\mathbf{A} \wedge \mathbf{B})_{ij} \equiv \min(A_{ij}, B_{ij})$, dok je neizraziti operator *ili* određen s $(\mathbf{A} \vee \mathbf{B})_{ij} \equiv \max(A_{ij}, B_{ij})$.

Izlazni neuron s najvećom vrijednosti funkcije odabira aktivira se prema izrazu:

$$T_{ji} = \max \{T_{ji}, j = 1, \dots, N_i, i = 1, 2\} \quad (2.7)$$

Slijedi provjera testa rezonancije (engl. *resonance test*) prema funkciji preklapanja (engl. *match function*):

$$m_{iJ} = \frac{\|\mathbf{I} \wedge \mathbf{W}_{ai}^j\|}{\|\mathbf{I}\|} \quad (2.8)$$

Vrijednost m_{iJ} računa se za oblik strukture, dok je za računanje ukupne vrijednosti rezonancije koja će se usporediti s parametrom budnosti mreže potrebno uzeti u obzir parametre pojedinačnih objekata unutar strukture.

2.3.3. Prepoznavanje struktura prema obliku i rasporedu

Inicijalna vrijednost rezonancije m_{iJ} može se smanjiti ili povećati uzimajući u obzir sličnosti ulazne strukture \mathbf{I} te težinskih vrijednosti \mathbf{W} na razini pojedinačnih objekata. Predložena su tri osnova parametra za određivanje sličnosti dviju prostornih struktura:

- razlika broja jedinstvenih objekata u \mathbf{I} i \mathbf{W}
- razlika relativnog položaja objekata u novoj i postojećoj strukturi
- razlika u ukupnom broju objekata: a) višak objekata, b) manjak objekata.

Predlaže se novi podsustav prilagodbe vrijednosti rezonancije unutar ART mreže koji je zadužen za promjenu osnovne vrijednosti rezonancije m_{iJ} temeljen prethodna tri parametra. Svaki neuron u G_{a2} i G_{a3} nivou sadrži matricu objekata (MTO) s informacijama o položaju (x , y), orijentaciji (r) te aktivnosti (a) pojedinog objekta u strukturi. Položaj i orijentacija pojedinog objekta odnose se na centar mase objekta u referentnom koordinatnom sustavu. Aktivnost objekta a za sve je objekte inicijalizirana na 1. Uslijed procesa učenja tj. promjene težinskih vrijednosti mreže, koji se odvija prema izrazu (2.4), aktivnost pojedinih objekata može se povećati ili smanjiti s obzirom na nadolazeće strukture i njima pripadajuće objekte. Uspoređivanjem i računanjem pripadajućih vrijednosti preklapanja između matrice objekata ulazne strukture \mathbf{I} te težinskih vrijednosti \mathbf{W} izabranog neurona iz (2.7) traži se postotak sličnosti između ulazne strukture i postojeće klasificirane strukture. Prvi parametar koji se računa je r_1 :

$$r_1 = \frac{j_obj(\{I\})}{j_obj(\{I, W_T^1\})} \quad (2.9)$$

Funkcija $j_obj()$ računa broj svih jedinstvenih objekata ulaznog skupa matrice objekata. Parametar r_2 označava sličnost dviju struktura na temelju razlike relativne udaljenosti d_i identičnih objekata. U tu svrhu uvodi se prag udaljenosti centara mase dvaju objekata. Parametar r_2 računa se iz udaljenosti dvaju objekata d prema izrazu (2.10). Donji prag određen je kao d^{\min} te je inicijalno postavljen kao četvrtina najmanje dimenzije najmanjeg objekta, dok je gornji prag d^{\max} postavljen kao pola dimenzije najmanjeg objekta. Parametar r_2 računa se prema sljedećem izrazu:

$$r_2 = \begin{cases} 1 & \forall d \leq d^{\min} \\ \frac{2(d^{\min} - d)}{d^{\min}} & \forall d^{\min} \leq d \leq d^{\max} \\ 0 & \forall d \geq d^{\max} \end{cases} \quad (2.10)$$

Parametri r_3 i r_4 određuju mjeru razlike ukupnog broja objekata u \mathbf{I} u odnosu na \mathbf{W} . Računaju se prema izrazu:

$$(r_3, r_4) = \begin{cases} t = br_obj(\mathbf{W}_i^j) - br_obj(\mathbf{I}) \quad , i = 1, 2, \dots, N_j \quad j = 1, 2 \\ r_3 = -t, \quad r_4 = 0 \quad \forall t \leq 0 \\ r_3 = 0, \quad r_4 = t \quad \forall t > 0 \end{cases} \quad (2.11)$$

Funkcija $br_obj()$ računa ukupan broj lokaliziranih objekata iz matrice objekata pripadajuće strukture. Formira se vektor $\mathbf{r}_i = (r_1, r_2, r_3, r_4)$ koji se skalarno množi vektorom težinskih vrijednosti prilagodbe rezonancije \mathbf{v}_i . Težinske vrijednosti postavljene su inicijalno na 0,5. Skalarni umnožak $g_i = \mathbf{r}_i \cdot \mathbf{v}_i$ pribraja se prethodno izračunanoj vrijednosti rezonancije kako bi se izračunala završna vrijednost za pojedinu klasu uzimajući u obzir sličnosti te razlike na temelju individualnih objekata u ulaznoj strukturi \mathbf{I} te postojećih klasa \mathcal{W} .

$$m_{iJ} = m_{iJ} g_i \quad (2.12)$$

2. Učenje i klasifikacija prostornih struktura

Cjelokupan proces učenja novih klasa te ažuriranja i pretraživanja postojećih klasa u dvorazinskoj konfiguraciji mreže prikazan je algoritmom (Algoritam 2-2.).

Algoritam 2-2. Dvorazinski ART s podsustavom prilagodbe vrijednosti rezonancije.

Algoritam : Dvorazinski ART

```
1  Ulaz:  $\mathbf{I}, \mathcal{W}, \beta_1, \beta_2, \varphi_1, \varphi_2$ 
2  Izlaz: klasa_rb1, klasa_rb2,  $m_{ij}$ 
3  Za ulaznu strukturu  $\mathbf{I}$  pronađi najbližnju klasu u  $\mathcal{W}\{\text{rb1}\}\{\text{rb2}\}$  ili generiraj novu klasu
4  za svaki  $W_i \in \mathcal{W}_{\text{nivo}_1}$ 
5       $T_i \leftarrow$  funkcija odabira prema (2.6)
6  kraj
7  za svaki  $T_i \in T: T_i \leftarrow \max(T)$ 
8       $m_i \leftarrow$  vrijednost preklapanja prema izrazu (2.8) // Inicijalno preklapanje samo
9      strukture.
10      $m_i \leftarrow$  računaj RAS prema izrazu (2.12) // Adaptacija vrijednosti
11     rezonancije na temelju rasporeda objekata.
12     ako  $m_i > \varphi_1$  onda
13         neuron  $i$  je pobjednik,  $\text{klasa\_rb1} \leftarrow i$ 
14         učenje  $\mathcal{W}(i)$  prema izrazu (2.4) sa  $\beta_1$ 
15         za svaki  $W_j \in \mathcal{W}(i) \rightarrow$  računaj  $T_{ij}$  prema izrazu (2.6)
16         za svaki  $T_{ij}$  izaberi  $\max(T_{ij}), m_{ij} \leftarrow$  prema (2.7)
17              $m_{ij} \leftarrow$  računaj RAS prema izrazu (2.8) i (2.12) // Adaptacija vrijednosti
18             rezon.
19             ako  $m_{ij} > \varphi_2$  onda
20                 učenje  $\mathcal{W}\{i\}\{j\}$  prema izrazu (3.4) s  $\beta_2$ 
21                 prekid_programa
22             inače
23                  $T_{ij} \leftarrow (-1)$  // Inhibiraj neuron  $\mathcal{W}\{i\}\{j\}$ .
24             kraj
25         kraj
26     Ako  $\max(T_{ij}) = -1$  onda
27          $\mathcal{W}\{i\}\{j+1\} \leftarrow \mathbf{I}$  // Kopiranje  $\mathbf{I}$  u novu nivo2 klasu.
28     kraj
29     inače
30          $T_i \leftarrow (-1)$  // Inhibiraj neuron  $\mathcal{W}\{i\}$ .
31     kraj
32 kraj
```

2.3.4. Određivanje ulaznog redoslijeda struktura

Kod ART mreža jedan od osnovnih problema je osjetljivost mreže na različit redoslijed uzoraka prilikom procesa učenja mreže. Algoritam određivanja redoslijeda ulaznih struktura [66,68] uzima u obzir isključivo ulaznu matricu $\mathbf{I}_i(G_{a1}^1)$ pojedine strukture bez analize matrice objekata (G_{a1}^2). Određuje se relativna razlika pojedine ulazne strukture sa svim preostalim strukturama te se struktura s najvećom vrijednošću razlike izabire kao sljedeća struktura koja će se aplicirati. Kao prva inicijalna ulazna struktura uzima se struktura s najmanjom te gustoćom prema izrazu:

$$d_1 = \min_p \left\{ \sum_{i=1}^m \sum_{j=1}^n \mathbf{I}_{p_{ij}}, \quad p = 1, \dots, N \right\}, \quad (2.13)$$

dok se kao druga ulazna struktura uzima struktura s najvećom gustoćom prema izrazu:

$$d_2 = \max_p \left\{ \sum_{i=1}^m \sum_{j=1}^n \mathbf{I}_{p_{ij}}, \quad p = 1, \dots, N \right\}. \quad (2.14)$$

Parametar $C_{\max} \in [2, N]$ određuje ukupan broj struktura za koje se računa relativna razlika. Eksperimentalno je utvrđen raspon parametra $C_{\max} = [0,15 \times N - 0,20 \times N]$. Ulazna struktura \mathbf{I}_{d_1} aplicira se prva, dok se ulazna struktura \mathbf{I}_{d_2} aplicira kao druga u procesu učenja ART mreže, pri čemu je $d_1, d_2 \in [1, N]$. Sljedeća ulazna matrica (primjer ulazne matrice prikazan je na slikama 2-6. i 2-7.) koja se izabire je ona koja ima najmanju relativnu sličnost sa svim prethodnim matricama prema izrazu:

$$\mathbf{S}_p = \frac{\sum_{i=1}^m \sum_{j=1}^n \sum_{l=1}^C (\mathbf{I}_{p_{ij}} \mathbf{Y}_{l_{ij}})}{C \sum_{i=1}^m \sum_{j=1}^n \mathbf{I}_{p_{ij}}}, \quad (2.15)$$

pri čemu je skup izabranih matrica označen s $\mathbf{Y}_s = (Y_{s_{11}}, Y_{s_{12}}, \dots, Y_{s_{mm}})$, $s \in [1, C]$, dok je sa C označen trenutni broj izabranih matrica. Matrica $\mathbf{S}_{\min} = \min \{ \mathbf{S}_p, p = 1, \dots, N - C \}$, koja označava matricu s najmanjom relativnom sličnosti u odnosu na preostale neizabrane matrice,

2. Učenje i klasifikacija prostornih struktura

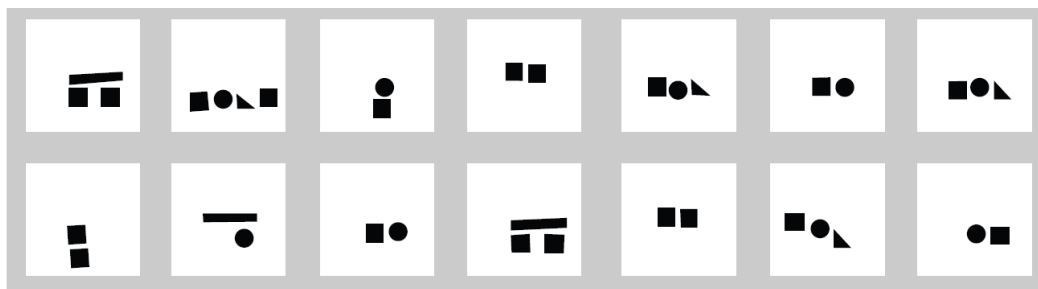
izabire se kao sljedeća. Korak (2.15) ponavlja se dok se ne zadovolji uvjet $C = C_{\max}$. Primjer ulaznih matrica prikazan je na slici (Slika 2-6.).

Algoritam 2-3. *Generator slučajnih struktura.*

Algoritam: Generator_struktura

```
1  Ulaz:  $\mathcal{U}, b, p_x, p_y, p_\varphi, \mu_x, \sigma_x, \mu_y, \sigma_y, \mu_\varphi, \sigma_\varphi$ 
2  Izlaz:  $\mathcal{O}$ 
   // generiraj slučajne prostorne strukture prema inicijalnim podacima
3  iz  $\mathcal{U}$ 
4  // slučajne prostorne strukture spremi u skup za učenje ART mreže  $\rightarrow \mathcal{O}$ 
5  za_svaki element iz  $\mathcal{U}, j = 0$ 
6      za_svaki  $j < b, j++$ 
7          Objekti =  $\mathcal{U}(j)$ 
8          promijeni vrstu svakog objekta s vjerojatnosti  $P(\text{vrsta}) = p_x$ 
9          promijeni položaj svakog objekta s vjerojatnosti  $P(\Delta x, \Delta y) = p_y \dots$ 
10         ... prema normalnoj razdiobi s parametrima  $N(\mu_x, \sigma_x)$  i  $N(\mu_y, \sigma_y)$ 
11         promijeni kut zakreta svakog objekta s vjerojatnosti  $P(\Delta \varphi) = p_\varphi \dots$ 
12         ... prema normalnoj razdiobi  $N(\mu_\varphi, \sigma_\varphi)$ 
13         zapiši sve vrijednosti u Objekti
14         O  $\leftarrow$  generiraj_strukturu(Objekti)
15         status  $\leftarrow$  evaluacija_kolizije(O)
16         ako status = 1, vrati se na korak 8
17         k++,  $\mathcal{O}(k) \leftarrow \mathbf{O}$ 
18     kraj
19 kraj
```

Algoritam (Algoritam 2-3.) služi za generiranje većeg broja slučajnih struktura za verifikaciju ART mreže. U prvom se koraku inicijalno određuju vrste objekata u pojedinoj strukturi te početni položaj i orijentacija objekata. Za svaki od objekata zadaju se vrijednosti vjerojatnosti koje utječu na promjenu vrste, položaja i orijentacije, čime se generiraju raznovrsne strukture. Ove vrijednosti zadaje korisnik kako bi se omogućilo generiranje većeg broja raznovrsnih prostornih struktura. Oznaka $P(A)$ određuje vjerojatnost događaja A , dok je $N(\mu, \sigma)$ označena funkcija Gaussove razdiobe s očekivanjem μ i standardnom devijacijom σ . Na slici (Slika 2-6.) prikazano je 14 slučajno generiranih struktura pomoću algoritma (Algoritam 2-3.). Kako bi se mogla verifikirati te kvantificirati točnost i pouzdanost klasifikacije ART mreže, potrebno je generirati skup za učenje koji se sastoji od raznovrsnih prostornih struktura.



Slika 2-6. *Primjer 14 slučajno generiranih prostornih struktura.*

Na slici (Slika 2-7.a) prikazan je primjer jedne izlazne klase koja je dobivena procesom apliciranja 12 struktura prikazanih na slici (Slika 2-7.b).

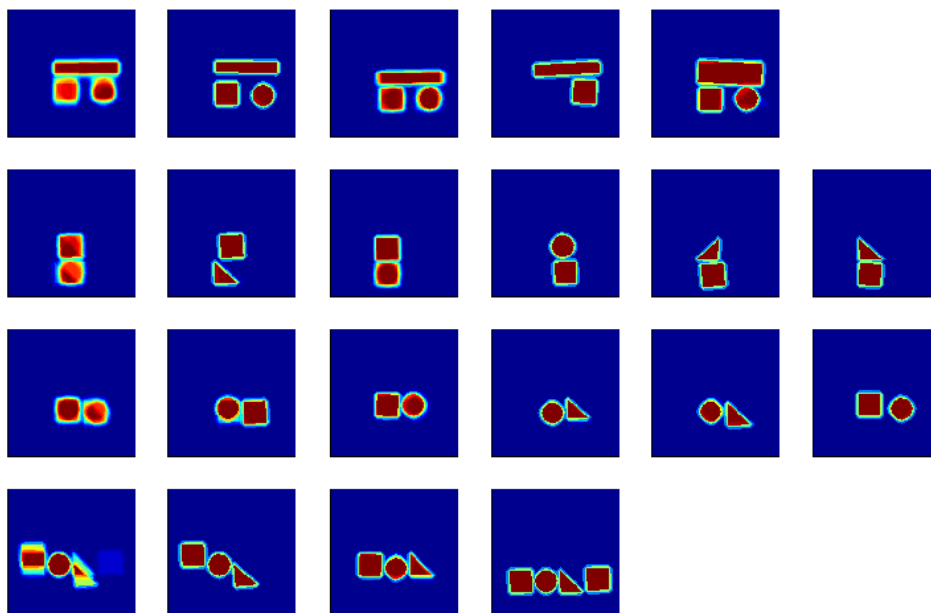


a)

b)

Slika 2-7. *a) Primjer izlazne klase karakterizirane matricom 200 x 200 b) 12 prostornih struktura koje su primijenjene na klasu prikazanu pod a).*

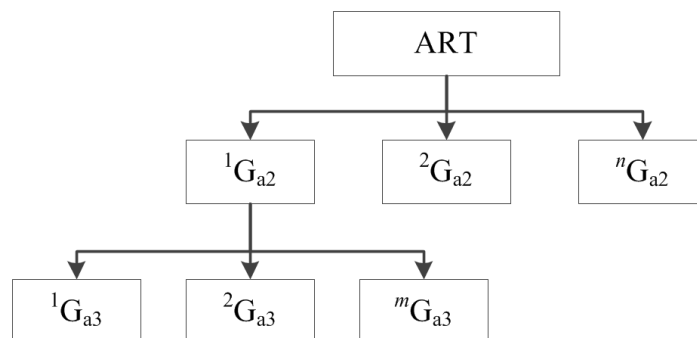
Primjer klasifikacije u 4 izlazne klase većeg broja različitih struktura generiranih algoritmom (Algoritam 2-3.) prikazano je na slici (Slika 2-8.). Prvi stupac s lijeva označava klase u polju klasa 1, dok strukture u preostalim stupcima označavaju polje klasa 2 prema slici (Slika 2-5.).



Slika 2-8. Primjer klasifikacije u 4 izlazne G_{a1} klase (prikazane u 1. stupcu s lijeva).

2.3.5. Karakteristika učenja ART mreže

Jedna od karakteristika koju se željelo poboljšati je brzina pretraživanja postojećih klasa te brzina procesa učenja. U tu je svrhu razvijena dvorazinska klasifikacija čija je zadaća podjela postojećih klasa u dvije razine G_{a2} i G_{a3} , kao što je prikazano na slikama (Slika 2-8.) i (Slika 2-9.).



Slika 2-9. Arhitektura predložene ART mreže.

Osnovni cilj podjele postupka pretraživanja postojećih klasa u dva nivoa postavljanjem parametra budnosti mreže $\varphi_1 < \varphi_2$, je generiranje općenitijih G_{a2} klasa, koje tada sadrže određen broj specifičnih G_{a3} klasa. Na taj je način proces učenja, a analogno tome i proces pretraživanja postojećih G_{a2} i G_{a3} klasa, razgranat. U postojećim ART arhitekturama kao što je Fuzzy ART [37] ili u modificiranim arhitekturama kao što su Fusion ART [39], CobART [45], AFS [66] proces pretraživanja klasa i učenja je slijedan te je potrebno ispitati sve klase u

2. Učenje i klasifikacija prostornih struktura

svrhu evaluacije izraza za funkciju odabira (2.6). U algoritmu (Algoritam 2-4.) prikazan je proces učenja i pretraživanja klasa u AFS mreži.

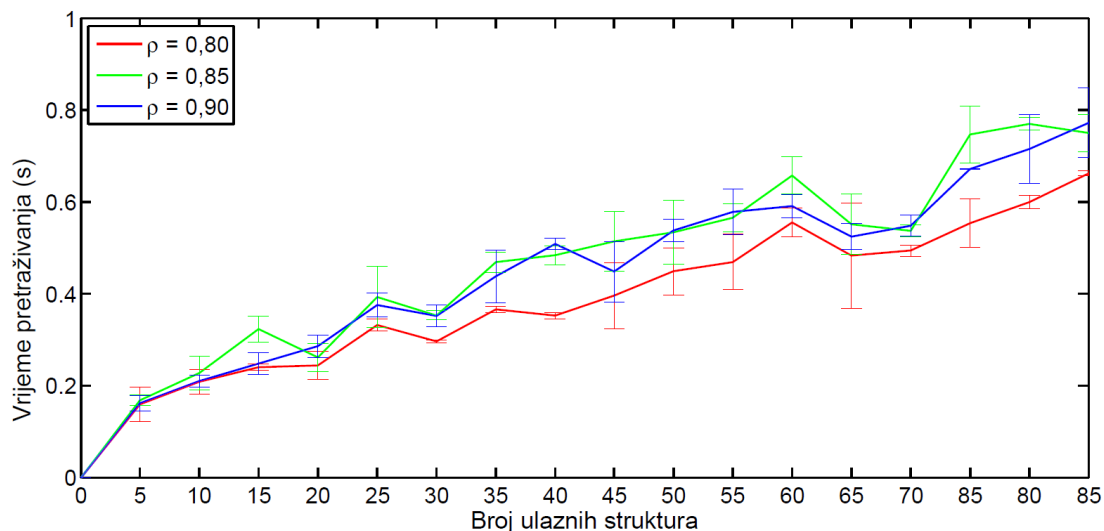
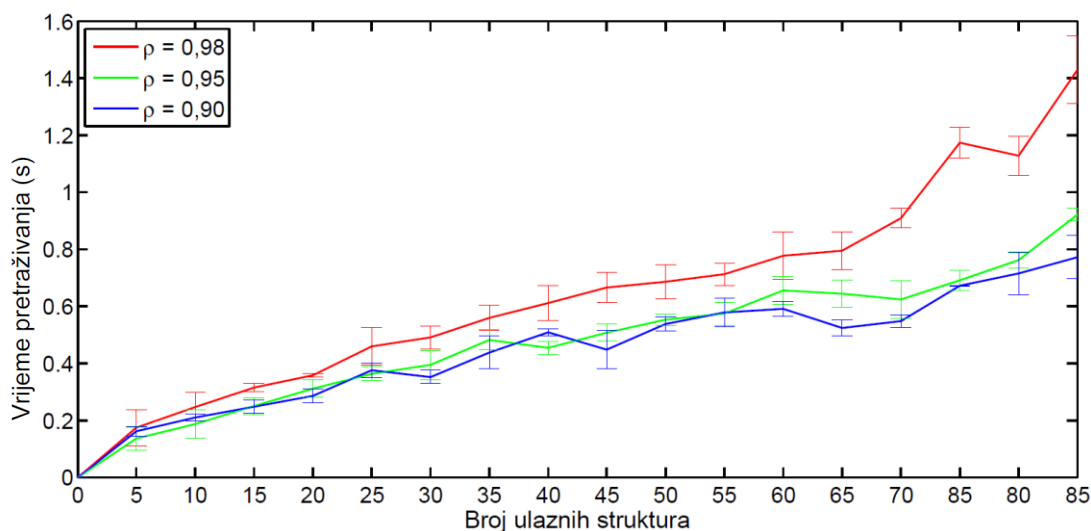
Algoritam 2-4. *Pretraživanje klasa i učenje mreže – AFS.*

Algoritam : Pretraživanje_klasa_AFS

```

Ulaz:  $I, \mathcal{W}, \beta, \varphi$ 
Izlaz: pobjednicka_klasa
1  \Za ulaznu strukturu  $I$  traži najbližnju klasu u  $\mathcal{W}$  ili generiraj novu
   klasu  $W_n$ .
2  za svaki  $W_i \in \mathcal{W}, i++$ 
3      izračunaj funkciju sličnosti prema izrazu (2.2)  $\rightarrow T_i$ 
4      Izaberi  $\mathcal{W}(i)$  koji ima  $\max(T)$ 
5      ako  $T_i > \varphi$  onda
6          //Neuroni je pobjednik.
7          učenje pobjedničke klase  $\mathcal{W}(i)$  prema izrazu (2.3)
8          pobjednicka_klasa  $\leftarrow i$ , prekid_programa
9      inače
10         //inhibiraj neuron  $\mathcal{W}(i)$ , onemogućeno natjecanje neurona  $i$ .
11          $T_i \leftarrow (-1)$ 
12     kraj
13 kraj
14 ako  $\max(T_i) = (-1)$  onda //Svi neuroni inhibirani.
15      $\mathcal{W}(i+1) \leftarrow I$  // Napravi novu izlaznu klasu.
16     pobjednicka_klasa  $\leftarrow (i+1)$ 
17 kraj
```

Kod predložene dvorazinske ART arhitekture omogućeno je pretraživanje klasa tako da se ne ispituju sve postojeće klase, već se prvo evaluira izraz (2.6) za sve klase u G_{a2} razini te se klasa koja ima najveću vrijednost funkcija odabira T_{J_i} prema izrazu (2.7) izabire kao pobjednička. Tada se proces pretraživanja nastavlja unutar navedene klase u polju G_{a3} . Slučajno su generirane prostorne strukture pomoću generatora struktura opisanog algoritmom (Algoritam 2-3.). Ukupan broj generiranih struktura postavljen je u intervalu [5 85] s korakom 5. Za svaki je ulazni skup podataka napravljen slučajni redosljed na temelju kojeg su dobivena rasipanja oko srednje vrijednosti prikazana na dijagramima na slici (Slika 2-10.) i (Slika 2-11.).

Slika 2-10. Brzina pretraživanja postojećih klasa razvijene ART mreže za $\varphi=0,8$ $\varphi=0,85$ i $\varphi=0,9$.Slika 2-11. Brzina pretraživanja postojećih klasa razvijene ART mreže za $\varphi=0,9$ $\varphi=0,95$ i $\varphi=0,98$.

Usporedene su brzine pretraživanja postojećih klasa predložene ART mreže (ARTgrid) za vrijednosti parametra budnosti φ iz intervala $[0,8 - 0,98]$. Na dijagramima (Slika 2-10.) i (Slika 2-11.) vidljivo je kako je brzina pretraživanja u intervalu $\varphi \cong [0,8 - 0,95]$ vrlo ujednačena. Tek pri izrazitom povećanju parametra budnosti mreže na vrijednost $\varphi = 0,98$ dolazi do izraženog povećanja broja klasa [77], tj. na takozvani *category proliferation problem*. U tom slučaju broj klasa koje se generiraju u dvorazinskoj mreži više ne poboljšava brzinu pretraživanja u odnosu na standardnu arhitekturu ART mreže. U standardnoj ART arhitekturi uspoređuju se sve postojeće klase sa ulaznim uzorkom (Algoritam 2-4.). Granica $\varphi \cong 0,98$ postavljena je eksperimentalno simulacijom većeg broja slučajno generiranih struktura.

2.3.6. Pouzdanost klasifikacije – verifikacija

Kako bi se razvijena ART mreža verificirala, potrebno ju je testirati na prepoznatljivim strukturama s očekivanim rješenjima te izvršiti statističku obradu dobivenih rezultata.

U ovom poglavlju napravljena je analiza pouzdanosti klasifikacije primjenom parametara klasifikacijskih algoritama TP , TN , FP i FN , detaljno opisanih u poglavlju 2.3.6.1. Na temelju tih parametara mjeri se uspješnost klasifikacije koja se izražava pomoću: osjetljivosti (engl. *sensitivity*), odziva (engl. *recall*), preciznosti (engl. *precision*) i točnosti (engl. *accuracy*). Navedena četiri parametra su statističke veličine koje su određene omjerima vrijednosti iz tablice (Tablica 2-1), a detaljno su opisani u poglavlju 2.3.6.1. Ujedno je napravljena i Kappa-statistika [78,79] u svrhu evaluacije klasifikacije u više razreda (engl. *multiclass classification*). Podaci dobiveni klasifikacijom izraženi su pomoću konfuzijske matrice (Tablica 2-1.).

Tablica 2-1. Konfuzijska matrica dvaju razreda (dviju klasa).

| | Stvarni razredi | |
|----------------------------------|---------------------------|---------------------------|
| | Razred pozitivnih (P) | Razred negativnih (N) |
| Predviđanje pozitivnih (R^+) | Pravi pozitivni (TP) | Krivi pozitivni (FP) |
| Predviđanje negativnih (R^-) | Krivi negativni (FN) | Pravi negativni (TN) |

2.3.6.1. Opis validacije klasifikacije

Verifikacija klasifikatora koji klasificiraju ulazne uzorke u više od dvije izlazne klase koristi se sličnom metodologijom kao i binarni klasifikatori [80]. Pri apliciranju ulaznog uzorka potrebno je svesti mogući slučaj klasifikacije u veći broj izlaznih klasa na binarnu klasifikaciju, gdje se ispravna klasa promatra kao klasa-1, dok se sve ostale klase promatraju kao klasa-2. Ujedno je moguće ispitivati i pripadnost pojedinog uzorka svim dostupnim klasama te prema tome izraditi pouzdanost klasifikacije [81].

Validacija klasifikacije prema izvršenim eksperimentima (testovima) podijeljena je u dvije osnovne skupine:

- I. Pouzdanost klasifikacije pri učenju mreže tj. kada je mreža „prazna”. Pripadnost svake ulazne strukture označena je numeričkom oznakom klase kojoj pripada. Taj je postupak

sličan validaciji klasifikatora koji klasificira ljudska lica, gdje je u skupu za validaciju svaka slika označena kao slika koja sadrži ljudsko lice ili kao slika koja ne sadrži ljudsko lice. Na taj je način moguće nakon procesa učenja ART mreže s određenim brojem ulaznih uzoraka izračunati ukupni broj TP , TN , FP i FN . Pomoću tih vrijednosti tada se određuju osjetljivost, odziv, preciznost i točnost. Ponavljanjem procesa učenja mreže te mijenjanjem redoslijeda ulaznih struktura računaju se statistički parametri rasipanja pojedinih vrijednosti.

- II. Drugi skup eksperimenata određuje pouzdanost klasifikacije s prethodno inicijaliziranom ART mrežom koja već ima određene izlazne klase i težinske vrijednosti. Aplikira se novi skup ulaznih struktura, kao što je opisano i za skupinu I. Postupak se ponavlja veći broj puta kako bi se ispitali različiti redoslijedi ulaznih struktura te se bilježe parametri TP , TN , FP i FN . Navedeni postupak također je statistički obrađen sa svrhom određivanja rasipanja pojedinih vrijednosti.

Kao osnovno mjerilo pouzdanosti i uspješnosti klasifikacije uzima se konfuzijska matrica (engl. *confusion matrix*) [82], pomoću koje se uvodi razlikovanje vrste pogrešaka. Četiri osnovna parametra pomoću kojih se formira konfuzijska matrica odnosno tablica zabune prikazani su u tablici (Tablica 2-1.). Konfuzijska matrica odgovara predviđanju pojavljivanja ili nedogađaja (hipoteze), tzv. „razred pozitivnih i razred negativnih primjera“.

Postoje dva tipa pogreške [79]:

- Krivo klasificiranje ulaznog uzorka u određenu klasu iako njoj ne pripada – krivi pozitivni primjeri: FP (engl. *false positives*) – greška tipa I.
- Negativan rezultat klasifikacije ulaznog uzorka za klasu kojoj on stvarno pripada – krivi negativni primjeri: FN (engl. *false negatives*) – greška tipa II.

Postupak klasifikacije je ispravan u preostala dva slučaja koji se nalaze na glavnoj dijagonali:

- Ispravno klasificiranje ulaznog uzorka u klasu kojoj stvarno pripada – pravi pozitivni primjeri: TP (engl. *true positives*).
- Ispravna negativna klasifikacija za klasu kojoj uzorak stvarno ne pripada – pravi negativni: TN (engl. *true negatives*).

Prema četiri navedena moguća ishoda klasifikacije TP , TN , FP , FN određuju se standardni indikatori pogreške klasifikacije kojima se mjeri učinkovitost sustava.

2. Učenje i klasifikacija prostornih struktura

Neka $P = TP + FN$ označava stvarnu pozitivnu pripadnost klasi, a $N = FP + TN$ stvarnu negativnu pripadnost klasi. Tada se definiraju sljedeći omjeri [79,83]:

- osjetljivost ili odziv (engl. *sensitivity, recall*): $TPR = TP/P$
- specifičnost (engl. *specificity*): $SPC(TNR) = TN/N$
- točnost (engl. *accuracy*): $ACC = (TP+TN)/(P+N)$
- preciznost (engl. *precision, positive predictive value*): $PPV = TP/(TP+FP)$ tj. TP/R^+
- predviđena vrijednost negativnih (engl. *negative predictive value*): $NPV = TN/(TN+FN)$ tj. TN/R^-

Postupak klasifikacije u više od dva razreda moguće je svesti na seriju klasifikacije u dva razreda [83]. Ulazne uzorke moguće je serijski testirati na svakoj postojećoj klasi te bilježiti prethodno navedene parametre. Pri klasifikaciji u više razreda moguće je prikazati cjelokupnu konfuzijsku matricu koja tada na glavnoj dijagonali prikazuje broj ispravno klasificiranih uzoraka, dok su sve pogrešne klasifikacije iznad ili ispod dijagonale.

U tablici (Tablica 2-2.) prikazan je primjer konfuzijske matrice za skup od 130 uzoraka koji se klasificiraju u četiri razreda.

Tablica 2-2. Primjer konfuzijske matrice za klasifikaciju u četiri razreda.

| | | Predviđena klasa | | | | Σ |
|------------------|-----|------------------|----|-----|----|----------|
| | | I | II | III | IV | |
| Izvorna klasa | I | 21 | 0 | 4 | 1 | 26 |
| | II | 0 | 37 | 3 | 6 | 46 |
| | III | 2 | 3 | 19 | 0 | 24 |
| | IV | 4 | 3 | 1 | 26 | 34 |
| Σ | | 27 | 43 | 27 | 33 | |

Testni skup ima 130 podataka. Od toga je ispravno klasificirano $21 + 37 + 19 + 26 = 103$. Uspješnost je određena kao omjer broja uspješno klasificiranih uzoraka promatranog klasifikatora i idealnog klasifikatora, $s^{\text{klasifikator}}/s^{\text{idealni}}$ te iznosi $103/130 = 79\%$. Klasifikator je svrstao (predvidio) 21 uzorak u razred I, 37 uzoraka u razred II, 19 u razred III te 26 uzoraka u razred IV. Nadalje, potrebno je odrediti konfuzijsku matricu za slučajni klasifikator koji bi svrstao isti broj primjera 26-46-24-34 u omjeru $27 - 43 - 27 - 33$ tj. $21\% - 33\% - 21\% - 25\%$ kao i promatrani klasifikator. U tablici (Tablica 2-3.) prikazana je konfuzijska matrica slučajnog klasifikatora.

Tablica 2-3. *Primjer konfuzijske matrice za slučajni klasifikator koji raspoređuje ulazne uzorke u omjeru kao i promatrani klasifikator iz tablice (Tablica 2-2.).*

| | | Predviđena klasa | | | | Σ |
|------------------|-----|------------------|----|-----|----|----------|
| | | I | II | III | IV | |
| Izvorna klasa | I | 5 | 9 | 5 | 7 | 26 |
| | II | 10 | 15 | 10 | 11 | 46 |
| | III | 5 | 8 | 5 | 6 | 24 |
| | IV | 7 | 11 | 7 | 9 | 34 |
| Σ | | 27 | 43 | 27 | 33 | |

Takav slučajni klasifikator koji uzima u obzir omjer klasifikacije izvornog klasifikatora prikazanog u tablici (Tablica 2-2.) ispravno je klasificirao $5+15+5+9=34$ primjera. Kako bi se dobila vrijednost Kappa-statistike koja je dana izrazom (2.16), rezultat uspješnosti (34) slučajnog klasifikatora oduzima se izvornog klasifikatora (103-34) te se stavlja u omjer korigiranog idealnog klasifikatora (130-34). Kappa-statistika za prikazani primjer iznosi $\kappa = (103-34)/(130-34) = 72\%$. Maksimalna vrijednost Kappa-statistike iznosi 100%. Kappa-statistika izražava mjeru uspješnosti promatranog klasifikatora (103) u odnosu na idealni klasifikator (130) korištenjem korekcije slučajnog izbora (34).

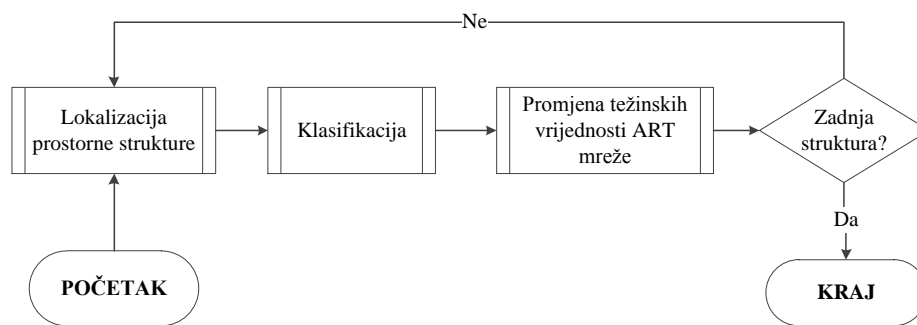
$$\kappa = \frac{\left(s^{\text{klasifikator}} - s^{\text{rand}} \right)}{\left(s^{\text{idealno}} - s^{\text{rand}} \right)}. \quad (2.16)$$

2.3.6.2. Evaluacija klasifikacije – testna skupina I.

Za razvijenu ART neuronsku mrežu prikazanu algoritmom (Algoritam 2-2.) odredit će se parametri točnosti klasifikacije. Osnovno svojstvo razvijene ART mreže očituje se u inkrementalnom učenju ulaznih struktura te u pripadajućoj promjeni težinskih vrijednosti na temelju funkcije učenja prikazanoj u (2.4). Jedan od problema koji proizlazi iz procesa inkrementalnog učenja je zaboravljivost mreže što je opisano u poglavlju 1.2. Jedan od načina na koji je moguće smanjiti utjecaj retroaktivne interferencije je promjena parametra budnosti mreže φ . Veće vrijednosti parametra budnosti utječu na kreiranje većeg broja izlaznih klasa mreže čime je moguće utjecati na interferenciju. Nadalje, smanjivanjem vrijednosti stope učenja β nakon određenog vremena ili broja iteracija, znanje o novo dolazećim strukturama je moguće prigušiti i tako sačuvati informacije iz primarne faze učenja.

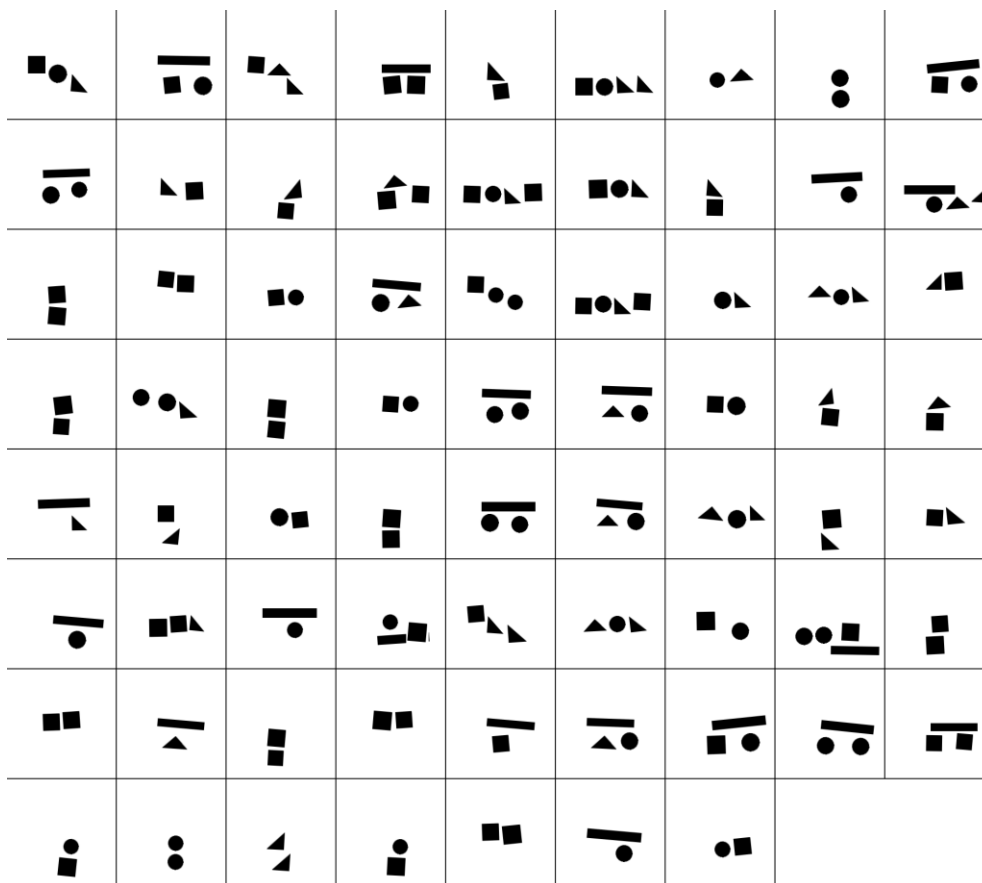
2. Učenje i klasifikacija prostornih struktura

Kao osnovni princip kasnije implementacije ART mreže očekuje se kako će robot čiji upravljački model koristi ART mrežu moći klasificirati slične prostorne strukture u iste razrede. Nadalje, potrebno je omogućiti da se cijeli postupak izvodi u inkrementalnom režimu rada, gdje će se robotu prikazati određene prostorne strukture koje će ART mreža klasificirati te ažurirati težinske vrijednosti, kao što će biti prikazano u poglavlju 5.1. Dijagram toka na slici (Slika 2-12.) prikazuje osnovne komponente procesa. Lokalizacija prostornih struktura u ovom je poglavlju zamijenjena određenim brojem prethodno generiranih slučajnih struktura pomoću algoritma (Algoritam 2-3.). Na taj se način simulira realan proces akvizicije i lokalizacije objekata u radnom prostoru robota koji je kasnije prikazan i validiran u poglavlju 5.1. Proces započinje s neinicijaliziranom ART mrežom koja u početku ne sadrži nijednu izlaznu klasu.



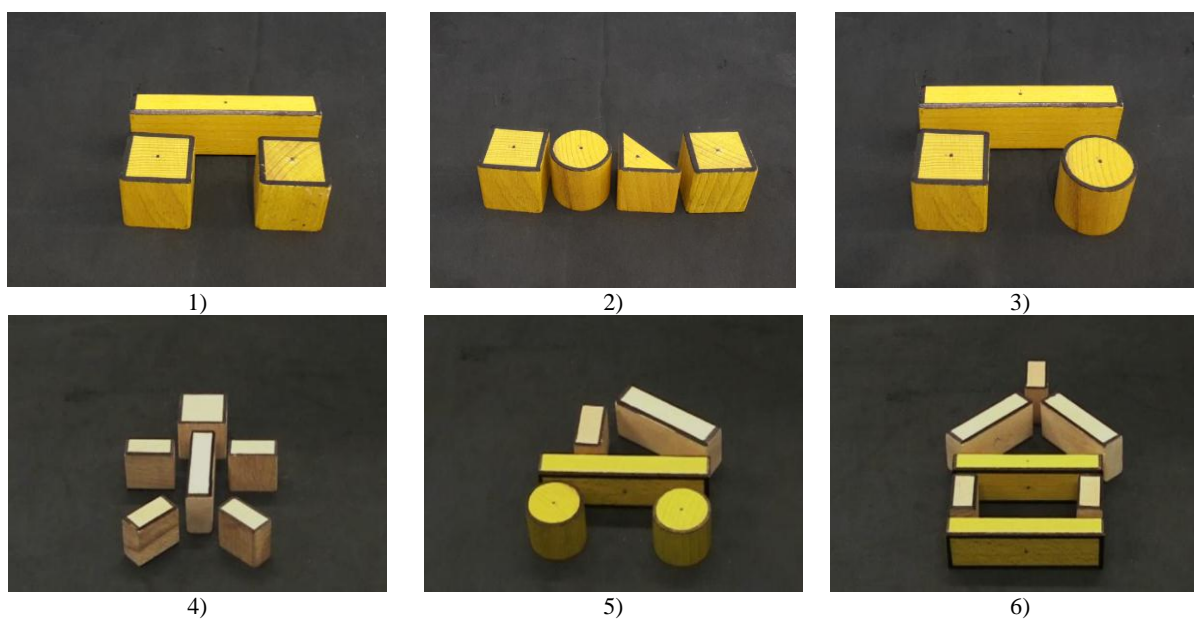
Slika 2-12. Učenje prostornih struktura.

Prostorne strukture prikazane na slici (Slika 2-13.) apliciraju se slučajnim redoslijedom te se izvršava njihova klasifikacija. Nakon svake aplicirane strukture i njezine klasifikacije dolazi do promjene težinskih vrijednosti mreže kao što je već naznačeno, korištenjem izraza (2.4).



Slika 2-13. *Primjer 70 prostornih struktura generiranih pomoću algoritma (Algoritam 2-3.).*

Svaka od simboličkih struktura prikazanih na slici (Slika 2-13.) predstavlja dvodimenzionalnu interpretaciju poznatih objekata, gdje su neki primjeri prikazani na slici (Slika 2-14.).



Slika 2-14. *Različite prostorne strukture sastavljene od poznatih objekata.*

Sljedeći korak podrazumijeva dodjeljivanje oznaka pojedinoj strukturi koja određuje pripadnost te strukture određenoj izlaznoj klasi, što je prikazano na slici (Slika 2-15.).

| | | | | | | | | |
|-------|-------|-------|-------|-------|------|-------|------|-------|
| (I) | (IV) | (I) | (IV) | (IV) | (I) | (III) | (II) | (IV) |
| (IV) | (III) | (II) | (IV) | (I) | (I) | (II) | (IV) | (I) |
| (II) | (III) | (III) | (IV) | (I) | (I) | (III) | (I) | (III) |
| (II) | (I) | (II) | (III) | (IV) | (IV) | (III) | (IV) | (II) |
| (IV) | (II) | (III) | (II) | (IV) | (IV) | (I) | (II) | (III) |
| (III) | (I) | (IV) | (IV) | (I) | (I) | (IV) | (I) | (II) |
| (III) | (IV) | (II) | (III) | (IV) | (IV) | (IV) | (IV) | (IV) |
| (II) | (II) | (IV) | (II) | (III) | (IV) | (III) | | |

Slika 2-15. Primjer 70 prostornih struktura generiranih pomoću algoritma (Algoritam 2-3.) te označenih prema pripadnosti jednoj od 4 moguće klase I, II, III ili IV.

Kao zlatni standard (engl. *gold standard*) referentnog označavanja ulaznog skupa podataka uzima se prosječna vrijednost od 3 označivača (nezavisna klasifikatora) [79]. Klasifikacija ulaznih podataka prikazanih na slici (Slika 2-13.) i (Slika 2-14.) u određenoj mjeri predstavlja subjektivni klasifikacijski problem za koji nije jednoznačno moguće dodijeliti pripadnost pojedinoj klasi. Kod ovakvih problema postoji mogućnost različitog postavljanja zlatnog standarda. Zbog toga je potrebno kvantificirati subjektivnost problema [83]. Kao mjera slaganja između označivača (engl. *Inter annotator agreement – IAA*) koristi se Kappa-statistika. Interpretacija vrijednosti dobivenih Kappa-statistikom temelji se na podacima iz [84]. Najčešće se uzima $\kappa \geq 0,67$ – sklonost slaganju (engl. *tentative agreement*) te $\kappa \geq 0,8$ – dobro slaganje tj. podudarnost (engl. *good agreement*). Kako bi se utvrdila mjera slaganja između označivača te kako bi se utvrdilo postoji li sklonost slaganju ili dobro slaganje,

2. Učenje i klasifikacija prostornih struktura

napravljene su tri konfuzijske matrice koje prikazuju neslaganje svakog mogućeg para označivača.

Tablica 2-4. Konfuzijska matrica između označivača 1 i označivača 2.

| | | Predviđena klasa | | | | Σ |
|---------------|-----|------------------|----|-----|----|----------|
| | | I | II | III | IV | |
| Izvorna klasa | I | 14 | 0 | 0 | 1 | 15 |
| | II | 1 | 10 | 1 | 3 | 15 |
| | III | 0 | 0 | 13 | 2 | 15 |
| | IV | 2 | 0 | 1 | 22 | 25 |
| Σ | | 17 | 10 | 15 | 28 | |

Tablica 2-5. Konfuzijska matrica između označivača 1 i označivača 3.

| | | Predviđena klasa | | | | Σ |
|---------------|-----|------------------|----|-----|----|----------|
| | | I | II | III | IV | |
| Izvorna klasa | I | 14 | 0 | 1 | 0 | 15 |
| | II | 1 | 14 | 0 | 0 | 15 |
| | III | 0 | 0 | 14 | 1 | 15 |
| | IV | 1 | 3 | 1 | 20 | 25 |
| Σ | | 16 | 17 | 16 | 21 | |

Tablica 2-6. Konfuzijska matrica između označivača 2 i označivača 3.

| | | Predviđena klasa | | | | Σ |
|---------------|-----|------------------|----|-----|----|----------|
| | | I | II | III | IV | |
| Izvorna klasa | I | 13 | 0 | 1 | 1 | 15 |
| | II | 0 | 14 | 1 | 0 | 15 |
| | III | 1 | 0 | 14 | 0 | 15 |
| | IV | 2 | 0 | 1 | 22 | 25 |
| Σ | | 16 | 14 | 17 | 23 | |

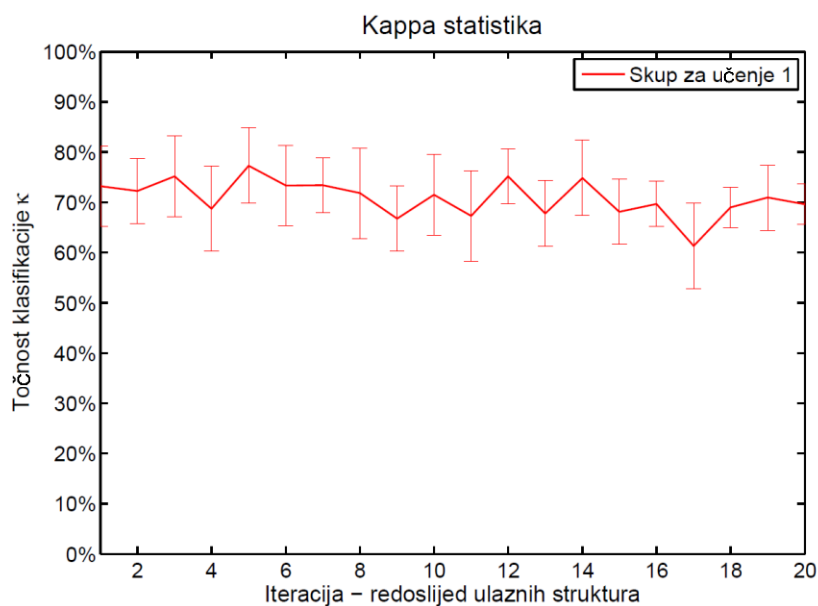
Za svaku od tri tablice (Tablica 2-4.) – (Tablica 2-6.) izračunana je Kappa-statistika prema izrazu (2.16). Na taj je način utvrđen postotak slaganja dviju nezavisnih klasifikacija uz korekciju slučajnog izbora. U tablici (Tablica 2-6.) prikazani su rezultati Kappa-statistike te je određena mjera sklonosti slaganju.

Tablica 2-7. Ukupna mjera sklonosti slaganju.

| | Kappa-statistika | | |
|---------------------|------------------|------------|------------|
| | I | II | III |
| κ | 0,78 | 0,84 | 0,86 |
| $K - \text{greška}$ | 0,06 | 0,05 | 0,05 |
| $\frac{s}{\kappa}$ | $\geq 0,67$ | $\geq 0,8$ | $\geq 0,8$ |
| $\bar{\kappa}$ | 0,827 | | |

Iz dobivenih rezultata za jedan slučaj dobivena je „sklonost slaganju” ($\kappa \geq 0,67$), dok je u preostala dva slučaja dobiveno „dobro slaganje” ($\kappa \geq 0,8$). Srednja vrijednost $\bar{\kappa} \geq 0,8$, te je time također dobiveno „dobro slaganje”. U slučajevima kada je $\kappa \leq 0,67$, dolazi do problema pri označavanju uzoraka te je upitno koliko su dodijeljene klase relevantne. Pri označavanju ulaznih uzoraka prikazanih na slici (Slika 2-13.) prihvaćeno je označavanje klasa prikazano slikom (Slika 2-15.).

Prema dijagramu prikazanom na slici (Slika 2-12.) i algoritmu (Algoritam 2-2.), ART mreža klasificira sve predstavljene prostorne strukture prikazane na slici (Slika 2-13.). U postupku klasifikacije bilježi se broj uspješnih i neuspješnih klasifikacija parametrima TP , TN , FP i FN . Za lakšu vizualizaciju dobivenih rezultata napravljene su konfuzijske matrice za klasifikaciju skupa za učenje i skupa za testiranje. Postupak klasifikacije ponavlja se 20 puta, pri čemu je ulazni redoslijed struktura potpuno slučajan za svaku iteraciju. Kao izlaz dobiva se 20 različitih kombinacija težinskih vrijednosti za pojedinačnu klasu u ART mreži. Rezultati Kappa-statistike (κ) prikazani su na dijagramu (Slika 2-16.). Postavljeni parametri mreže su: $\varphi_1 = 0,57$, $\varphi_2 = 0,67$, $\beta_1 = 0,25$, $\beta_2 = 0,20$.



Slika 2-16. Prikaz Kappa-koeficijenta za 20 različitih slučajeva učenja ART mreže sa skupom za učenje koji je prikazan na slici (Slika 2-15.).

Primjer uprosječne konfuzijske matrice s iskazanom standardnom devijacijom prikazan je u tablici (Tablica 2-8.). Konfuzijska matrica prikazuje rezultate klasifikacije dobivene različitim slučajnim ulaznim redosljedom prostornih struktura prilikom učenja neuronske mreže u jednoj iteraciji.

Tablica 2-8. Uprosječena konfuzijska matrica za slučaj klasifikacije 70 prostornih struktura prikazanih na slici (Slika 2-15.). U zagradama je naznačena standardna devijacija.

| | | Predviđena klasa | | | | Σ |
|---------------|-----|------------------|------------|------------|-----------|----|
| | | I | II | III | IV | |
| Izvorna klasa | I | 8,9 (1,6) | 1,5 (1,1) | 2,3 (1,4) | 2,4 (1,2) | 15 |
| | II | 0,8 (0,7) | 11,3 (0,9) | 0,9 (0,7) | 2,1 (1) | 15 |
| | III | 1,7 (1,2) | 0,9 (0,9) | 10,5 (1,3) | 2 (1,1) | 15 |
| | IV | 1,6 (1,2) | 2,4 (1,3) | 2,1 (1,4) | 19 (1,7) | 25 |
| | Σ | 13 | 16,1 | 15,8 | 25,5 | |

Uprosječena tablica zabune prikazuje dobivene rezultate 20 neovisnih klasifikacija. Za svaku od navedenih klasifikacija izračunani su parametri TP , TN , FP i FN za pojedinu klasu j ($j = 1, \dots, K$). Ukupni broj ulaznih uzoraka određen je s N , pri čemu je za navedeni primjer $N = 70$.

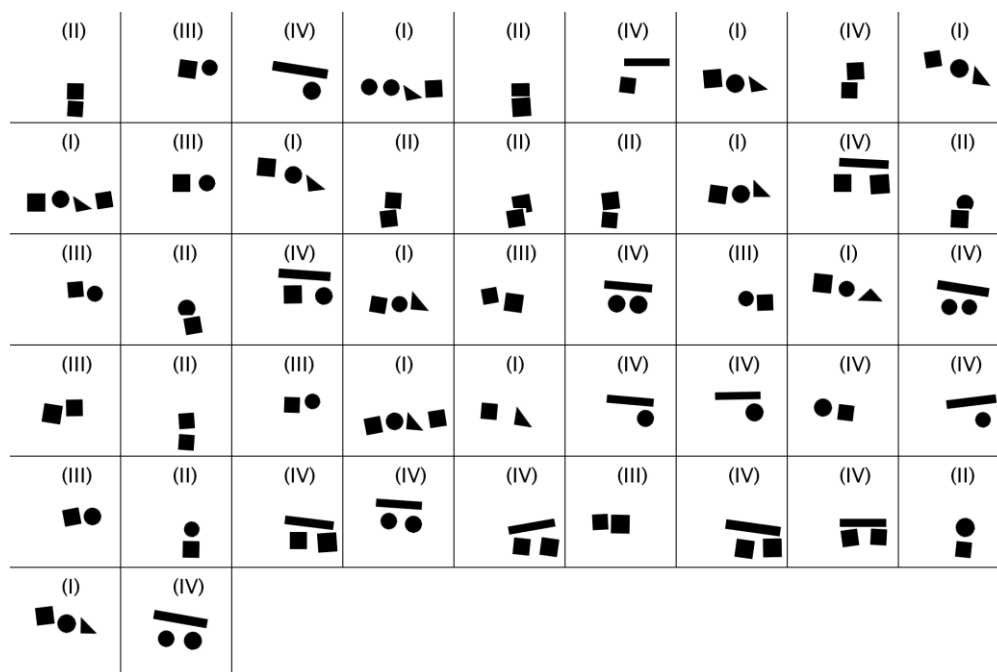
$$\begin{aligned}
 TP &= \sum_{j=1}^K TP_j \\
 FP &= \sum_{j=1}^K FP_j \\
 FN &= \sum_{j=1}^K FN_j
 \end{aligned}
 \tag{2.17}$$

$$TN = K \cdot N - TP - FP - FN
 \tag{2.18}$$

Kod klasifikacije u veći broj klasa ($j > 2$) uvijek vrijedi $FP = FN$, dok je TP vrijednost traga tablice zabune [83].

2.3.6.3. Evaluacija klasifikacije – testna skupina II.

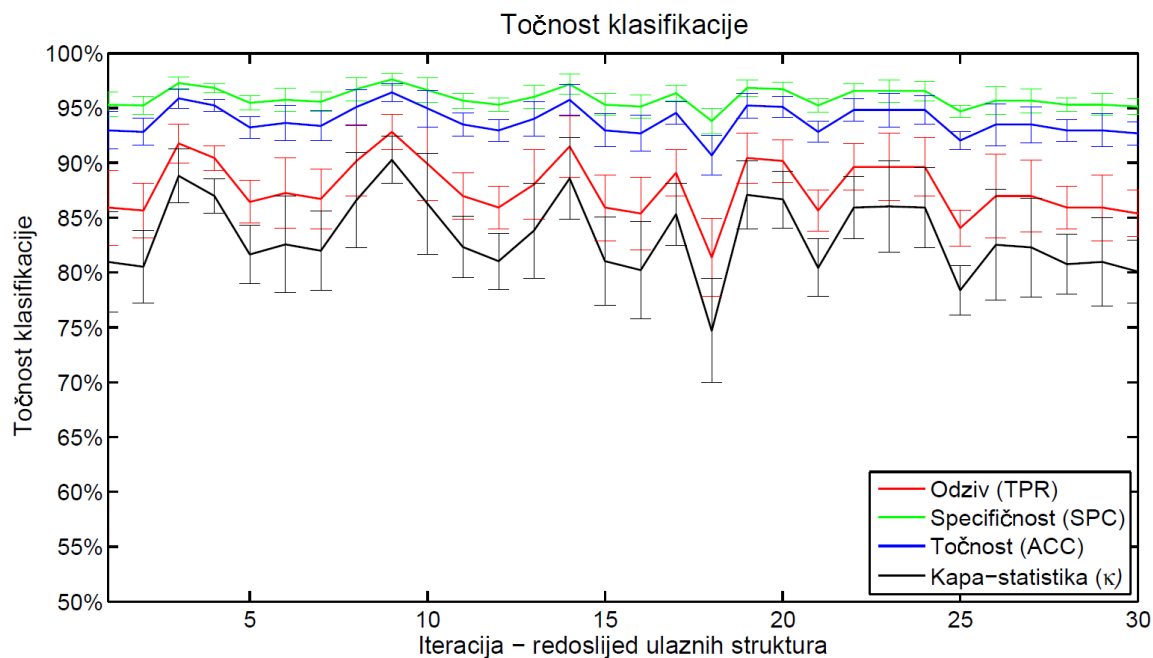
U svrhu testiranja parametara točnosti klasifikacije generirano je 47 slučajnih prostornih struktura (Slika 2-17.) primjenom algoritma (Algoritam 2-3.). Sve strukture označene su od I do IV, ovisno o tome kojoj klasi pripadaju. Strukture su klasificirane pomoću ART mreže te su dobiveni rezultati točnosti klasifikacije prikazani na dijagramu (Slika 2-18.). Za svaku od dobivenih klasifikacija ART mreže tj. pripadajućih težinskih vrijednosti iz poglavlja 2.3.6.2. napravljeno je 10 postupaka klasifikacije s različitim (slučajnim) ulaznim redoslijedom struktura.



Slika 2-17. Skup od 47 prostornih struktura za testiranje točnosti klasifikacije.

2. Učenje i klasifikacija prostornih struktura

Na dijagramu (Slika 2-18.) pri verifikaciji ART mreže prikazani su parametri odziva (*TPR*), specifičnosti (*SPC*), točnosti (*ACC*) te Kappa-statistika (κ). Iz dobivenih je rezultata vidljivo da je postignuta relativno stabilna klasifikacija.



Slika 2-18. Prikaz parametara točnosti klasifikacije za klasifikaciju prostornih struktura prikazanih na slici (Slika 2-17.).

Svojstvo ART mreže da kontinuirano mijenja težinske vrijednosti omogućava da se mreža adaptira promjenama u okolini. U slučaju razvijene ART mreže te su promjene određene različitim prostornim strukturama koje je potrebno klasificirati, a u narednim fazama će pojedina klasa imati određeno značenje pri procesu planiranja djelovanja. Nadalje, svojstvo učenja ART mreže rezultira različitim težinskim vrijednostima u pojedinim fazama učenja te su zbog toga vidljive određene oscilacije na dijagramu (Slika 2-18.).

3. POVEZIVANJE NEUREĐENOG PROSTORNOG STANJA S UREĐENIM

Jedan od ciljeva ovog istraživanja je razvoj i verifikacija upravljačkog modela koji će osposobiti robota za prepoznavanje i izgradnju uređenih prostornih struktura. Prepoznavanje uređenih prostornih struktura i njihova klasifikacija opisani su u poglavlju 2. U svrhu planiranja i kasnijeg postupka izgradnje uređenih prostornih struktura iz neuređenog prostornog stanja objekata, kao što je prikazano na slici (Slika 3-1.), potrebno je za navedeno neuređeno prostorno stanje pronaći najbližnju, prethodno klasificiranu uređenu prostornu strukturu. Ovaj postupak temelji se na asocijativnom povezivanju te identifikaciji najbližnjih objekata u postojećim klasama s lokaliziranim objektima na sceni, tj. u radnom prostoru robota.



Slika 3-1. *Neuređeno prostorno stanje objekata.*

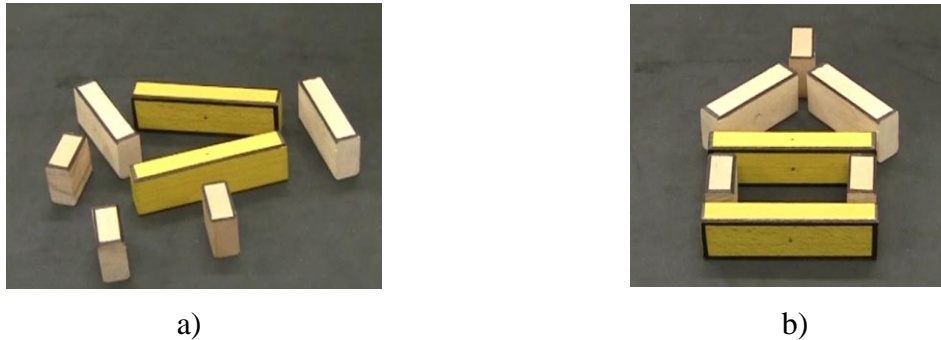
Nakon postupka klasifikacije uređenih prostornih struktura potrebno je za određeno neuređeno prostorno stanje objekata, kao što je prikazano na slici (Slika 3-1.), generirati izlaznu strukturu ovisno o sličnostima individualnih objekata.

3.1. Funkcija povezivanja

Funkcija povezivanja određuje preslikavanje objekata iz neuređenog prostornog stanja na određenu uređenu prostornu strukturu. Primjer trivijalnog slučaja prikazan je na slici (Slika 3-2.). Objekti u lokaliziranoj strukturi i objekti u matrici objekata jedne od klasificiranih

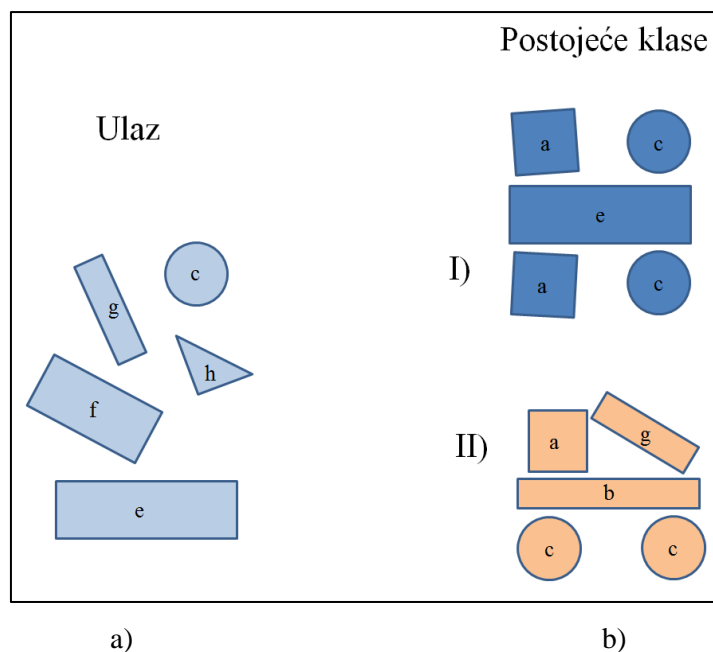
3. Povezivanje neuređenog prostornog stanja s uređenim

struktura identični te su aktivnosti svih objekata 100%, odnosno 1. U tom slučaju nije potrebno računati preklapanje, već se navedena struktura izabire kao pobjednička te se svi objekti na sceni preslikavaju na identične objekte u postojećoj klasificiranoj strukturi. U ovom je slučaju potrebno generirati optimalno preslikavanje za istu vrstu objekata (na slici su prikazane tri vrste objekata), kao što će biti opisano u poglavlju 3.2.



Slika 3-2. Trivijalan slučaj preslikavanja neuređenog stanja a) na postojeću, prethodno klasificiranu strukturu b).

Slučaj koji nema trivijalno rješenje prikazan je na slici (Slika 3-3.). Na slici (Slika 3-3.a) prikazan je shematski primjer lokaliziranog radnog prostora robota u kojem se nalazi pet objekata. Neka ART mreža (Slika 3-3.b) sadrži dvije moguće klase: I. i II. Zadatak je pronaći optimalno preslikavanje (pridruživanje) lokaliziranih objekata u jednu od postojećih izlaznih klasa te generirati izlaznu strukturu za koju će se kasnije planirati slijed izgradnje opisan u poglavlju 4.

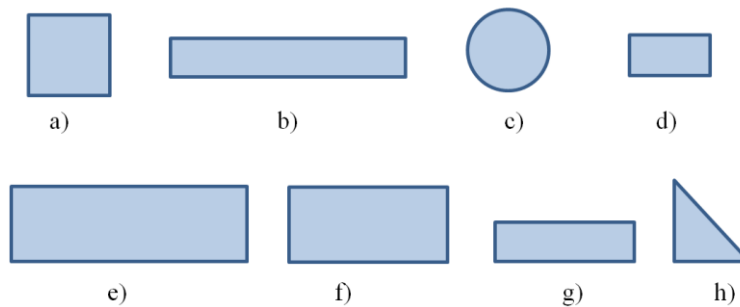


Slika 3-3. a) neuređeno prostorno stanje objekata, b) postojeće izlazne klase.

Objekti koji su lokalizirani na sceni razlikuju se u odnosu na objekte u postojećim strukturama I. i II. Potrebno je kvantificirati sličnosti na razini pojedinačnih objekata, za što se primjenjuje funkcija (3.1), koja u inicijalnom koraku određuje sličnosti pojedinačnih objekata prema sljedećem izrazu:

$$Q_{ij}^k = \frac{\|U_i \wedge S_j^k\|}{\|U_i \vee S_j^k\|} \cdot a_i, \quad i = 1, 2, \dots, M, \quad j = 1, 2, \dots, N_k, \quad k = 1, 2, \dots, P. \quad (3.1)$$

Funkcija kvantifikacije sličnosti individualnih objekata koristi se omjerom definiranim izrazom (2.6), korištenim u postupku kvantifikacije sličnosti dviju prostornih struktura. Za svaki od lokaliziranih objekata računa se sličnost s postojećim objektima u pojedinoj klasi, gdje U_i označava lokalizirani objekt, S_j^k označava objekt koji se nalazi u pohranjenoj klasi, $a_i \in [0,1]$ označava aktivnost pojedinog objekta iz postojeće klase. Ukupan broj objekata na sceni označen je s M , broj postojećih klasa s P , a broj objekata klase k kao N_k . Analogno postupku klasifikacije prostornih struktura opisanih u poglavlju 2.3.2., $U_g = (U_{g_{11}}, U_{g_{12}}, \dots, U_{g_{mn}})$, $g \in [1, M]$ označava lokalizirani objekt, pri čemu je $U_{g_{ij}} \in [0,1]$. Analogno vrijedi i za S_j^k . Svaki piksel na slici koji odgovara objektu poprima vrijednost 1, dok svi ostali pikseli poprimaju vrijednost 0, tvoreći na taj način inicijalnu binarnu matricu. Iz izraza (3.1) vidljiva je analogija s kvantifikacijom sličnosti dviju struktura prema izrazu (2.6). Kvantifikacija sličnosti dvaju objekata dodatno je određena s parametrom aktivnosti pojedinog objekta u postojećoj klasificiranoj strukturi prema postupku koji je opisan u poglavlju 2.3.3. Sličnost dvaju objekata a i b, odnosno njihovih oblika, određena je u intervalu $s(a,b) \in [0,1]$, pri čemu vrijednost 1 označava identične objekte, dok se vrijednost s smanjuje uzimajući u obzir izraz (3.1). Na slici (Slika 3-4.) prikazano je osam vrsta objekata.



Slika 3-4. Osam vrsta objekata.

3. Povezivanje neuređenog prostornog stanja s uređenim

Postupak preslikavanja i asocijativnog povezivanja neuređenog prostornog stanja s postojećim klasificiranim strukturama bit će objašnjen na primjeru sa slike (Slika 3-3.). Prvi korak u postupku određivanja izlazne strukture koja odgovara lokaliziranom neuređenom stanju objekata odnosi se na kvantificiranje sličnosti svih objekata. Za primjer prikazan na slici (Slika 3-3.) generirat će se sve potrebne vrijednosti za određivanje kvadratne matrice ukupnih sličnosti. Matrica kvantifikacije sličnosti svih objekata (a, b, c, d, e, f, g, h) je kvadratna matrica

$$\mathbf{H} = \begin{bmatrix} 1 & 0,25 & 0,79 & 0,49 & 0,34 & 0,5 & 0,33 & 0,52 \\ 0,25 & 1 & 0,27 & 0,34 & 0,49 & 0,4 & 0,67 & 0,15 \\ 0,79 & 0,27 & 1 & 0,59 & 0,27 & 0,4 & 0,37 & 0,45 \\ 0,49 & 0,34 & 0,59 & 1 & 0,17 & 0,25 & 0,5 & 0,34 \\ 0,34 & 0,49 & 0,27 & 0,17 & 1 & 0,67 & 0,33 & 0,18 \\ 0,5 & 0,4 & 0,4 & 0,25 & 0,67 & 1 & 0,49 & 0,26 \\ 0,33 & 0,67 & 0,37 & 0,5 & 0,33 & 0,49 & 1 & 0,21 \\ 0,52 & 0,15 & 0,45 & 0,34 & 0,18 & 0,26 & 0,21 & 1 \end{bmatrix}$$

pri čemu element matrice H_{ij} odgovara sličnosti i -tog i j -tog objekta. Uređeni skup objekata dan je kao $O = \{a, b, c, d, e, f, g, h\}$. Na dijagonali matrice nalaze se jedinice jer je sličnost dvaju identičnih objekata jednaka 1. Nakon što je određena matrica \mathbf{H} , potrebno je generirati pripadajuće matrice sličnosti za ulaznu strukturu i postojećih klasa I i II. Skup ulaznih objekata određen je $O_1 = \{c, e, f, g, h\}$, skup objekata klase I: $O_2 = \{a, a, c, c, e\}$ te skup objekata klase II: $O_3 = \{a, b, c, c, g\}$. Generiraju se dvije matrice sličnosti $\mathbf{Q}_1^2(O_1, O_2)$ i $\mathbf{Q}_1^3(O_1, O_3)$:

$$\mathbf{Q}_1^2 = \begin{bmatrix} 0,79 & 0,79 & 1 & 1 & 0,27 \\ 0,34 & 0,34 & 0,27 & 0,27 & 1 \\ 0,5 & 0,5 & 0,4 & 0,4 & 0,67 \\ 0,33 & 0,33 & 0,37 & 0,37 & 0,33 \\ 0,52 & 0,52 & 0,45 & 0,45 & 0,18 \end{bmatrix}, \quad \mathbf{Q}_1^3 = \begin{bmatrix} 0,79 & 0,27 & 1 & 1 & 0,37 \\ 0,34 & 0,49 & 0,27 & 0,27 & 0,33 \\ 0,5 & 0,4 & 0,4 & 0,4 & 0,49 \\ 0,33 & 0,67 & 0,37 & 0,37 & 1 \\ 0,52 & 0,15 & 0,45 & 0,45 & 0,21 \end{bmatrix}.$$

Za navedene matrice \mathbf{Q}_1^2 i \mathbf{Q}_1^3 potrebno je pronaći optimalno pridruživanje objekata iz ulazne strukture na objekte u pojedinoj izlaznoj strukturi. Optimalno pridruživanje određeno je kao ono pridruživanje koje ima maksimalnu kumulativnu sumu svih sličnosti. Nadalje, isti je problem moguće promatrati kroz njegov komplement, gdje se za svaki element $\mathbf{Q}_{1_{ij}}^2$ računa $\overline{\mathbf{Q}_{1_{ij}}^2} = 1 - \mathbf{Q}_{1_{ij}}^2$. Tada su komplementarne matrice $\overline{\mathbf{Q}_{1_{ij}}^2}$ i $\overline{\mathbf{Q}_{1_{ij}}^3}$ određene s:

3. Povezivanje neuređenog prostornog stanja s uređenim

$$\overline{\mathbf{Q}}_1^2 = \begin{bmatrix} 0,21 & 0,21 & 0 & 0 & 0,73 \\ 0,66 & 0,66 & 0,73 & 0,73 & 0 \\ 0,5 & 0,5 & 0,6 & 0,6 & 0,33 \\ 0,67 & 0,67 & 0,63 & 0,63 & 0,67 \\ 0,48 & 0,48 & 0,55 & 0,55 & 0,82 \end{bmatrix}, \quad \overline{\mathbf{Q}}_1^3 = \begin{bmatrix} 0,21 & 0,73 & 0 & 0 & 0,63 \\ 0,66 & 0,51 & 0,73 & 0,73 & 0,67 \\ 0,5 & 0,6 & 0,6 & 0,6 & 0,51 \\ 0,67 & 0,33 & 0,63 & 0,63 & 0 \\ 0,48 & 0,85 & 0,55 & 0,55 & 0,79 \end{bmatrix}.$$

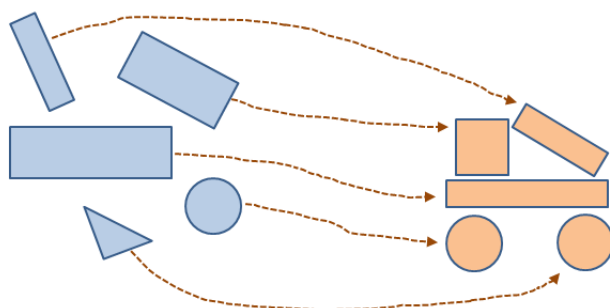
Za navedene komplementarne matrice potrebno je pronaći takvo pridruživanje elemenata (objekata) iz redova na elemente iz stupca matrice kako bi se minimizirala ukupna suma svih parcijalnih sličnosti $\sum \overline{\mathbf{Q}}_{1,ij}^2$. Iz ove komplementarne matrice treba pronaći minimalnu vrijednost preslikavanja objekata iz ulazne strukture na pripadajuće objekte postojeće klase. Vremenska složenost za izračunavanje optimalnog preslikavanja iznosi $n!$. Za navedeni primjer potrebno je ispitati ukupno $3! = 6$ mogućih kombinacija, dok za slučaj od deset objekata broj mogućih kombinacija iznosi $10! = 3\,628\,800$. Iz navedene kratke analize vidljivo je da treba upotrijebiti algoritam koji ima mogućnost efikasnijeg izračunavanja optimalnog pridruživanja. Jedan od postojećih algoritama koji se koristi u srodnim problemima kombinatorne optimizacije je Kuhn-Munkresov algoritam [85] (poznat pod nazivom Mađarska metoda). Kuhn-Munkresov algoritam upotrebljava se za određivanje optimalnog načina dodjeljivanja n elemenata na n mjesta, i to tako da svaki element bude pridružen samo jednome mjestu. Svako pridruživanje ima svoj trošak ili vrijeme potrebno za obavljanje određene radnje koja je vezana za pridruživanje (npr. obavljanje posla na stroju). Potrebno je pronaći rješenje koje minimizira trošak izvođenja cijelog skupa radnji tj. aktivnosti. Modelom teorije grafova problem se definira pronalaženjem najmanjeg troška savršenog sparivanja težinskog bipartitnog grafa [86]. Problem pridruživanja poseban je slučaj problema linearnog programiranja, te se može rješavati pomoću algoritama koji rješavaju te općenite slučajeve, no zbog specifičnosti ovog problema, postoje efikasniji algoritmi koji ga rješavaju. Mađarska metoda jedan je od najpoznatijih algoritama kombinatorne optimizacije za rješavanje linearnog problema pridruživanja u polinomnom vremenu [87].

Za matrice $\overline{\mathbf{Q}}_{1,ij}^2$ i $\overline{\mathbf{Q}}_{1,ij}^3$ optimalno preslikavanje korištenjem Mađarske metode iznosi:

3. Povezivanje neuređenog prostornog stanja s uređenim

$$\mathbf{F}_1^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{F}_1^3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix},$$

pri čemu jedinice označavaju pridružene objekte. Pripadajuće vrijednosti kumulativne sličnosti iznose $s_1 = \sum Q_{ij} = 1,61$ i $s_2 = \sum Q_{ij} = 1,56$. Kao izlazna struktura izabire se klasa II. Na slici (Slika 3-5.) shematski je prikazano pridruživanje pojedinačnih objekata iz neuređenog prostornog stanja u postojeću klasu II.



Slika 3-5. *Skica prostornog preslikavanja neuređenog stanja objekata u uređeno.*

Jedan od mogućih slučajeva javlja se kada broj objekata na sceni, tj. u radnom prostoru robota nije identičan broju objekata koji se nalaze u prethodno klasificiranim strukturama. U tom je slučaju moguće odabrati dva načina rada sustava. U oba načina rada potrebno je dodati fiktivne objekte kako bi se broj izjednačio te kako bi bilo moguće izračunati optimalno pridruživanje pomoću Mađarske metode. U slučaju kada je broj objekata na sceni veći od broja objekata u postojećoj strukturi:

- korisnik izabire od kojih će se objekata izgraditi struktura. U daljnjem se postupku primjenjuje prethodno opisan postupak za pronalaženje optimalnog pridruživanja.
- unutar matrice sličnosti dodaju se pomoćni (fiktivni) objekti kako bi se omogućila konstrukcija kvadratne matrice, vrijednost sličnosti pomoćnog objekta J te objekta iz postojeće klasificirane strukture i iznosi $H_{iJ} = 0$. Na taj se način iz ulaznog skupa objekata korištenjem Mađarske metode pronalazi skup onih objekata koji rezultira najvećom sličnošću.

Višak objekata u slučaju a) i slučaju b) ne upotrebljava se za izgradnju strukture, pa ostaju neiskorišteni. U slučaju kada je broj objekata u postojećoj strukturi veći od broja objekata na sceni, upotrebljavaju se svi objekti, i to prema sljedećem postupku:

3. Povezivanje neuređenog prostornog stanja s uređenim

- a) Unutar matrice sličnosti dodaju se fiktivni objekti u redove matrice, analogno prema prethodno opisanom postupku. Iz kvadratne matrice računa se optimalno preslikavanje objekata te se svi objekti sa scene preslikavaju na postojeće objekte u strukturi, dok se na preostale objekte u strukturi preslikavaju fiktivni objekti.

Cjelokupni proces preslikavanja neuređenog prostornog stanja na postojeće klasificirane strukture prikazan je algoritmom (Algoritam 3-1.).

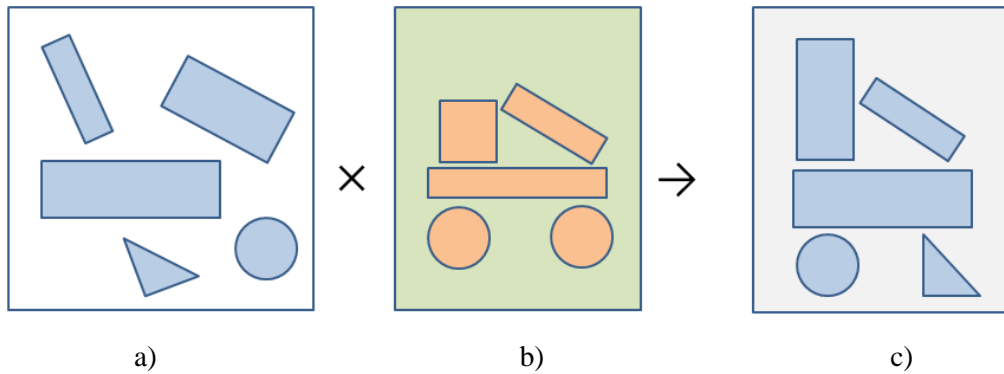
Algoritam 3-1. *Preslikavanje neuređenog prostornog stanja na klasificirane strukture.*

Algoritam: Povezivanje_prostornih_stanja

```

Ulaz:  $I_{MTO}$ ,  $\mathcal{W}$ ,
Izlaz:  $I_{WIN}$ ,  $klasa\_rb1$ ,  $klasa\_rb2$ ,  $\mathbf{P}$ ,  $\mathbf{K}$ 
1 // Računaj preklapanje objekata u  $I_{MTO}$  i objekata u  $\mathcal{W}$ .
2 za svaki  $\mathcal{W}(k)_{nivo1} \in \mathcal{W}$ 
3     za svaki  $W_i \in \mathcal{W}_{nivo1}$ 
4         // Računaj preklapanje.
5          $preklop_i \leftarrow \text{preklop}(W_i(MTO), I_{MTO})$  prema izrazu (3.1)
6         Ako  $preklop_i > \max(\text{preklop})$  onda
7             // Zapamti pobjedničke klase.
8              $klasa_{nivo1} \leftarrow k, klasa_{nivo2} \leftarrow i$ 
9         kraj
10    kraj
11 kraj
12 // Generiraj strukture od pobjedničkih klasa(nivo1).
13 za svaki  $W\{klasa1\}\{klasa2\} \in \mathcal{W}, j=j+1$  // za svaku pobjedničku klasu
14     // generiraj strukturu s objektima iz  $I_{MTO}$ , a koordinatama iz
15     //  $W\{rb1\}\{rb2\}$ .
16     za svaki  $O_i \in I_{MTO}$ 
17         // Pronađi na koji objekt se preslikao.
18         postavi  $O_i$  na najbliži slobodni položaj iz  $W_i(MTO) \rightarrow U_{temp}$ 
19          $U_j \leftarrow U_j + U_{temp}$ 
20     kraj
21     // apliciraj generiranu strukturu  $U$ 
22      $(rb1, rb2, m_j) \leftarrow (U_j/W)$  // Prema algoritmu Pretraži_klase( $\beta_1, \beta_2 \leftarrow 0$ )
23 kraj
24  $J \leftarrow \max(m_j)$  // Pobjednička klasa.
24 učenje pobjedničke klase  $W\{rb1_j\}\{rb2_j\}$  prema (2.4) s  $\beta_1, \beta_2$ 
24 Računaj matricu cijena  $\mathbf{C}$  i matricu pridruživanja  $\mathbf{F}$ 
25 Prema  $\mathbf{F}$  izvrši pridruživanje objekata iz  $\mathbf{P} \rightarrow \mathbf{K}$ 
```

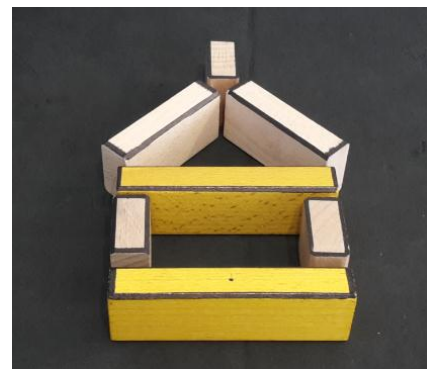
Na slici (Slika 3-6.) prikazan je postupak generiranja strukture c) iz neuređenog stanja objekata a) korištenjem informacija o položaju objekata iz postojeće strukture b).



Slika 3-6. *Skica prostornog preslikavanja neuređenog stanja objekata u uređeno.*

3.2. Funkcija preslikavanja

Nakon izbora optimalnog pridruživanja objekata sa scene (radnog prostora robota) \mathbf{P} te objekata iz postojeće izlazne klase \mathbf{K} postoji mogućnost da se pojedini objekti ponavljaju u strukturi \mathbf{P} i \mathbf{K} , kao što je slučaj na slici (Slika 3-7.). Za prikazani slučaj moguće je generirati 24 ($3! \times 2! \times 2!$) različitih preslikavanja jer se određeni objekti ponavljaju te ih je moguće iz početnog stanja u konačno postaviti na više različitih mjesta. U slučaju kada se u radnom prostoru nalazi deset identičnih objekata moguće je generirati 10! odnosno 3 628 800 različitih preslikavanja. Potrebno je odrediti jedinstveno preslikavanje koje će minimizirati ukupnu sumu putova preko svih objekata od početnog položaja do konačnoga koristeći se Euklidijevom udaljenošću.



Slika 3-7. *Primjer početnog (lijevo) i konačnog (desno) stanja za koje treba odrediti preslikavanje pojedinačnih predmeta iz početne u konačnu strukturu.*

Odredit će se matrica cijena \mathbf{C} koja sadrži kao element C_{ij} vrijednost Euklidijeve udaljenosti s mjesta i u početnom stanju na mjesto j u konačnom stanju. Za mjesta C_{ij} gdje se vrste objekata u \mathcal{P} i \mathcal{K} razlikuju upisuje se visoka vrijednost udaljenosti kako ti redovi ne bi bili izabrani pri traženju optimalnog rasporeda. Matrica cijena \mathbf{C} za strukture sa slike (Slika 3-7.) iznosi:

3. Povezivanje neuređenog prostornog stanja s uređenim

$$\mathbf{C} = \begin{bmatrix} 72 & 133 & 999 & 999 & 999 & 999 & 999 \\ 47,4 & 29 & 999 & 999 & 999 & 999 & 999 \\ 999 & 999 & 44,9 & 111 & 111,9 & 999 & 999 \\ 999 & 999 & 109,5 & 169,9 & 87 & 999 & 999 \\ 999 & 999 & 112,6 & 42,7 & 53,3 & 999 & 999 \\ 999 & 999 & 999 & 999 & 999 & 61,5 & 22 \\ 999 & 999 & 999 & 999 & 999 & 80,6 & 47,5 \end{bmatrix}.$$

Vidljivo je da je, kao i u poglavlju 3.1., potrebno pronaći minimalnu sumu putova optimalnom kombinacijom pridruživanja pojedine vrste objekata iz \mathbf{P} u \mathbf{K} . Primjenom Mađarske metode moguće je odrediti optimalno pridruživanje tj. ono pridruživanje koje kao rezultat daje najmanju vrijednost funkcije cilja, pri čemu je matrica pridruživanja \mathbf{F} dana s:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Cilj ovog koraka je priprema uređenih skupova objekata \mathcal{P} i \mathcal{K} koji će se upotrijebiti u procesu planiranja robotskog djelovanja, čime se generira jedinstveno preslikavanje pojedinačnih objekata iz \mathcal{P} u \mathcal{K} . Uređeni skup objekata označava da se u istim redovima matrice \mathbf{P} i \mathbf{K} nalaze objekti iste vrste koje treba iz početnog stanja P_i transformirati u konačno K_j . U tablici (Tablica 3-1.) nalazi se prikaz matrice \mathbf{P} $\{\text{Objekt}_1, x_1, y_1, \delta_1\}$ te matrice \mathbf{K} $\{\text{Objekt}_2, x_2, y_2, \delta_2\}$ u inicijalnom slučaju.

Tablica 3-1. Inicijalno pridruživanje objekata iz početnog stanja u konačno.

| Objekt ₁ | x_1 | y_1 | δ_1 | Objekt ₂ | x_2 | y_2 | δ_2 |
|---------------------|-------|-------|------------|---------------------|-------|-------|------------|
| 1 | 96 | 161 | -4 | 1 | 168 | 164 | 0 |
| 2 | 210 | 186 | 22 | 2 | 229 | 164 | 0 |
| 3 | 160 | 99 | -80 | 3 | 198 | 123 | 91 |
| 4 | 234 | 228 | 7 | 4 | 75 | 168 | -89 |
| 5 | 144 | 221 | 93 | 5 | 198 | 205 | -92 |
| 6 | 184 | 144 | 23 | 6 | 123 | 136 | 44 |
| 7 | 137 | 153 | -15 | 7 | 125 | 199 | -50 |

3. Povezivanje neuređenog prostornog stanja s uređenim

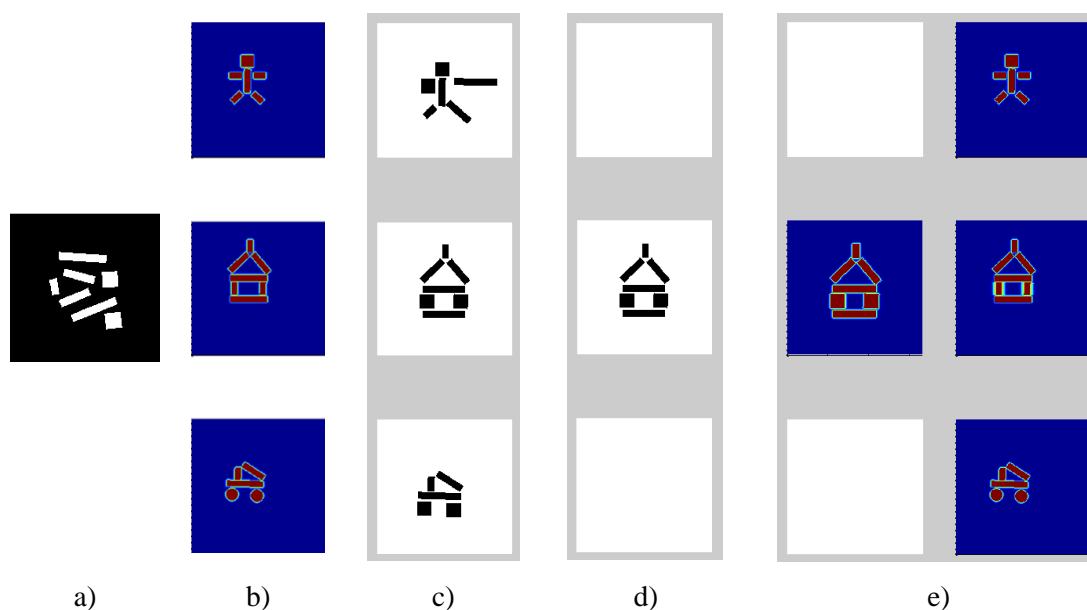
U tablici (Tablica 3-2.) prikazana je permutacija matrice \mathbf{K} prema dobivenoj matrici pridruživanja \mathbf{F} . Funkcija cilja tj. minimalna vrijednost matrice \mathbf{C} iznosi 378,4, dok za inicijalni slučaj prikazan tablicom (Tablica 3-1.) iznosi 481,4.

Tablica 3-2. *Optimalno pridruživanje objekata iz početnog stanja u konačno prema \mathbf{F} .*

| Objekt ₁ | x_1 | y_1 | δ_1 | Objekt ₂ | x_2 | y_2 | δ_2 |
|---------------------|-------|-------|------------|---------------------|-------|-------|------------|
| 1 | 96 | 161 | -4 | 1 | 168 | 164 | 0 |
| 2 | 210 | 186 | 22 | 2 | 229 | 164 | 0 |
| 3 | 160 | 99 | -80 | 3 | 198 | 123 | 91 |
| 4 | 234 | 228 | 7 | 5 | 198 | 205 | -92 |
| 5 | 144 | 221 | 93 | 4 | 75 | 168 | -89 |
| 6 | 184 | 144 | 23 | 7 | 125 | 199 | -50 |
| 7 | 137 | 153 | -15 | 6 | 123 | 136 | 44 |

3.3. Primjer povezivanja neuređenog stanja objekata s uređenim

Kao završni postupak potrebno je implementirati cjelokupnu funkcionalnost unutar ART neuronske mreže. Za objekte na sceni kreira se virtualni prikaz, kao što je prikazano na slici (Slika 3-8.a). Za navedeno neuređeno stanje objekata računa se funkcija preslikavanja na postojeće strukture prikazane slikom (Slika 3-8.b) prema izrazu (3.1). Za pojedinu matricu sličnosti \mathbf{Q}^i računa se matrica preslikavanja $\overline{\mathbf{Q}}^i$ za svaku postojeću klasu i . Prema algoritmu (Algoritam 3-1.) generiraju se hipoteze o mogućim izlaznim klasama, kao što je prikazano na slici (Slika 3-8.c).

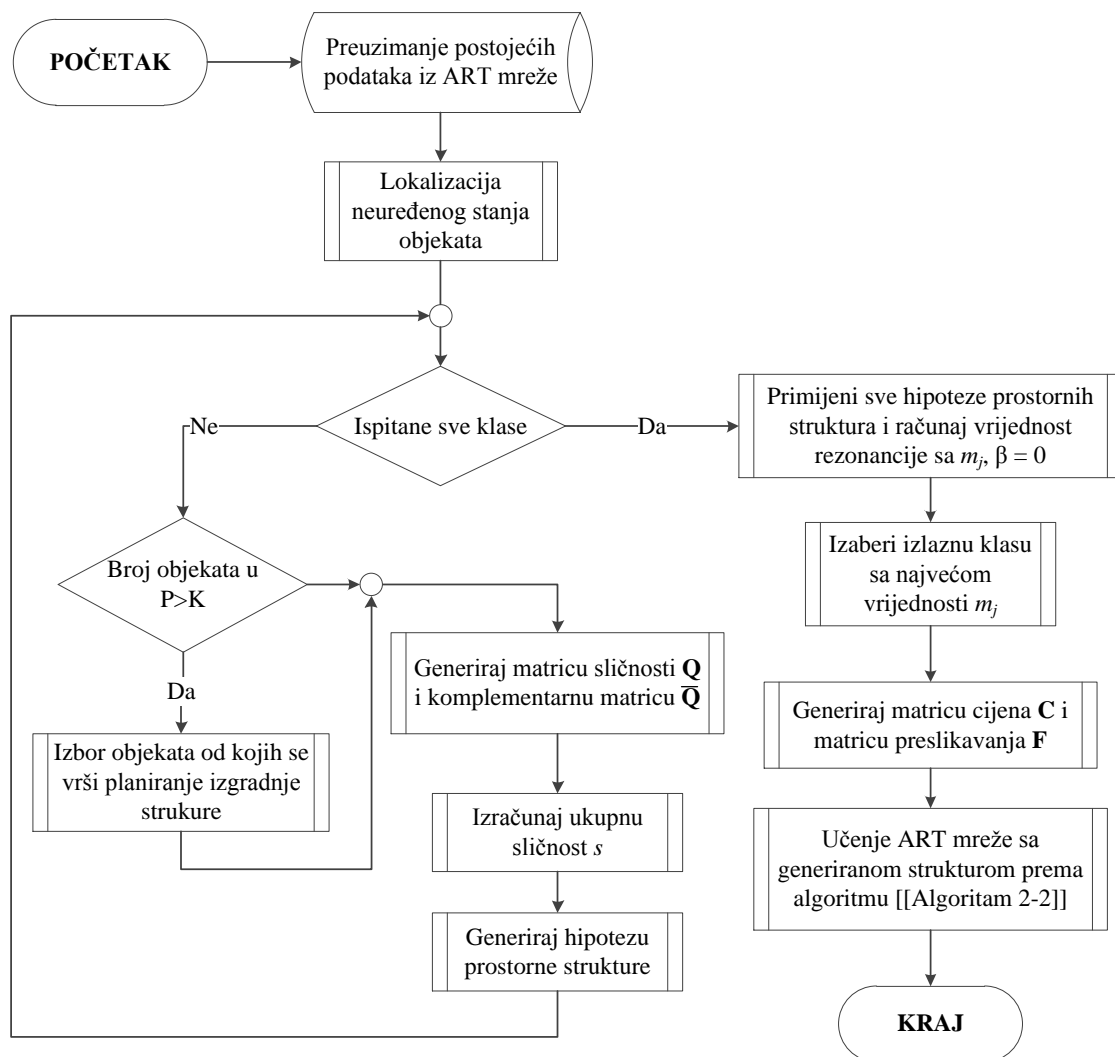


Slika 3-8. *Povezivanje neuređenog prostornog stanja s poznatim klasificiranim strukturama.*

3. Povezivanje neuređenog prostornog stanja s uređenim

Svaka od generiranih struktura, tj. svaka od postavljenih hipoteza ima određenu sličnost s već postojećim klasama koja se računa prema izrazu (2.8). Na slici (Slika 3-8.d) prikazana je pobjednička struktura koja se u sljedećem koraku aplicira na ART mrežu, dok je u sljedećem poglavlju opisan princip planiranja robotskog djelovanja za izgradnju prostornih struktura.

Dijagram toka na slici (Slika 3-9.) prikazuje cjelokupni proces povezivanja neuređenog prostornog stanja s postojećim klasificiranim strukturama. U posebnom slučaju, kada su objekti na sceni identični objektima u postojećoj strukturi tj. kada vrijedi $\mathcal{P} \equiv \mathcal{K}$, moguće je izravno izabrati izlaznu klasu jer je ukupna sličnost jednaka 1.



Slika 3-9. Dijagram toka procesa povezivanja neuređenog prostornog stanja s postojećim klasificiranim strukturama.

4.PLANIRANJE ROBOTSKOG DJELOVANJA

U osnovi, planiranje djelovanja (planiranje) moguće je definirati kao proces kojim se obavlja izbor te organizacija radnji i zadataka softverskog ili hardverskog agenta. U procesu planiranja svaka radnja ima očekivani tj. pretpostavljeni ishod. Proces planiranja najčešće je računalno intenzivan, te se pri planiranju kao cilj postavlja pronalaženje izvedivog i dovoljno dobrog plana [88,89].

Planiranje kao područje koje se može rješavati metodama umjetne inteligencije (engl. *artificial intelligence*) moguće je prema krajnjem cilju podijeliti na sljedeće [90]:

- Planiranje putanje i gibanja (engl. *path and motion planning*) bavi se sintezom geometrijske putanje od početne do završne točke u prostoru [91], kao i generiranjem trajektorije koja opisuje stanja upravljačkih varijabli u konfiguracijskom prostoru [92].
- Percepcijsko planiranje (engl. *perception planning*) bavi se generiranjem planova koji uključuju prikupljanje informacija. Ova vrsta planiranja vidljiva je u problemima modeliranja okoline ili objekata, njihovoj identifikaciji i lokalizaciji, te generalno identifikaciji trenutnog stanja okoline.
- Navigacijsko planiranje (engl. *navigation planning*) uključuje dva spomenuta tipa planiranja sa svrhom uspješnog dolaska do cilja ili istraživanja okoline.
- Planiranje manipulacije (engl. *manipulation planning*) odnosi se na rukovanje objektima te stvaranje i izgradnju sklopova.
- Komunikacijsko planiranje (engl. *communication planning*) odnosi se na višeagentske sustave (MAS) [93]. Bavi se problemima prijenosa informacija tj. kada i kako zatražiti potrebnu informaciju te koju povratnu informaciju treba pružiti.
- Postoji i niz drugih vrsta planiranja, no ona najčešće nisu primjenjiva u tehničkim sustavima [90].

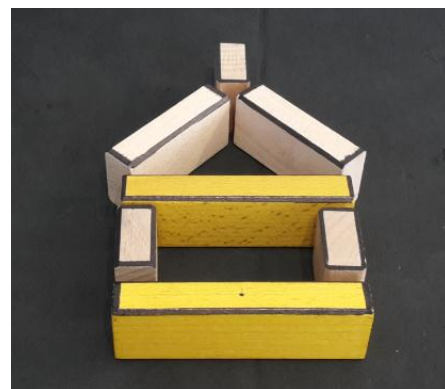
Problem planiranja opisan u ovom poglavlju zasniva se na dvjema prethodno opisanim podjelama: planiranju manipulacije i planiranju kretanja. Informacije o povezivanju neuređenog prostornog stanja s postojećim klasificiranim strukturama dobiveno je kao izlaz iz ART mreže opisane u poglavlju 2.

Kao osnovni cilj planiranja djelovanja robota, ističe se generiranje niza aktivnosti koji će svojim slijednim izvršavanjem dovesti do promjene stanja okoline: iz početnog (percipiranog) stanja u odabrano konačno stanje.

4.1. Neformalni opis problema planiranja

Robotsko djelovanje opisano je kao uređeni skup elemenata rada koji svojim slijednim izvršavanjem transformiraju početno stanje objekata u konačno stanje koje je određeno preko nove funkcije preslikavanja, opisane u poglavlju 3.

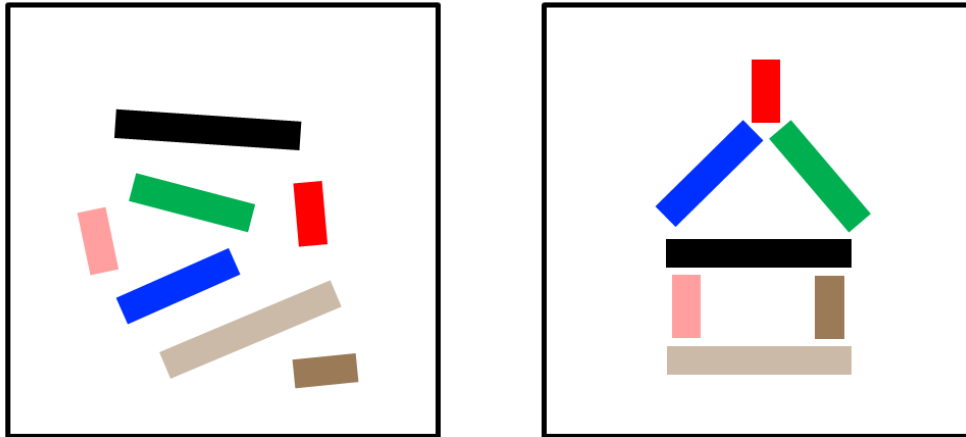
Zadatak planiranja je pronaći mogući slijed radnji robota kojima će se početno stanje okoline, interpretirano kao položaj i orijentacija lokaliziranih objekata, kroz konačan broj koraka preurediti u traženo konačno stanje uz najmanji utrošak robotskog rada. Robotski rad interpretira se kroz utrošeno vrijeme i prijedeni put za obavljanje potrebnih kretnji. Nadalje, utrošeni put moguće je izraziti pomoću utrošenog vremena poznavanjem profila brzine te ubrzanja robotske ruke kao i generirane putanje kretanja, kao što je prikazano izrazom (4.4). Na taj je način moguće optimirati jedan parametar tj. ukupno vrijeme trajanja svih potrebnih koraka za izvršavanje plana djelovanja. Planiranje djelovanja robota kao ulazne parametre uzima dvodimenzionalnu interpretaciju rasporeda objekata u početnom i u konačnom stanju, kao što je prikazano na slici (Slika 4-1.) i (Slika 4-2.).



Slika 4-1. *Primjer početnog (lijevo) i konačnog (desno) stanja za koje je potrebno generirati plan robotskog djelovanja.*

4. Planiranje robotskog djelovanja

Nadalje, problem rasporeda tj. pridruživanja objekata iz \mathcal{P} u \mathcal{K} određen je u poglavlju 3.2. funkcijom preslikavanja primjenom Mađarske metode. Jedinstveno preslikavanje iz \mathcal{P} u \mathcal{K} poslužit će kao polazišna točka određivanja plana kretanja te manipulacije pojedinim objektima.



Slika 4-2. Lijevo: početno stanje objekata \mathcal{P} , desno: konačno stanje objekata \mathcal{K} .

Na slici (Slika 4-1.) i (Slika 4-2.) prikazano je početno stanje (lijevo) i konačno stanje (desno) objekata u radnom prostoru robota. Zadatak modula za planiranje odnosi se na generiranje mogućeg slijeda radnji pomoću kojih će robot u konačnom broju koraka transformirati početno stanje objekata u konačno stanje. Pri generiranju plana moguće je identificirati dva osnovna problema: problem redoslijeda i problem položaja. Problem redoslijeda definira permutaciju (uz mogućnost ponavljanja) koja određuje koji se objekt izuzima i odlaže prvi, koji će biti drugi, itd. Problem položaja (i orijentacije) određuje se za svaki međukorak. Položaj i orijentacija definirani su za krajnji položaj predmeta, no kao što će biti prikazano kasnije (Slika 4-8.), postoji mogućnost da su krajnji položaji zauzeti te da predmete nije moguće izravno postaviti na krajnji položaj. Ovaj problem donekle je sličan problemu zaglavljenja (engl. *deadlock*), gdje nijedan od mobilnih agenata ne može izravno doseći svoj krajnji položaj [94]. U slučaju kada je krajnji položaj zauzet, objekt je moguće odložiti na bilo koju slobodnu lokaciju u radnom prostoru u proizvoljnoj orijentaciji.

Kombinacija svake od permutacija i izbora položaja dovodi do eksplozije prostora mogućih rješenja koja se ne mogu u potpunosti istražiti sa svrhom pronalaska optimalnog rješenja. Za klasu problema gdje nije moguće pronaći optimalno rješenje pomoću egzaktnih matematičkih metoda [95] moguće je prostor stanja pretražiti samo djelomično, gdje je jedan od mogućih pravaca korištenje heuristika. Neke od postojećih heurističkih metoda uključuju evolucijsko

računarstvo, genetske algoritme, algoritme inteligencije roja čestica, lokalne algoritme pretraživanja te druge [96]. Karakteristika svih heurističkih metoda je da one pretražuju samo dio prostora stanja te da mogu pronaći rješenja koja su vrlo blizu optimalnih, no bez formalnih dokaza da su ta rješenja globalno optimalna. Nadalje, prilikom primjene heurističkih metoda pretrage potrebno je napraviti određene kompromise, od kojih su najznačajniji spomenuta optimalnost te dovršenost rješenja i vrijeme izvođenja algoritma. Kompromis optimalnosti: garantira li heuristika najbolje rješenje za zadani problem u slučaju kada postoji više rješenja te je li za zadani problem potrebno pronaći najbolje rješenje. Kompromis dovršenosti: ako postoji više rješenja za navedeni problem, možemo li pomoću heuristike identificirati sva rješenja te jesu li zapravo potrebna sva rješenja [97].

Za planiranje djelovanja robota koje kao ulazni skup parametara uzima početno i konačno stanje objekata u radnom prostoru razvijen je novi genetski algoritam planiranja opisan u narednim poglavljima.

4.2. Definicija problema planiranja

Zadatak (cilj): Odrediti vremenski i prostorno diskretni konačni skup aktivnosti \mathcal{A} nad n poznatih predmeta koji se nalaze u početnoj neuređenoj strukturi \mathcal{P} tako da rezultat bude konačna uređena struktura \mathcal{K} . Konačni skup aktivnosti ne smije rezultirati međusobnim kolizijama predmeta ni u jednoj instanci skupa \mathcal{A} .

Pretpostavke:

- Pretpostavlja se da postoji barem jedno rješenje $\mathcal{A}_i \in \mathcal{A}$
- Pretpostavlja se da se predmeti u konfiguraciji \mathcal{P} i \mathcal{K} se ne nalaze u koliziji tj. nijedan predmet ne preklapa nijedan drugi predmet gledajući iz perspektive vizijskog sustava (pozitivna koordinata na osi z , smjer gledanja prema ishodištu referentnog koordinatnog sustava).
- Radna površina na kojoj se nalaze predmeti je ravnina.
- Svi su predmeti slobodno raspoređeni na radnoj površini
- Prostor i vrijeme mogu se diskretizirati.
- Dodir predmeta nije dozvoljen (zbog kasnije implementacije na realnom robotskom sustavu, poglavlje 5.)

Definicija 2. *Aktivnost* je složeno djelovanje robota koje je opisano izuzimanjem određenog predmeta, podizanjem na zadanu visinu ($z > 0$) u referentnom koordinatnom sustavu te linearno (ili krivuljno) gibanje do krajnje točke, spuštanje predmeta na radnu plohu i otpuštanje. U određenom (konačnom) vremenskom intervalu $t = [t_i, t_{i+j}]$ ($i \geq 0, j > 0, j > i$) moguće je izvršavati isključivo jednu aktivnost. Uslijed prirode problema gdje jedan robot obavlja sve potrebne aktivnosti s jednim izvršnim članom (hvataljkom) nije moguće imati paralelne procese.

4.3. Prijedlog rješenja planiranja djelovanja

Prvi je korak određivanje funkcije cilja koja će omogućiti da se kreiraju inicijalna rješenja. Prema [98] postoje različite preporuke koje se odnose na generiranje početnog rješenja u sklopu GA. Početno rješenje moguće je generirati potpuno slučajno ili je moguće koristiti se određenom heuristikom kako bi već početno rješenje bilo napravljeno u skladu s minimizacijom tražene funkcije cilja. U skladu s problemom koji se rješava minimizirat će se ukupno vrijeme potrebno za izvršavanje svih elemenata rada robota. Početno rješenje generirat će se korištenjem slučajnih mutacija koje imaju svojstvo lokalnog pretraživanja gdje je vjerojatnost mutacije postavljena na najveću vrijednost u području krajnjeg položaja pojedinog objekta. Na taj način moguće je pretražiti cijeli prostor stanja s najvećom vjerojatnosti za pojedini objekt na području krajnjeg položaja iz **K**.

4.3.1. Kreiranje inicijalnog skupa rješenja – konstruktivni algoritam

Svaki je objekt opisan matricom cjelobrojnih vrijednosti, pri čemu je vrijednost pojedinog piksela svakog objekta jedinstvena kako bi ih bilo moguće identificirati u **P** i **K** matrici. Algoritam za traženje rješenja inicijaliziran je sa sljedećim parametrima:

- usporedno kreiranje 30 rješenja (N)
- međusobno uspoređivanje trenutnih rješenja svaka 3 ciklusa (S)
- Odbacivanje jednog od 25% najlošijih rješenja svaka 3 ciklusa te kloniranje jednog od 25% najboljih rješenja.

Usporedno se kreira 30 rješenja kako bi ona poslužila kao inicijalna populacija za genetski algoritam koji će biti opisan u poglavlju 4.3.2. Eksperimentalno je utvrđeno da je za promatranu domenu problema prikladna inicijalna populacija od 30 rješenja. Međusobno uspoređivanje rješenja daje veću vjerojatnost trenutno najboljim rješenjima u svrhu proširenja

domene pretrage. Jedini operator koji se koristi u ovoj fazi je operator mutacije trenutnog rješenja. Cilj mutacije trenutnog rješenja je promjena položaja i orijentacije slučajno odabranog objekta koji se ne nalazi u konačnom položaju. Mjeri se uspješnost izvršene mutacije te se uzima samo ono rješenje koje je dovelo do smanjenja funkcije cilja. Funkcija cilja sastoji se od dviju funkcija koje se evaluiraju u svakom koraku. Prva parcijalna funkcija Φ_1 određena je izrazom:

$$\Phi_1 = \sum_{i=1}^m \sum_{j=1}^n \operatorname{sgn}(|P_{ij} - K_{ij}|) a, \quad a = \begin{cases} 1, & \forall (P_{ij} - K_{ij}) \neq 0 \\ 4, & \forall P_{ij} \neq K_{ij} \neq 0 \\ 0, & \forall (P_{ij} - K_{ij}) = 0 \end{cases} \quad (4.1)$$

Funkcija $\operatorname{sgn}()$ označava matematičku signum funkciju. Optimalna vrijednost $\Phi_1 = 0$, što označava da se svi objekti nalaze na identičnim mjestima u početnom i konačnom stanju. Parametar a određuje vrijednost „kazne“ (engl. *penalty*) ako se neki od objekata u \mathbf{P}^i nalazi na mjestu različitog objekta u \mathbf{K} . Ujedno, određena je kazna kad se objekt u \mathbf{P}^i nalazi na mjestu gdje nema objekta u \mathbf{K} te obrnuto. Ova funkcija omogućava da svaka sljedeća iteracija algoritma konvergira ka rješenju. Uvodi se dodatna funkcija Φ_2 Euklidijske udaljenosti svih objekata u \mathbf{P}^i i \mathbf{K} :

$$\Phi_2 = \sum_{i=1}^l \sqrt{(x_i^P - x_i^K)^2 + (y_i^P - y_i^K)^2} \quad (4.2)$$

Funkcija Φ_2 poprima vrijednost 0 kada su svi objekti iz \mathbf{P}^i na istim položajima kao u \mathbf{K} . Pomoću dvije navedene parcijalne funkcije moguće je odrediti ukupnu funkciju cilja koju je potrebno minimizirati te koja treba poprimiti vrijednost 0 kada se trenutno i konačno stanje preklapaju. Funkcija $\Phi_{1,2}$ dana je izrazom:

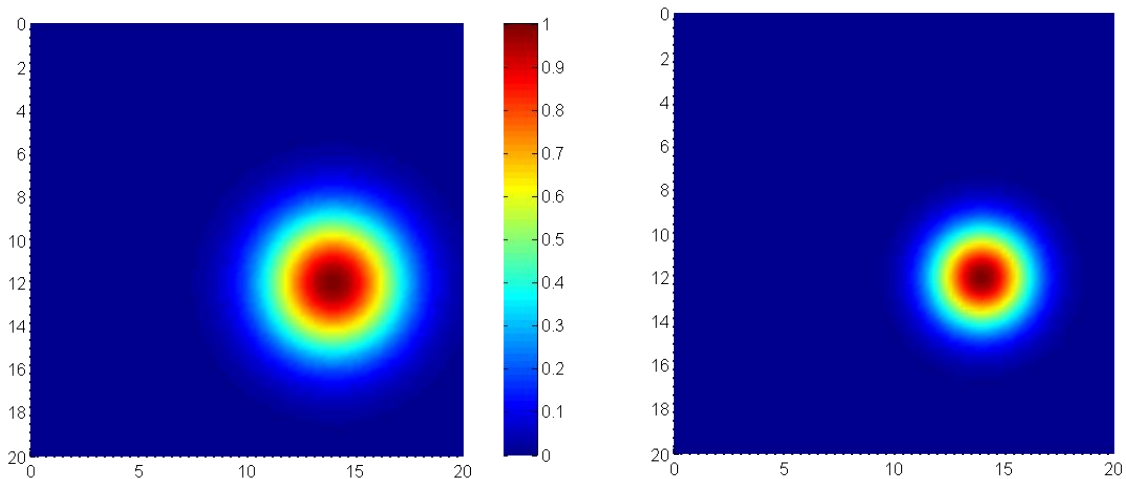
$$\Phi_{1,2} = t_1 \sum_{i=1}^m \sum_{j=1}^n \operatorname{sgn}(|P_{ij} - K_{ij}|) a + t_2 \sum_{i=1}^l \sqrt{(x_i^P - x_i^K)^2 + (y_i^P - y_i^K)^2} \rightarrow \min \quad (4.3)$$

Težinske vrijednosti pojedine funkcije cilja označene su s t_1 i t_2 kako bi se mogao odrediti utjecaj pojedine parcijalne funkcije cilja.

U procesu generiranja rješenja matrica \mathbf{P} mijenja se nizom mutacija (slučajnim pretragama prostora stanja) $\mathbf{P}^0 \rightarrow \mathbf{P}^1 \rightarrow \dots \rightarrow \mathbf{P}^{n-1} \rightarrow \mathbf{K}$ sve dok se završno stanje svih objekata u potpunosti ne

4. Planiranje robotskog djelovanja

podudara s matricom $\mathbf{P}^n = \mathbf{K}$. Iako su mutacije definirane kao slučajno pretraživanje, implementirana je funkcija vjerojatnosti mutacije na određenoj lokaciji. Vjerojatnost mutacije tj. promjene položaja objekta O najveća je u području srednje vrijednosti (x^K, y^K) sa standardnom devijacijom koja je jednaka dimenziji najmanjeg objekta u \mathcal{K} . Gustoća funkcije vjerojatnosti određena je prema normalnoj (Gaussovoj) raspodjeli.



Slika 4-3. Vjerojatnost mutacije oko točke $(12, 14)$. Lijevo – $\sigma = 2a$, desno $\sigma = a$.

Zbog izrazito velikog prostora mogućih rješenja gdje u domeni diskretizacije položaja u radnom području diskretiziranom matricom veličine 100×100 te diskretizacijom kuta rotacije od 1° najmanji broj mogućih rješenja za problem koji uključuje 5 objekata na sceni iznosi $\sim 10^{35}$. Zbog toga se uvodi određeno pojednostavljenje povezano s mogućom rotacijom objekata. Prilikom pomicanja pojedinog objekta položaj se određuje prema jednadžbi vjerojatnosti mutacije, dok se rotacija postavlja na konačnu rotaciju iz \mathbf{K} . Na taj način je moguće isti problem pojednostaviti, pa je sada najmanji broj mogućih rješenja određen s $\sim 10^{22}$.

Ukupni broj tranzicija stanja pojedinog rješenja ne mora biti jednak. Ujedno je redoslijed promjena pojedinih rješenja slučajan, tako da svako od rješenja istražuje prostor mogućih stanja u različitom smjeru. Kreiranje inicijalnog skupa rješenja prikazano je algoritmom (Algoritam 4-1.).

4. Planiranje robotskog djelovanja

Algoritam 4-1. *Konstruktivni algoritam.*

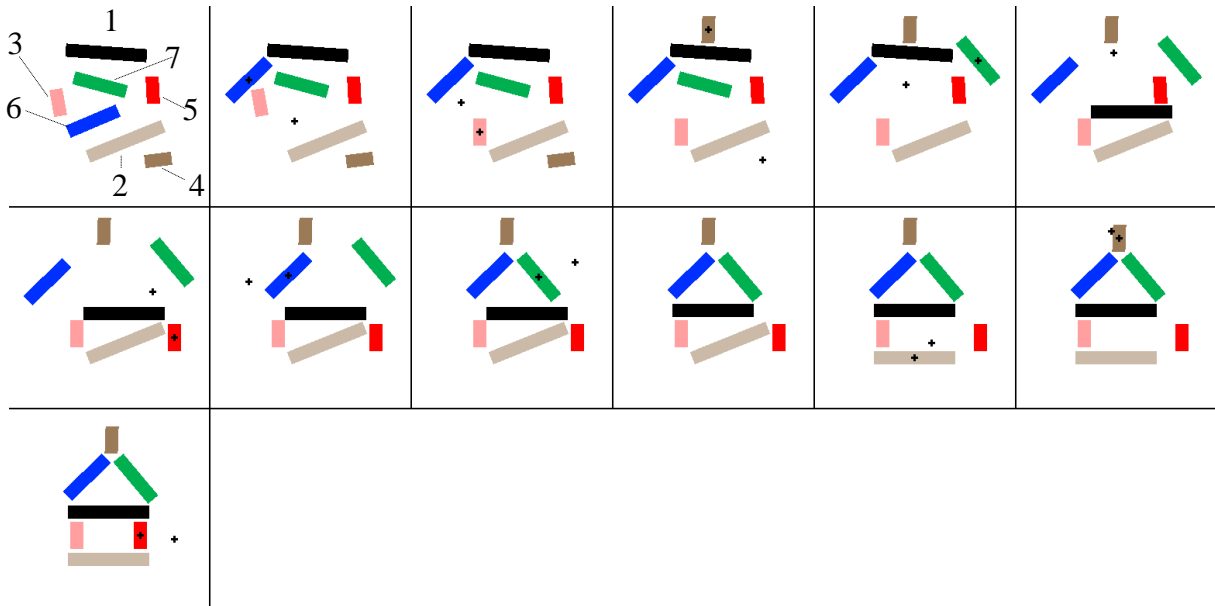
Algoritam: Konstruktivni_algoritam

```
Ulaz: P, K, N, S
Izlaz: PLAN, O
1 // Za koordinate objekata u P treba generirati skup planova djelovanja
  Otakav da  $O: P \rightarrow K$ 
2 Inicijaliziraj b planova prema početnom stanju  $P: O_i \rightarrow O$ 
3 za svaki plan u O, c++
4   |   evaluiraj  $f_{cilja}$  prema (4.3):  $\Phi_i \leftarrow f(O_i)$ 
5   |   Ako  $\Phi_i > 0$  onda // Ako plan nije dovršen  $O_i: P \rightarrow K$ .
6   |   |   slučajno izaberi jedan objekt iz  $O_i$  koji se ne nalazi na konačnom mjestu iz  $\rightarrow O_i(j)$ 
7   |   |   promijeni položaj  $\Delta x$  i  $\Delta y$  prema s  $\mu_1, \mu_2, \sigma_1$ 
8   |   |   |    $O_i(kraj+1) \leftarrow \text{rand}(O_i(j))$  // Računaj novi plan, mutirani  $O^{novi}(t+1)$ .
9   |   |   |   promijeni  $\varphi \rightarrow \varphi_{konačni}$ 
10  |   |   |   status  $\leftarrow \text{evaluacija\_kolizije}(O_i)$ 
11  |   |   |   ako status = 1, idi na red 7
12  |   |   |   evaluiraj  $f_{cilja}$  prema (4.3):  $\Phi_i^{t+1} \leftarrow f(O_i)$ 
13  |   |   |   ako  $\Phi_i^{t+1} \leq \Phi_i^t$  onda
14  |   |   |   |    $O(i) \leftarrow O_i(t+1)$ 
15  |   |   |   kraj
16  |   |   |   ako  $(\Phi_i^{t+2} = \Phi_i^{t+1} = \Phi_i^t) \wedge (\Phi_i^{t+1} > \Phi_i^t)$  onda
17  |   |   |   |    $O(i) \leftarrow O_i^{t+1}$  // Prihvati lošije rješenje.
18  |   |   |   kraj
19  |   |   |    $LOG\_PUT_i \leftarrow$  zapiši ukupno prijedeni put u planu  $O_i$ 
20  |   |   inače
21  |   |   |   plan  $O_i$  je završen
22  |   |   kraj
23  |   |   ako  $\max(\Phi_i) = 0$  onda
24  |   |   |   prekid_programa // Svi planovi su završeni.
25  |   |   kraj
26  |   |   ako  $\text{mod}(c, b) = 0$  onda
27  |   |   |   // Ukidanje loših rješenja izvršavaj svaki b ciklus.
28  |   |   |   B  $\leftarrow O$  // Izaberi jedno loše nedovršeno rješenje iz O prema
29  |   |   |   |   turnirskoj selekciji s elitizmom.
30  |   |   |   G  $\leftarrow O$  // Izaberi jedno dobro nedovršeno rješenje iz O prema
31  |   |   |   |   turnirskoj selekciji s elitizmom.
32  |   |   |   zamijeni B s G u O // Dobro rješenje je klonirano, loše je
33  |   |   |   |   ukinuto.
34  |   |   |   kraj
35  |   |   kraj
36  |   kraj
```

Na slici (Slika 4-4.) prikazan je primjer generiranog rješenja koje se sastoji od ukupno 12 koraka. Konstruktivnim algoritmom dobiveno je 30 rješenja dok prikazano rješenje sadrži 12 diskretnih koraka koji transformiraju početno neuređeno stanje objekata (Slika 4-4. lijevo gore) u uređenu strukturu (Slika 4-4. desno dolje). U svakom od navedenih koraka dolazi do

4. Planiranje robotskog djelovanja

promjene stanja (položaj i orijentacija) jednog od objekata na sceni. Radi bolje vidljivosti, svaki od objekata u **P** i **K** stanju označen je jedinstvenom bojom. Ujedno su na prvoj slici svi objekti označeni (1 – 7) prema oznakama iz plana koji je prikazan u tablici (Tablica 4-1.).



Slika 4-4. Primjer generiranog rješenja koje se sastoji od 12 koraka.

Za navedeni plan određen je uređeni niz instrukcija robotu čijim se slijednih izvršavanjem pojedini objekt oznake *ID* iz početnog stanja $(x_1, y_1, \varphi_1)^t$ dovodi u sljedeće stanje $(x_2, y_2, \varphi_2)^{t+1}$. Prvi položaj uvijek je određen kao položaj izuzimanja, dok je drugi položaj odlaganja. Niz koraka prikazan je u tablici (Tablica 4-1.).

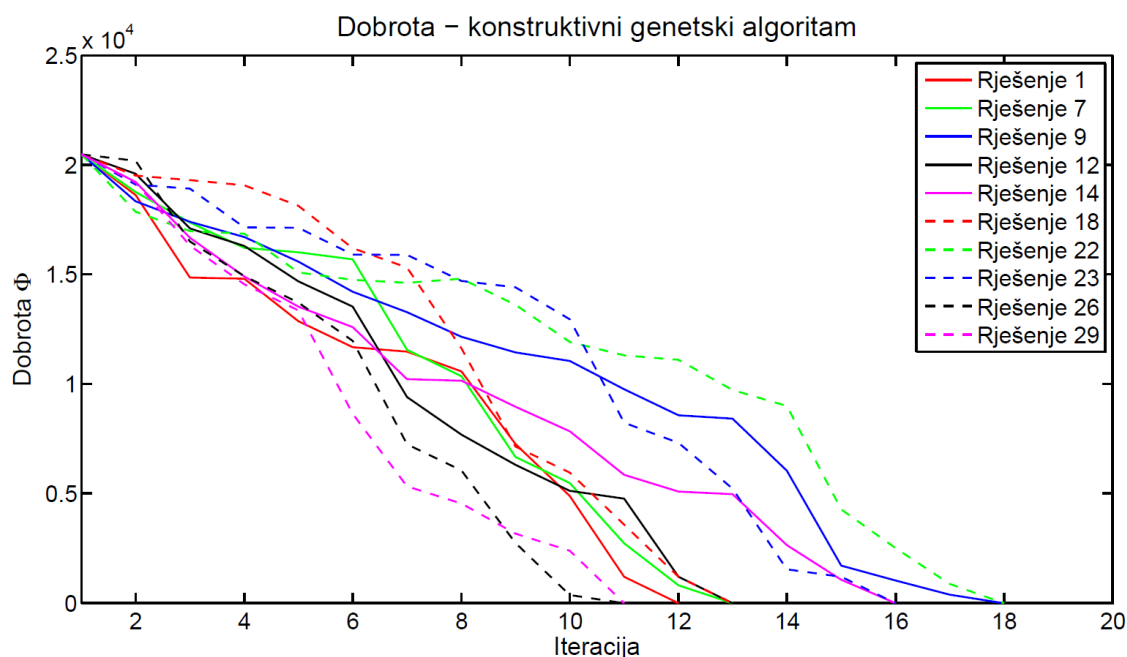
Tablica 4-1. Niz instrukcija generiranog plana prikazanog na slici (Slika 4-4.).

| Korak | Objekt | x_1 | y_1 | δ_1 | x_2 | y_2 | δ_2 |
|-------|--------|-------|-------|------------|-------|-------|------------|
| 1 | 6 | 184 | 144 | 23 | 131 | 85 | 44 |
| 2 | 3 | 160 | 99 | -80 | 198 | 123 | 91 |
| 3 | 4 | 234 | 228 | 7 | 66 | 158 | -89 |
| 4 | 7 | 137 | 153 | -15 | 106 | 246 | -50 |
| 5 | 1 | 96 | 161 | -4 | 172 | 184 | 0 |
| 6 | 5 | 144 | 221 | 93 | 203 | 249 | -92 |
| 7 | 6 | 131 | 85 | 23 | 123 | 136 | 44 |
| 8 | 7 | 106 | 246 | -15 | 125 | 199 | -50 |
| 9 | 1 | 172 | 184 | -4 | 168 | 164 | 0 |
| 10 | 2 | 210 | 186 | 22 | 229 | 164 | 0 |
| 11 | 4 | 66 | 158 | 7 | 75 | 168 | -89 |
| 12 | 5 | 203 | 249 | 93 | 198 | 205 | -92 |

4. Planiranje robotskog djelovanja

Plan djelovanja moguće je zapisati kao permutaciju redoslijeda koja određuje redoslijed manipulacije s pojedinačnim objektima $\mathbf{p} = (101, 103, 102, 103, 101, 103)$. Uz vektor permutacije redoslijeda pridružen je vektor položaja i orijentacije $([x_1, y_1, \varphi_1], [x_2, y_2, \varphi_2], \dots, [x_n, y_n, \varphi_n])$.

Postupak generiranja inicijalnog skupa rješenja opisan algoritmom (Algoritam 4-1.) podrazumijeva usporedno kreiranje određenog broja rješenja. Za probleme razmatrane u okviru ovog istraživanja generira se 30 usporednih planova pomoću konstruktivnog algoritma. U svakom se koraku algoritma pomiče se slučajno odabrani objekt, pa zbog toga dijagram prikaza dobrote rješenja (Slika 4-5.) za pojedini plan ima različitu vrijednost dobrote u svakom od koraka. Dobrota rješenja evaluira se prema izrazu (4.3). Na dijagramu (Slika 4-5.) prikazano je 10 slučajno odabranih rješenja. Može se vidjeti da je dužina najkraćeg generiranog plana 11 koraka (iteracija), a najdužeg plana 18 koraka (iteracija).

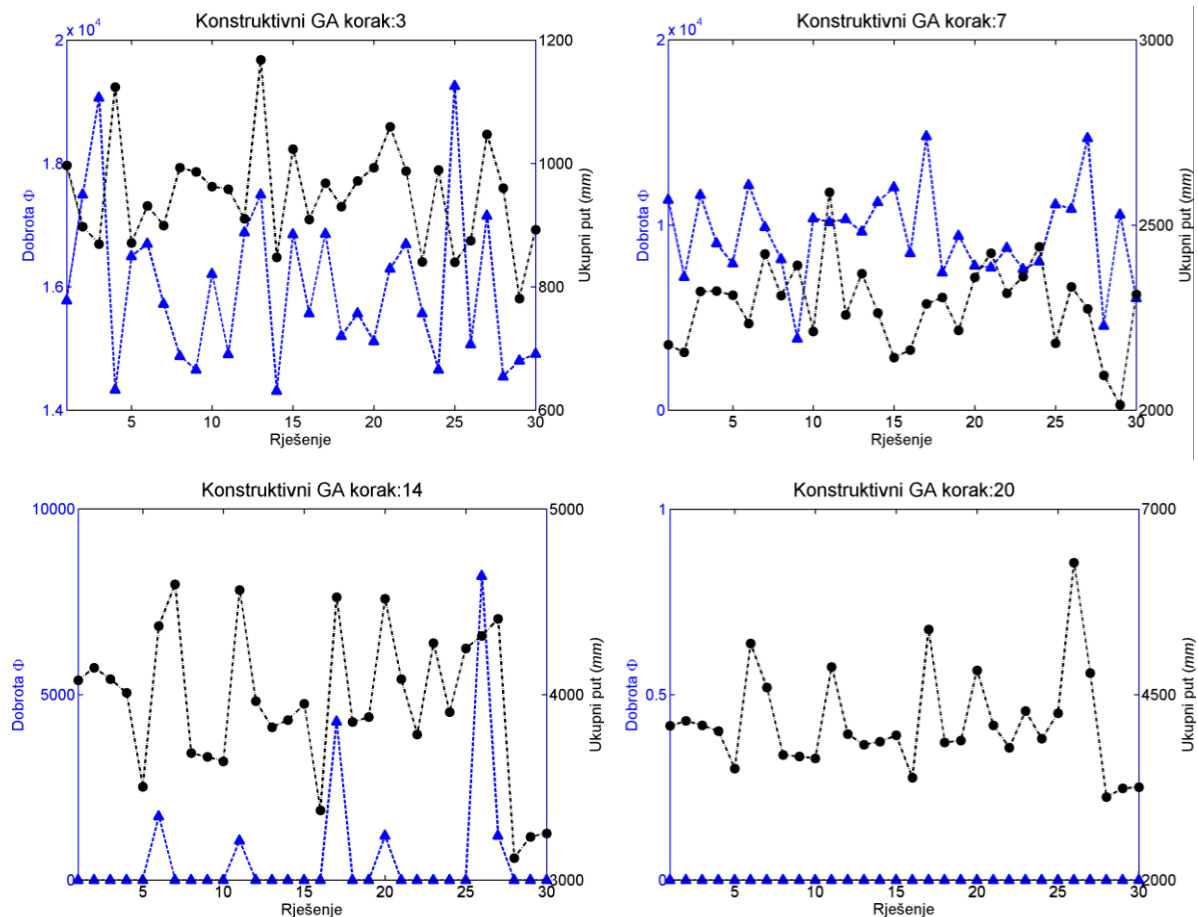


Slika 4-5. Usporedba dobrote 5 različitih rješenja dobivenih konstruktivnim genetskim algoritmom evaluacijom funkcije dobrote iz jednadžbe (4.3).

U početnom je koraku dobrota svih rješenja jednaka jer se sva rješenja počinju usporedno razvijati iz istog početnog stanja prikazanog slikama (Slika 4-1.) i (Slika 4-2.).

Dijagrami na slici (Slika 4-6.) prikazuju dobrotu parcijalnih planova u određenim koracima algoritma: korak 3, 7, 14 i 18. Ujedno je prikazan i ukupni prijedeni put za pojedini plan.

4. Planiranje robotskog djelovanja



Slika 4-6. Prikaz 4 diskretna koraka pri generiranju rješenja pomoću konstruktivnog genetskog algoritma. Prikazani su koraci 3, 7, 14 i završni korak 18, kada je generirano posljednje rješenje.

Ukupna dobrota pojedinog rješenja, koja je prikazana u domeni prijednog puta, može se izraziti putem utrošenog vremena za izvođenje plana na realnom ili virtualnom (simuliranom) robotskom sustavu. Kao što je vidljivo iz generiranog plana, dostupni su pomaci koji se mogu izraziti pomoću Euklidijske udaljenosti. Za svaki od navedenih koraka dodaje se i vrijeme trajanja prilaza i odmaka (engl. *approach and retract*). Određivanje vremena potrebnog za izvršavanje određenog gibanja na robotskoj ruci napravljeno je eksperimentalno za kontinuirane linearne brzine u rasponu od 100 do 500 mm/s, što je prikazano u tablici (Tablica 4-2.). Određeno je vrijeme T_1 za prilazak i odmak, koje uzima u obzir procesno vrijeme od 0,5 s potrebno da se predmet prihvati pomoću vakuumske hvataljke, kao i vrijeme T_2 , koje određuje potrebno vrijeme akceleracije te deceleracije pri linearnom gibanju (trapezni oblik profila brzine) robotske ruke. Vrijeme T_2 neovisna je o brzini gibanja robotske ruke, što je vidljivo iz tablice (Tablica 4-2.). Kako je jedina konstanta koju je moguće definirati u upravljačkoj jedinici robota u eksperimentalnom postavu linearna brzina, svi su proračuni

4. Planiranje robotskog djelovanja

izrađeni za različite linearne brzine robota v . Linearna brzina robota je brzina gibanja koja se zadaje robotu korištenom u postupku eksperimentalne validacije koji će biti opisan u poglavlju 5. Vrijeme T_1 i T_2 za brzine unutar testiranog raspona, a različite od izmjerenih, računat će se linearnom interpolacijom pomoću dviju najbližih izmjerenih vrijednosti. U kasnijim će eksperimentima brzina robotske ruke biti postavljena u intervalu [100, 500] mm/s.

Tablica 4-2. Izmjerena vremena T_1 i T_2 linearnih gibanja robotske ruke.

| Brzina v | 100 mm/s | 200 mm/s | 300 mm/s | 400 mm/s | 500 mm/s |
|------------|----------|----------|----------|----------|----------|
| T_1 | 2,36 | 1,6 | 1,36 | 1,23 | 1,15 |
| T_2 | 0,16 | 0,16 | 0,16 | 0,16 | 0,16 |

Pomoću jednadžbe (4.4) pojedini je plan moguće transformirati iz domene prijednog puta i izvršenih rotacija u duljinu plana izraženog u vremenskoj domeni za postojeću brzinu gibanja robota v .

$$T_{plan} = \sum_{i=1}^n \left(T_1(v) + \frac{\sqrt{((x_i^P - x_i^K)^2 + (y_i^P - y_i^K)^2)}}{v} + T_2(v) \right) \quad (4.4)$$

Naredni korak opisan u poglavlju 4.3.2. tijekom procesa generiranja plana djelovanja robota odnosi se na optimiranje planova. Moguće je optimirati vektor redoslijeda tj. permutaciju redoslijeda te pridruženi vektor položaja i orijentacije pojedinog objekta. Kao osnovni kriterij optimalnosti određen je minimalni utrošak vremena.

4.3.2. Genetski algoritam

Rješenje koje je dobiveno u prethodnoj fazi moguće je prikazati kao permutaciju redoslijeda. Permutacija određuje kojim će se redoslijednom izuzimati i odlagati objekti s određenošnog na konačni ili određeni međupoložaj. Generirani planovi dobiveni konstruktivnim algoritmom iz poglavlja 4.3.1. predstavljaju inicijalnu populaciju genetskog algoritma. Kao što je bilo napomenuto ranije, tijekom izrade genetskih algoritama inicijalna populacija rješenja može biti dobivena potpuno slučajno ili se može upotrijebiti određena heuristika koja će kao inicijalnu populaciju dati pojedince veće dobrote od one dobivene potpuno slučajno. U razmatranom slučaju dobrotu rješenja mjeri se kao vrijeme trajanja izvođenja pojedinog plana s ciljem minimizacije utrošenog vremena prema jednadžbi (4.4). Slučajno generirano rješenje

najčešće ne daje izvedive planove jer postoje ograničenja pri koliziji objekata. Potpuno slučajan izbor položaja te orijentacije najčešće je izrazito nepovoljan i rezultira kolizijama, pa su zbog toga upotrijebljene heuristike, što je vidljivo i u rješavanju srodnih problema [99]. Heuristika koja je upotrijebljena za generiranje inicijalnog skupa rješenja pomoću konstruktivnog algoritma opisana je u prethodnom poglavlju. Glavne karakteristike su lokalna pretraga pomoću mutacije s najvećom vjerojatnosti u području konačnog položaja objekta, kao što je prikazano na slici (Slika 4-3.).

4.3.2.1. Operatori mutacije

U okviru genetskog algoritma primijenjuju se tri operatora mutacije permutacija koji se temelje na standardnim operatorima „izuzmi i ubaci” (engl. *remove and reinsert*) [100], zamijeni (engl. *swap*) [101] i invertiraj (engl. *invert*) [102]. Svaki od navedenih operatora modificiran je kako bi se mogao upotrijebiti za mutaciju permutacije plana djelovanja robota. Zbog prirode problema, gdje pojedina permutacija može sadržavati više ponavljanja istog objekta, njihove mutacije ne mogu biti potpuno slučajne, već je svaku mutaciju potrebno u određenoj mjeri ograničiti na parcijalnu permutaciju. Primjer djelovanja originalnih operatora mutacija „izuzmi i ubaci” (M1), zamijeni (M2) te invertiraj (M3) kao i njihovih prilagođenih varijanti u ovom radu ($M1^1$, $M2^1$, $M3^1$) prikazan je u nastavku teksta. Za zadanu permutaciju koja nema ponavljajuće elemente $p = (1\ 2\ 3\ 4\ 5\ 6\ 7)$ slučajnim odabirom izabiru se dvije pozicije: 2 i 5. Tada svaki od navedenih operatora mutacije M1-M3 daje sljedeće permutacije:

$$p(M1) = (1\ 3\ 4\ 5\ 2\ 6\ 7), \quad (4.5)$$

$$p(M2) = (1\ 5\ 3\ 4\ 2\ 6\ 7), \quad (4.6)$$

$$p(M3) = (1\ 5\ 4\ 3\ 2\ 6\ 7). \quad (4.7)$$

U slučaju permutacije s ponavljajućim elementima $p = (1\ 2\ 3\ 4\ 5\ 2\ 6\ 7)$, potrebno je za pojedini operator mutacije $M1^1$ - $M3^1$ izabrati takvu parcijalnu permutaciju s početnim elementom $p(i)$ i završnim $p(j)$ koja zadovoljava sljedeće uvjete:

- Za operator $M1^1$: $p(k) \neq p(i), \forall k \ i < k < j$
- Za operator $M2^1$: $p(k) \neq p(i) \neq p(j), \forall k \ i < k < j$

- Za operator $M3^1$: $p(k) \neq p(i) \neq p(j), \forall k \ i < k < j$.

Tako izabranu parcijalnu permutaciju $p(i-j)$ moguće je mutirati pomoću operatora mutacije M1-M3.

Ujedno, uveden je novi operator mutacije M4, koji mijenja položaj (x, y) pojedinog objekta. Primjena operatora mutacije M4 uključuje slučajan izbor objekta u planu te slučajnu promjenu njegova položaja:

$$(x, y)^{t+1} = (x^t + \Delta x, y^t + \Delta y), \quad (4.8)$$

u novi položaj. Promjena položaja Δx i Δy radi se prema normalnoj razdiobi s parametrima $\mu_1 = x^K$ i $\mu_2 = y^K$, $\sigma_{1,2} = 10$. Specifičnost problema koji se razmatra i rješava uključuje mogućnost da se pojedini objekt unutar permutacije pojavi više puta, dok u slučaju kada se pojedini objekt unutar permutacije pojavi samo jedanput to označava da se odmah postavlja na konačno mjesto iz \mathcal{K} .

Ujedno rješenja koja generira genetski algoritam mogu uključivati permutacije tipa (1 5 7 3 3 4 5 3 6 2). Sve se permutacije nakon mutacije evaluiraju pomoću algoritma (Algoritam 4-3.) kako bi se utvrdilo da generirani plan nema kolizije između objekata. Nakon evaluacije moguće je filtrirati navedenu permutaciju i ukloniti višak objekata koji se ponavljaju: (1 5 7 3 3 4 5 3 6 2) \rightarrow (1 5 7 3 4 5 3 6 2). Cjelokupni razvijeni genetski algoritam prikazan je u algoritmu (Algoritam 4-2.).

4. Planiranje robotskog djelovanja

Algoritam 4-2. *Optimizacijski genetski algoritam.*

Algoritam : Optimizacijski_GA

```
Ulaz:  $\mathcal{O}, p_c, p_m, it_{\max}, h$   
Izlaz:  $\mathcal{O}_{\text{opt}}, \mathbf{O}_{\text{opt}}$   
1  $\mathcal{O}_{\text{opt}} \leftarrow \mathcal{O}$  // Inicijaliziraj  $\mathcal{O}_{\text{opt}}$ .  
2 Dok  $g < it_{\max}$  ponavljaj // ponavljaj postupak  $it_{\max}$  iteracija  
3   za svaki  $\mathbf{O}_i \in \mathcal{O}_{\text{opt}}, b++$   
4      $s \leftarrow \text{randi}$  // Generiraj slučajni broj.  
5     Ako  $s < p_m$  onda  
6       mutiraj  $\mathbf{O}_i$  s  $M1^1$  i  $M2^1 \rightarrow \mathbf{O}_i(s)$ , prema izrazu (4.6)  
7       mutiraj  $\mathbf{O}_i(s)$  s  $M3^1 \rightarrow \mathbf{O}_i(s.i)$ , prema izrazu (4.7)  
8       mutiraj  $\mathbf{O}_i(s.i)$  sa  $M4(\Delta x$  i  $\Delta y) \rightarrow \mathbf{O}_i(s.i.m)$ , prema izrazu (4.8)  
9     kraj  
10     $[\Phi^{t+1}, \Phi^t] \leftarrow \text{verificiraj\_plan\_djelovanja}(\mathbf{O}_i(s.i.m), \mathbf{O}_i)$   
11    ako  $\Phi^{t+1} < \Phi^t$  onda  
12       $\mathcal{O}_{\text{opt}}(i)^{g+1} \leftarrow \mathbf{O}_i(s.i.m)$   
13    inače  
14       $\mathcal{O}_{\text{opt}}(i)^{g+1} \leftarrow \mathcal{O}_{\text{opt}}(i)^g$   
15    kraj  
16     $s \leftarrow \text{randi}$  // Generiraj slučajni broj.  
17    ako  $s < p_c$  onda  
18      slučajnom selekcijom izaberi dva dobra kromosoma iz turnira  $\rightarrow \mathbf{O}_1$  i  $\mathbf{O}_2$   
19       $[\mathbf{O}_3, \mathbf{O}_4] \leftarrow \text{križaj\_u\_jednoj\_točki}(\mathbf{O}_1$  i  $\mathbf{O}_2)$   
20      verificiraj  $\mathbf{O}_3$  i  $\mathbf{O}_4$ , po potrebi ponovi križanje  $\rightarrow$  skok na red 14  
21      ubaci bolje od dva rješenja u  $\mathcal{O}_{\text{opt}}(kraj+1:) \leftarrow \max([\mathbf{O}_3, \mathbf{O}_4])$   
22      napravi turnirsku selekciju u  $\mathcal{O}_{\text{opt}}$  i izbaci jedno loše rješenje  
23    kraj  
24     $p_m \leftarrow p_m \times h; g \leftarrow g + 1;$   
25  kraj  
26  $\mathbf{O}_{\text{opt}} \leftarrow \text{min\_dobrota}(\mathcal{O}_{\text{opt}})$ 
```

Cjelokupni skup svih optimiranih rješenja označen je s \mathcal{O}_{opt} , dok je najbolje rješenje iz skupa označeno kao \mathbf{O}_{opt} . Algoritam se izvršava dok se ne izvrši it_{\max} iteracija. Varijabla h poprima vrijednost iz intervala $h \in [0,9...1]$ sa svrhom smanjivanja vjerojatnosti mutacije u svakoj iteraciji, kao što je označeno u redu 24 algoritma (Algoritam 4-2.).

4. Planiranje robotskog djelovanja

Algoritam 4-3. Verifikacija plana djelovanja.

Algoritam: Verifikacija_plana

```
Ulaz:  $\mathbf{P}, \mathbf{O}_i$   
Izlaz: status,  $\Phi$   
1 status  $\leftarrow 0, \Phi \leftarrow 0$ ;  
2 // Evaluiraj izvodljivost plana uslijed mogućih kolizija  $\rightarrow$  status  
3 // Izračunaj ukupnu duljinu trajanja izvođenja plana  $\rightarrow \Phi$   
4 za svaki red  $j$  iz plana djelovanja  $\mathbf{O}_i(j, :)$   
5     objekt  $\leftarrow \mathbf{O}_i(j, 2)$   
6     stari_položaj  $\leftarrow \mathbf{P}(\text{objekt}, 3:5)$   
7     novi_položaj  $\leftarrow \mathbf{O}_i(j, 3:5)$   
8      $\mathbf{M} \leftarrow$  generiraj_strukturu( $\mathbf{P}(\text{objekt}, \text{stari\_položaj})$ )  
9      $\mathbf{N} \leftarrow$  generiraj_strukturu( $\mathbf{P}(\text{objekt}, \text{novi\_položaj})$ )  
10     $\mathbf{P} \leftarrow \mathbf{P} - \mathbf{M} + \mathbf{N}$   
11    status  $\leftarrow$  evaluacija_kolizije( $\mathbf{P}, \mathbf{M}, \mathbf{N}$ )  
12    ako status = 1, prekid programa // Kolizija  
13 kraj  
14  $\Phi \leftarrow$  izračunaj vrijeme plana prema izrazu (3.4)
```

Početno lokalizirano stanje objekata kao što je prikazano na slici (Slika 4-2.a) označeno je s \mathbf{P} . Plan \mathbf{O}_i dobiven pomoću algoritma (Algoritam 4-2.) verificira se uslijed mogućih kolizija provjerom svakog koraka u planu te uzimajući u obzir početno stanje objekata \mathbf{P} .

4.3.2.2. Operator križanja

Križanje rješenja obavlja se prema standardnom operatoru križanja u jednoj točki [103], prema kojem se slučajnim odabirom izabiru dvije jedinke u populaciji \mathbf{p}_1 i \mathbf{p}_2 te mjesto križanja k . Od svake jedinke izabire se permutacija redoslijeda. Kao primjer moguće je uzeti sljedeće dvije permutacije:

$$\mathbf{p}_1 = (1\ 2\ 3\ 6\ 5\ 4\ 2\ 8),$$

$$\mathbf{p}_2 = (3\ 4\ 2\ 5\ 6\ 2\ 8\ 4).$$

Mjesto križanja k treba biti takvo da su elementi nakon točke križanja obaju roditelja identični, no u različitoj permutaciji prema algoritmu (Algoritam 4-4.). Slučajnim odabirom izabire se mjesto križanja $k = 4$. Formiraju se dva nova pojedinca p_1^c i p_2^c :

$$\mathbf{p}_{12}^c = (1\ 2\ 3\ 5\ 6\ 2\ 8\ 4),$$

$$\mathbf{p}_{21}^c = (3\ 4\ 2\ 6\ 5\ 4\ 2\ 8).$$

4. Planiranje robotskog djelovanja

Nadalje, moguće je i križanje planova različitih dužina kromosoma (permutacija). Kao primjer moguće je uzeti permutaciju p_1 i permutaciju

$$p_3 = (1\ 2\ 1\ 5\ 4\ 3\ 6\ 5\ 4\ 8\ 2).$$

Mjesto križanja k ponovno treba biti takvo da su elementi nakon točke križanja obaju roditelja identični, no u različitoj permutaciji, kao što je prikazano u prethodnom primjeru. U ovom se slučaju mjesto križanja izabire od zadnjeg elementa pojedinog kromosoma. Slučajnim odabirom izabire se mjesto križanja $k = 5$. Formiraju se dva nova pojedinca p_1^c i p_2^c :

$$p_{13}^c = (1\ 2\ 3\ 6\ 5\ 4\ 8\ 2),$$

$$p_{31}^c = (1\ 2\ 1\ 5\ 4\ 3\ 5\ 6\ 2\ 8\ 4).$$

Algoritam 4-4. *Usporedi permutacije.*

Algoritam: Usporedi_permutacije

```
Ulaz:  $p_1, p_2$ 
Izlaz: status,
1  status ← 0
2  // Da li su dvije permutacije  $p_1$  i  $p_2$  identične → status
3   $p_1$  ← sortiraj_permutaciju( $p_1$ )
4   $p_2$  ← sortiraj_permutaciju( $p_2$ )
5   $akop_1 = p_2$ 
6  |   status ← 1           // Permutacije nisu identične
7  kraj
```

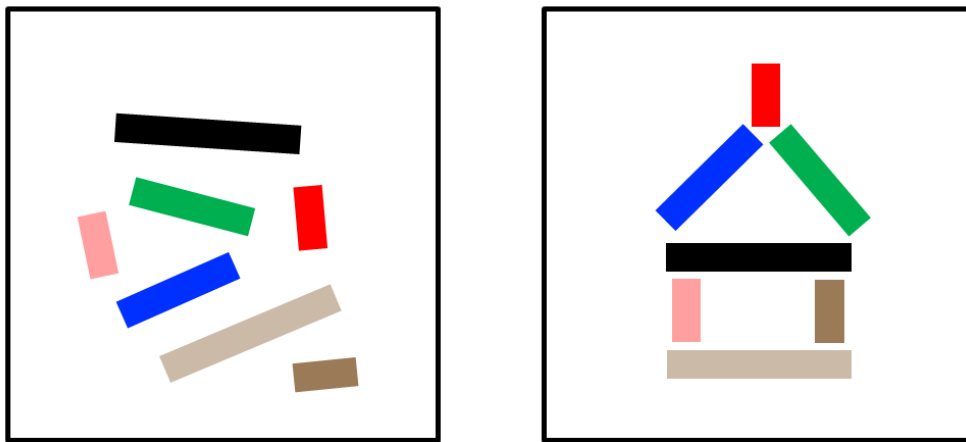
Oba se rješenja evaluiraju prema funkciji (Algoritam 4-3.). Rješenja koja daju izvediv plan ulaze u novu populaciju, iz koje će se napraviti turnirska selekcija s elitizmom. U svakoj se generaciji uvijek zadržava najbolja jedinka, dok se od ostalih jedinki radi turnirska selekcija.

Križanje dvaju rješenja uzima u obzir isključivo permutacije, dok se vektor položaja i rotacija objekata preuzima iz originalne permutacije. Usporedno križanje permutacija i vektora položaja (x, y), tj. uzimanje u obzir položaja iz jednog rješenja te ubacivanjem tog položaja u drugo rješenje, nije provedeno. Usporednim križanjem obiju komponenata dobiva se velik broj neispravnih rješenja tj. planova koji završavaju kolizijama objekata te nisu izvedivi prema algoritmu (Algoritam 4-3.). Zbog toga je upotrijebljeno križanje rješenja koje uzima u obzir permutaciju redoslijeda dok je se promjena položaja radi isključivo pomoću operatora mutacije M4.

4.3.2.3. Implementacija genetskog algoritma planiranja

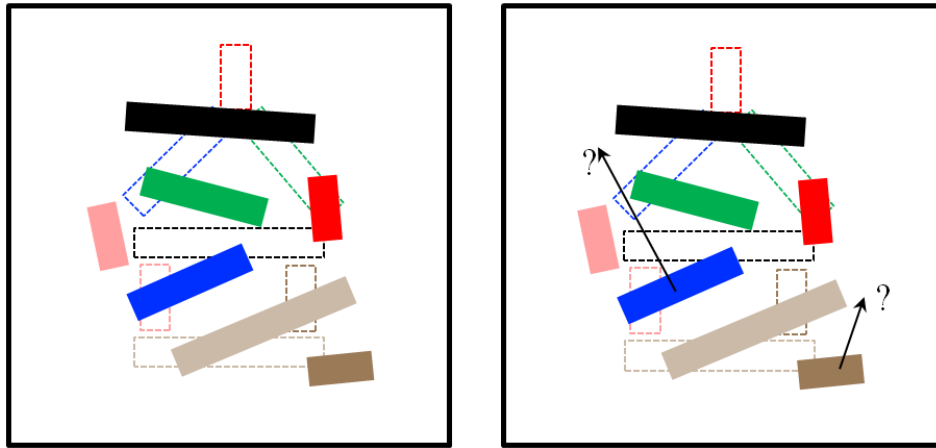
Osnovni cilj genetskog algoritma je optimiranje rješenja u vidu smanjenja ukupnog puta koji treba prijeći robot kako bi objekte iz \mathcal{P} doveo u željeno stanje \mathcal{K} . Prema tablici (Tablica 4-2.) moguće je prijedeni put i zadanu brzinu kretanja korelirati s utrošenim vremenom za izvođenje pojedinog plana koji u tom slučaju predstavlja vrijednost koja se optimira. Idealno rješenje predstavlja slučaj kada je svaki od objekata moguće izravno postaviti na konačno mjesto, kao što je prikazano na slici (Slika 4-9.). Takav je problem moguće riješiti standardnim algoritmima koji se bave problemom trgovačkog putnika (engl. *travelling salesman problem*) [100,104], te se taj problem neće razmatrati u okviru predloženog istraživanja.

Na slici (Slika 4-7.) prikazan je jedan primjer rasporeda objekata koji odgovara stvarnoj prostornoj strukturi prikazanoj na slici (Slika 4-1.). Na lijevoj slici prikazano je početno stanje objekata tj. njihov položaj i orijentacija. Na desnoj se slici nalazi prostorna struktura čiju izgradnju treba planirati generiranjem uređenog skupa elemenata rada robota.



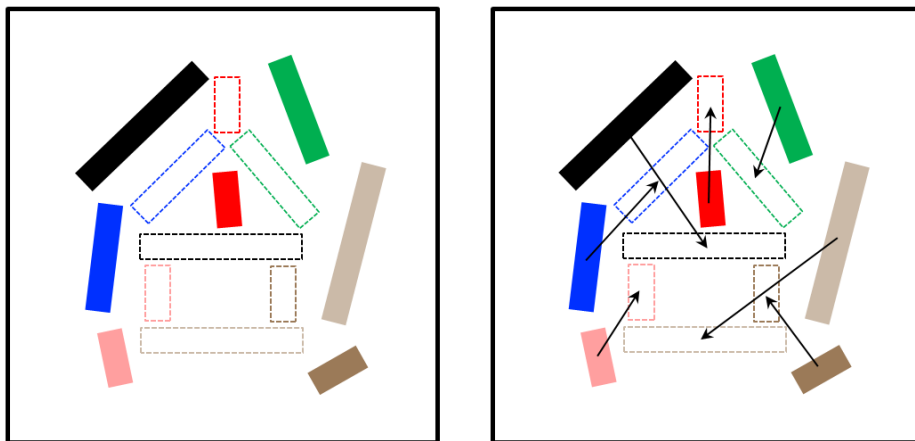
Slika 4-7. Lijevo: početno stanje objekata \mathcal{P} , desno: konačno stanje objekata \mathcal{K} .

Svaki od objekta označen je različitom bojom kako bi se ilustriralo koji se od objekata preslikava na koji dio radnog prostora. Na slici (Slika 4-8.) prikazan je problem planiranja koji uključuje nemogućnost postavljanja ijednog od objekata iz početnog stanja na izravan položaj iz konačnog stanja.



Slika 4-8. Prikaz početnog i konačnog stanja u radnom prostoru s vidljivim preklapanjem svih objekata.

Na slikama (Slika 4-9.) i (Slika 4-10.) prikazana su preostala dva moguća slučaja problema planiranja. Na slici (Slika 4-9.) ilustriran je problem u sklopu kojega nije potrebno napraviti planiranje jer je sve objekte moguće izravno postaviti u konačni položaj. Svaki redoslijed odnosno permutacija daju valjano rješenje te je broj koraka jednak broju objekata na sceni tj. u radnom prostoru robota. Taj je problem moguće riješiti standardnim algoritmima koji se bave problemom trgovačkog putnika [104]. Taj se slučaj neće proučavati u doktorskom radu.

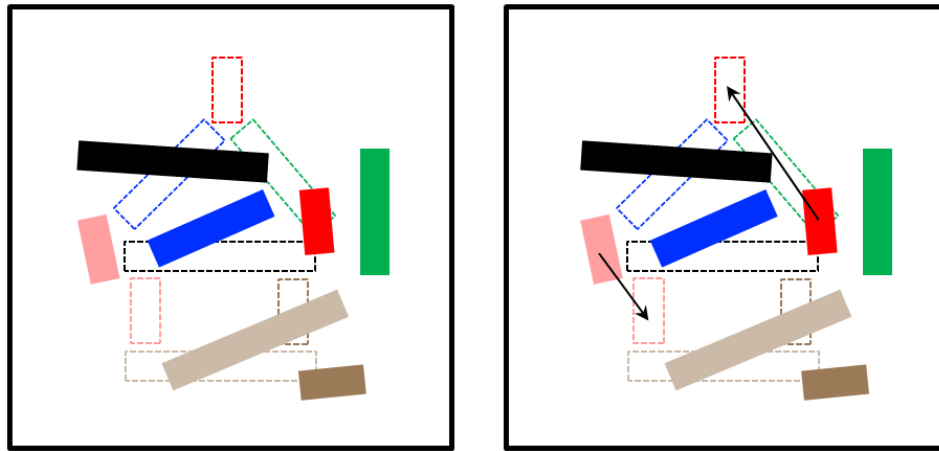


Slika 4-9. Jednostavan slučaj planiranja – izravno postavljanje svih objekata na konačni položaj.

Na slici (Slika 4-10.) prikazan je slučaj kada se objekti koji se nalaze u \mathcal{P} trebaju dovesti u stanje \mathcal{K} koje djelomično interferira s \mathcal{P} . To znači da ili nije moguće izravno postaviti sve objekte na konačno mjesto, ili nije moguće postaviti neke objekte na konačno mjesto.

Na slici (Slika 4-10.) prikazan je slučaj gdje je određene objekte moguće izravno postaviti na konačno mjesto. Nakon postavljanja određenog broja objekata na konačna mjesta dolazi se do

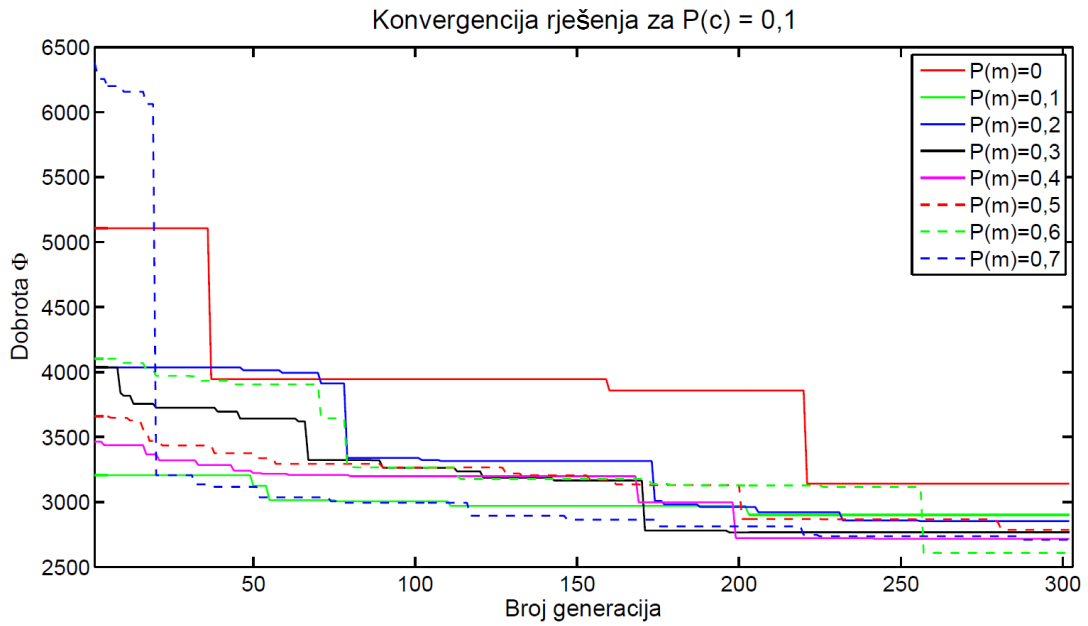
situacije gdje više nije moguće nijedan od preostalih objekata izravno postaviti na konačno mjesto.



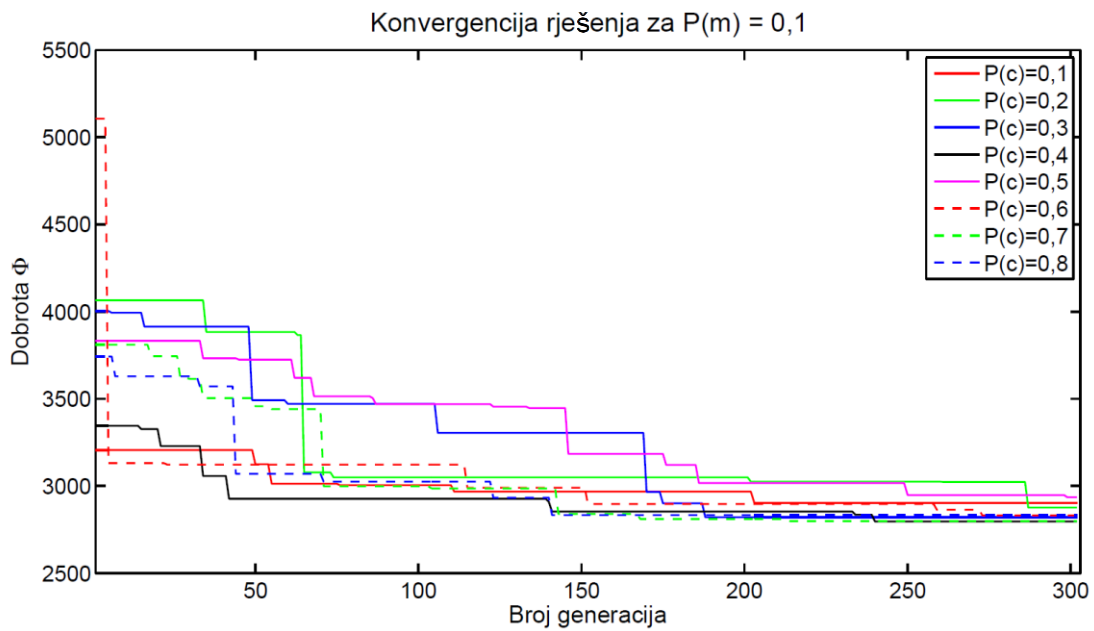
Slika 4-10. Slučaj kada nije moguće izravno postaviti sve objekte u konačni položaj zbog kolizija.

U okviru razvoja genetskog algoritma uzet će se u obzir problemi planiranja gdje postoji određen broj objekata koje nije moguće izravno postaviti na konačno mjesto, zbog čega je potrebno uz redosljed (permutaciju) optimirati i položaj objekata. Nadalje, u ovakvom općenitom slučaju broj potrebnih koraka veći je od ukupnog broja objekata u radnom prostoru, zbog čega se dobiva rješenje u obliku permutacije s ponavljanjem članova.

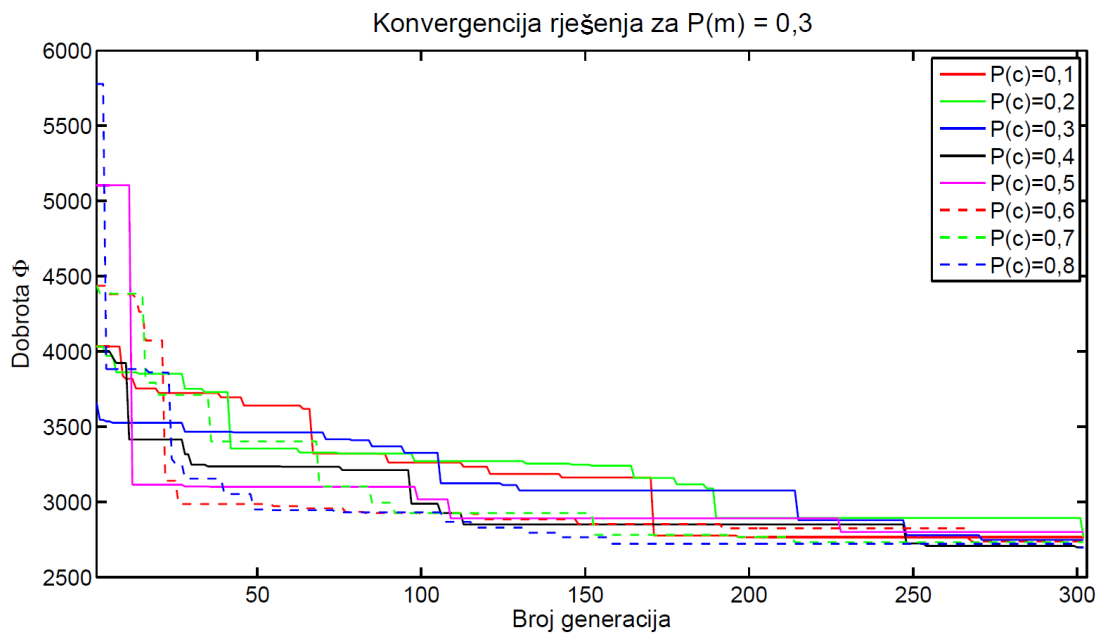
U svrhu evaluacije razvijenog genetskog algoritma izvršit će se analiza konvergencije promjenom parametara vjerojatnosti križanja p_c i vjerojatnosti mutacije p_m . Odredit će se eksperiment s vjerojatnostima križanja i mutacije u intervalima $p_c \in [0, \dots, 1]$, $p_m \in [0, \dots, 1]$ i korakom $k = 0,1$. Ukupno će se ispitati slučaj konvergencije za 121 kombinaciju vjerojatnosti križanja i mutacije. Broj iteracija algoritma postavljen je na 300. Prikazani su dijagrami s konvergencijom rješenja za $p_c = 0,1$ (Slika 4-12.), $p_m = 0,1$ (Slika 4-13.) i $p_m = 0,3$ (Slika 4-14.). Za svaku od 121 kombinacije p_c i p_m napravljeno je 10 simulacija. Iz dijagrama je vidljivo da algoritam najbrže konvergira s višim vrijednostima p_m , no u tim slučajevima ne postiže najbolja rješenja. Nadalje, trenutno nije moguće usporediti rješenje algoritma s globalno optimalnim rješenjem kao što je to izvršeno u [105] jer globalno optimalno rješenje za razmatranu domenu problema u nije poznato. Radi kasnije verifikacije i validacije, rješenja dobivena genetskim algoritmom usporedit će se s modificiranim algoritmom pohlepnog pretraživanja.



Slika 4-11. Konvergencija rješenja za vjerojatnost križanja $P(c) = 0,1$ te različite vrijednosti vjerojatnosti mutacija $P(m)$.

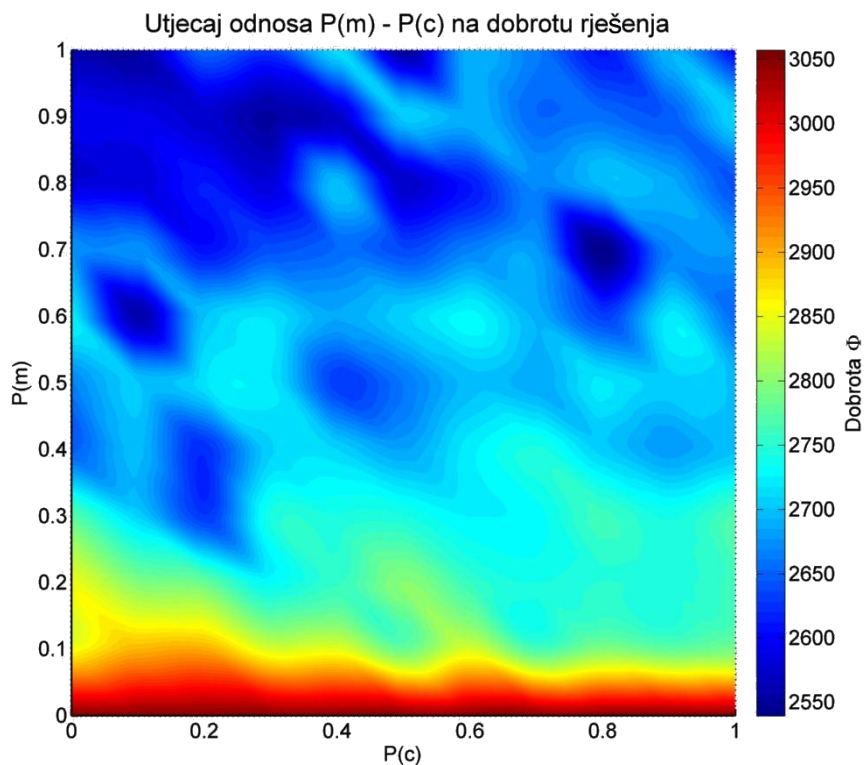


Slika 4-12. Konvergencija rješenja za vjerojatnost mutacije $P(m) = 0,1$ te različite vrijednosti vjerojatnosti križanja $P(c)$.

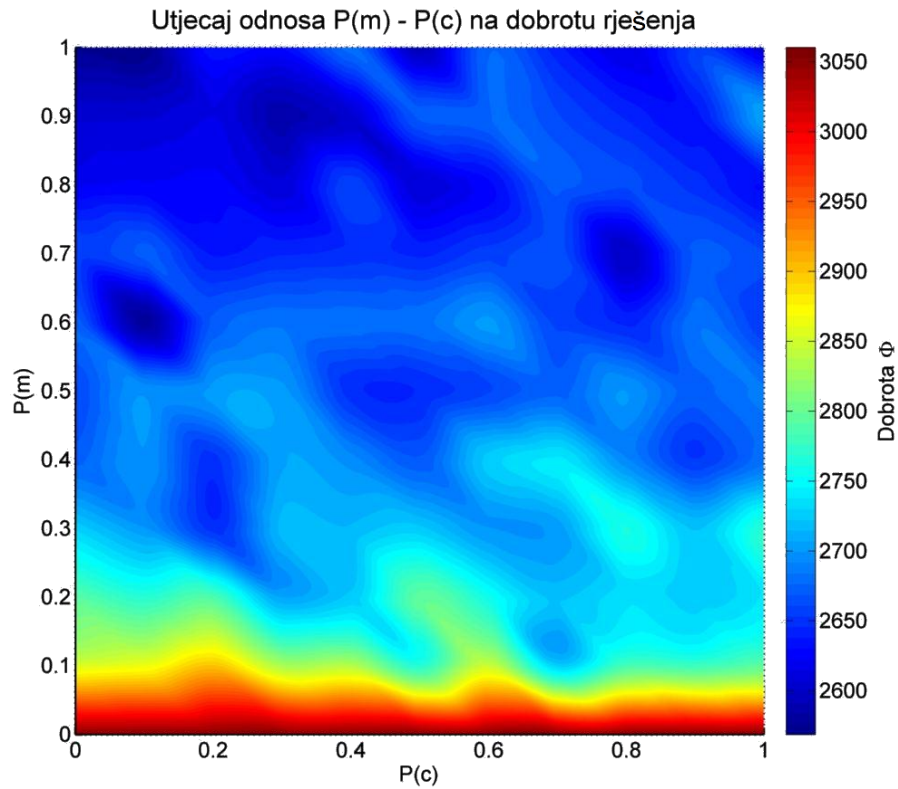


Slika 4-13. Konvergencija rješenja za vjerojatnost mutacije $p_m = 0,1$ te različite vrijednosti vjerojatnosti križanja p_c .

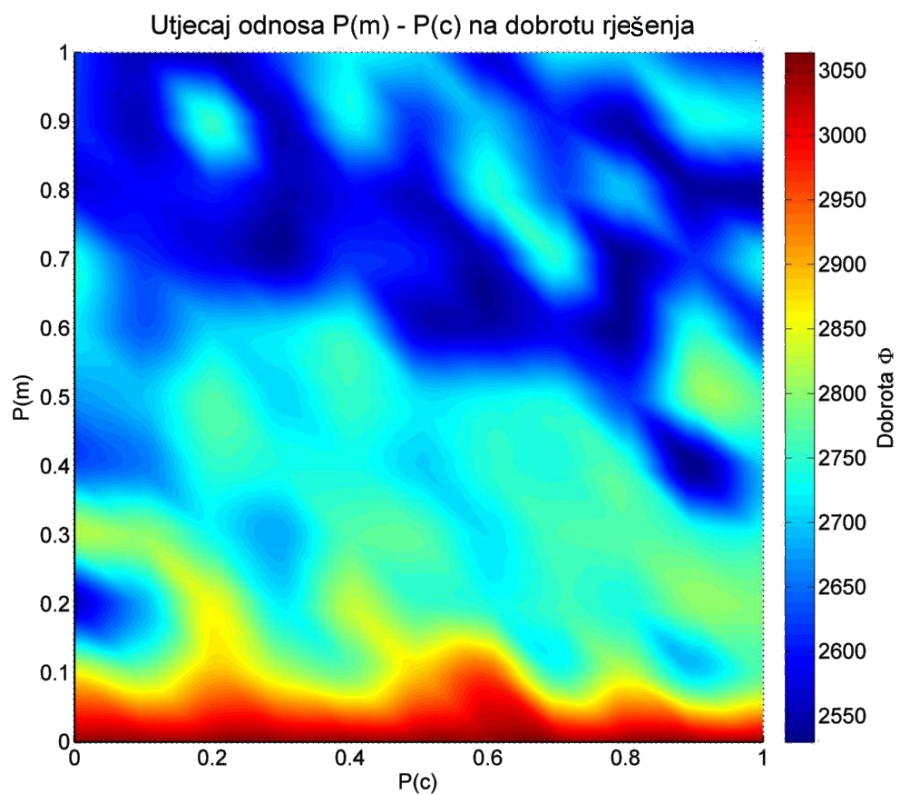
Kako bi se detaljnije analizirao utjecaj pojedine vrijednosti p_c i p_m , prikazana su sva dobivena rješenja u dvodimenzionalnom prikazu. Prikazan je medijan i srednja vrijednost vrijednosti funkcije dobrote.



Slika 4-14. Utjecaj odnosa p_m i p_c na dobrotu rješenja Φ . Raspon $p_m = [0 - 1]$, $p_c = [0 - 1]$. Na grafikonu je prikazan medijan od 10 simulacija za svaki par: ukupno 1210 simulacija.



Slika 4-15. Utjecaj odnosa p_m i p_c na dobrotu rješenja Φ . Raspon $p_m = [0 - 1]$, $p_c = [0 - 1]$. Na grafikonu je prikazana srednja vrijednost od 10 simulacija za svaki par: ukupno 1210 simulacija.

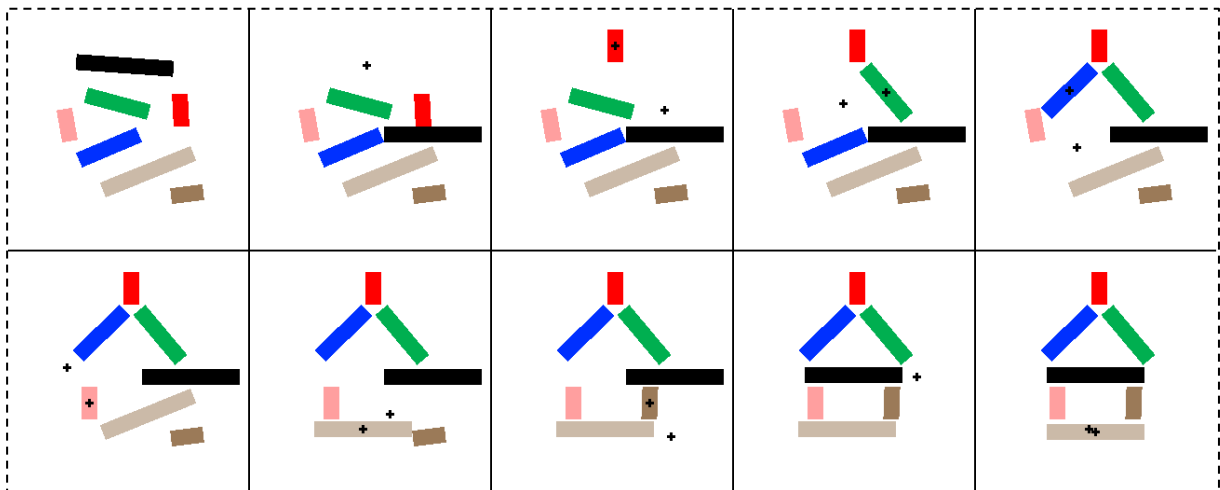


Slika 4-16. Utjecaj odnosa p_m i p_c na dobrotu rješenja Φ . Raspon $p_m = [0 - 1]$, $p_c = [0 - 1]$. Dijagram prikazuje jedno od 8 rješenja korištenih u generiranju dijagrama na slici (Slika 4-14.) i (Slika 4-15.).

4. Planiranje robotskog djelovanja

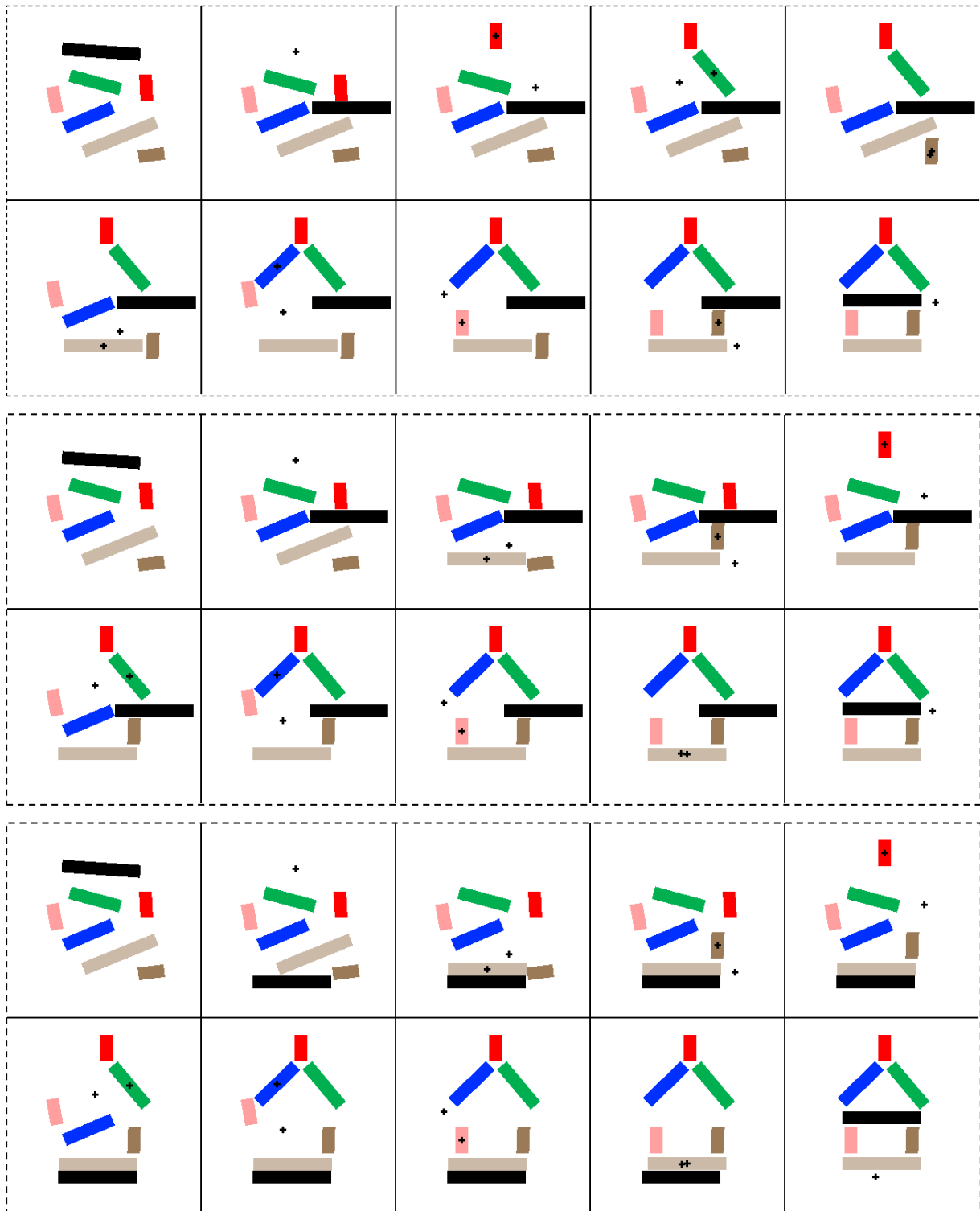
Prema dijagramima (Slika 4-14.), (Slika 4-15.), (Slika 4-16.) vidljivo je kako vrijednosti p_m u intervalu $[0... \sim 0,3]$ daju vrlo ujednačena rješenja, no imaju relativno visoku vrijednost funkcije dobrote. Nadalje, najlošiji su rezultati dobiveni postavljanjem $p_m = 0$. Zbog specifičnosti problema koji se razmatra, a koji uključuje mogućnosti mutacije položaja pojedinog objekta, operator mutacije je izrazito bitan za precizno namještanje svih položaja objekata te kako bi se dobila rješenja veće dobrote. Rješenja najveće dobrote dobivena su za veći broj kombinacija p_c i p_m , što je vidljivo iz dijagrama (Slika 4-14.).

Kako bi se omogućila usporedba najboljeg generiranog rješenja s ostalim rješenjima, uz rješenje najveće dobrote (Slika 4-17.) prikazana su tri rješenja dobivena ostalim kombinacijama parametra p_c i p_m (Slika 4-18.).



Slika 4-17. Najbolje rješenje za problem prikazan na slici (Slika 4-8.). Dobrota rješenja iznosi 2497.

Na slici (Slika 4-18.) prikazana su tri rješenja koja su različita od najboljeg rješenja prikazanog na slici (Slika 4-8.). Dobrota svih navedenih rješenja odstupa od optimalnog za 2,5 %. Iz ovih je primjera vidljivo kako je za isti problem moguće generirati veći broj različitih rješenja koja imaju izrazito malo rasipanje dobrote.



Slika 4-18. Tri rješenja za problem prikazan na slici (Slika 4-8.). Dobrota prvog rješenja: 2544, dobrota drugog rješenja: 2562, Dobrota trećeg rješenja: 2539.

4.3.3. Generiranje plana pohlepnim pretraživanjem

Kako bi se razvijeni genetski algoritam verificirao, usporedit će se i analizirati rezultati dobiveni algoritmom pohlepnog pretraživanja. Radi navedene verifikacije implementiran je

algoritam pohlepnog pretraživanja za generiranje planova robotskog djelovanja. U određenim je slučajevima kod problema planiranja moguće pronaći vrlo dobra rješenja koristeći se relativno jednostavnom heuristikom pohlepnog pretraživanja (engl. *greedy search*). Postavljena su sljedeća pravila na temelju kojih algoritam djeluje:

Korak 1.

Pomakni trenutno najbliži objekt O na konačno mjesto ako je slobodno. Ako nijedno mjesto nije izravno slobodno, idi na Korak 2, inače ponovi Korak 1.

Korak 2.

Izaberi najbliži objekt O iz \mathbf{P}^t ili objekt na čijem se mjestu u \mathbf{K} nalazi najveći broj objekata. Pomakni O na najbliže slobodno mjesto u \mathbf{K} .

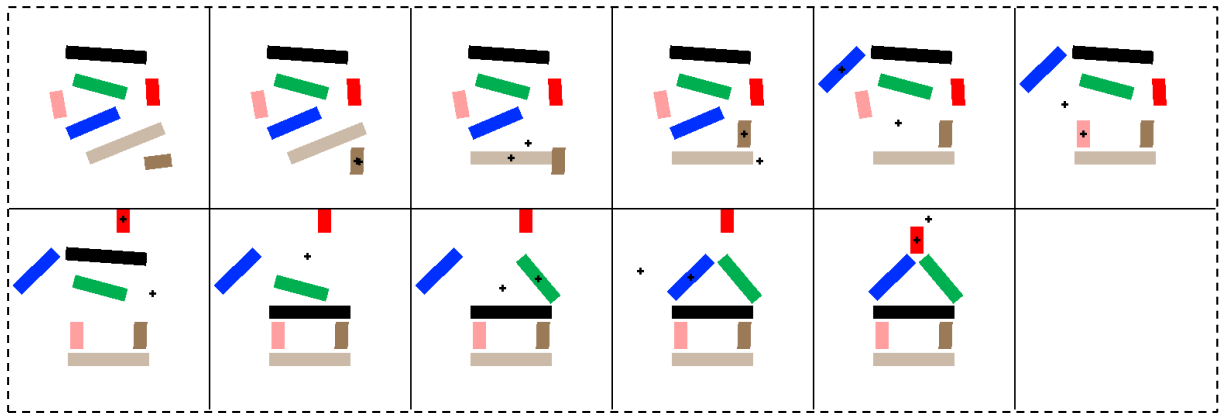
Korak 3.

Ako je $\mathbf{P}^t \neq \mathbf{K}$, idi na Korak 1, inače KRAJ.

Koristeći se tim dvama jednostavnim pravilima pohlepnog pretraživanja, robot će pomicati objekte koji „smetaju“ sve dok na njihovo mjesto neće moći postaviti neki drugi objekt, a kao inicijalni korak uvijek će probati postaviti najbliži objekt na konačno mjesto. Takav algoritam daje dobre rezultate kada je većina konačnih mjesta slobodna. No tada bi bilo moguće upotrijebiti već postojeće algoritme koji se primjerice primjenjuju za rješavanje problema trgovačkog putnika [104]. Problem planiranja koji se rješava u ovom radu karakteriziran je time što pozicije objekata nemaju nepromjenjiv položaj, već se u svakom međukoraku pojedini objekt može postaviti na bilo koje slobodno mjesto u radnom prostoru. Jedini položaji koji su nepromjenjivi su oni iz početnog i konačnog stanja. Problem se može svesti na problem trgovačkog putnika samo kada je sve objekte moguće izravno postaviti na konačna mjesta, kao što je ilustrirano u prethodnom poglavlju. Problem koji se razmatra je upravo suprotan, a određen je dinamikom promjena koje se događaju kada se pojedini objekt pomakne te dolazi do transformacije $\mathbf{P}^t \rightarrow \mathbf{P}^{t+1}$. Tada svaki izbor lokacije na koji se objekt postavlja u potpunosti mijenja sve udaljenosti između lokacija predmeta u \mathbf{P}^{t+1} , zbog čega nije moguće primijeniti standardne algoritme.

Ovisno o početnom položaju iz kojeg se određuje najbliži objekt, generirana su različita rješenja pomoću pohlepnog pretraživanja.

4. Planiranje robotskog djelovanja

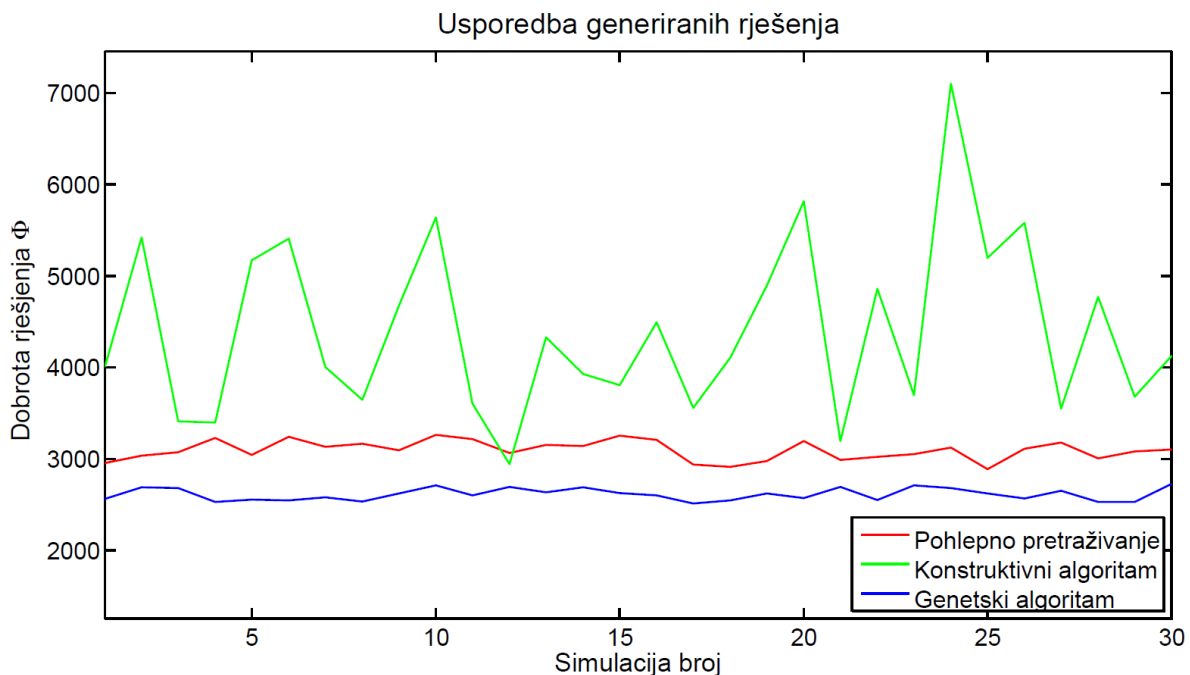


Slika 4-19. Prikaz najboljeg rješenja dobivenog pohlepnim pretraživanjem, $\Phi = 2889$.

Tablica 4-3. Najbolje rješenje (10 koraka) dobiveno pohlepnim pretraživanjem, $\{x_1, y_1, \delta_1\}$: pozicija izuzimanja, $\{x_2, y_2, \delta_2\}$: pozicija odlaganja, $\Phi_K = 2889$.

| Korak | Objekt | x1 | y1 | δ_1 | x2 | y2 | δ_2 |
|-------|--------|-----|-----|------------|-----|-----|------------|
| 1 | 4 | 234 | 228 | 7 | 233 | 225 | -92 |
| 2 | 2 | 210 | 186 | 22 | 229 | 164 | 0 |
| 3 | 4 | 233 | 225 | 7 | 198 | 205 | -92 |
| 4 | 6 | 184 | 144 | 23 | 115 | 71 | 44 |
| 5 | 3 | 160 | 99 | -80 | 198 | 123 | 91 |
| 6 | 5 | 144 | 221 | 93 | 48 | 183 | -89 |
| 7 | 1 | 96 | 161 | -4 | 168 | 164 | 0 |
| 8 | 7 | 137 | 153 | -15 | 125 | 199 | -50 |
| 9 | 6 | 115 | 71 | 23 | 123 | 136 | 44 |
| 10 | 10 | 48 | 183 | 93 | 75 | 168 | -89 |

Napravljeno je ukupno 30 simulacija za koje je generirano rješenje pomoću konstruktivnog algoritma (Algoritam 4-1.). Nadalje, pomoću genetskog algoritma optimirano je inicijalno rješenje te je generiran neovisan skup rješenja pohlepnim pretraživanjem. Na dijagramu (Slika 4-20.) prikazana su najbolja rješenja dobivena pomoću svake od triju metoda.



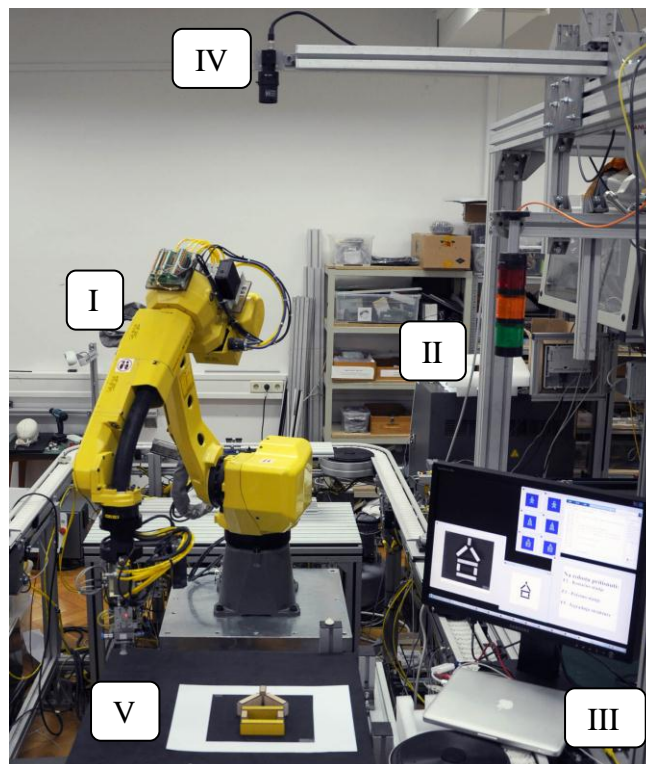
Slika 4-20. Usporedba najboljih rješenja dobivenih pohlepnim pretraživanjem s konstruktivnim i genetskim algoritmom.

Vidljivo je da su rješenja dobivena konstruktivnim algoritmom neujednačena, što je posljedica pravila konstruktivnog algoritma opisanog u poglavlju 4.3.1. Osnovni princip generiranja rješenja koristeći se konstruktivnim algoritmom podrazumijeva slučajan izbor objekata neovisno o trenutnom stanju radnog prostora sa svrhom pretrage čim većeg broja rješenja. Sljedeći korak uzima u obzir da se dobivena rješenja optimiraju pomoću genetskog algoritma opisanog u poglavlju 4.3.2. U tom se postupku međusobnim križanjem pojedinca te mutacijom redosljeda objekata u planu te njihovog položaja u određenom broju iteracija kreiraju rješenja veće dobrote. Iz dijagrama (Slika 4-20.) vidljivo je da su rješenja dobivena genetskim algoritmom prosječno 18% veće dobrote od onih dobivenih pohlepnim pretraživanjem. Rješenja dobivena pohlepnim pretraživanjem također imaju ujednačenu dobrotu. Uslijed ograničenja u pretraživanju prostora, koja su određena pravilima izbora objekata u planu, te nemogućnosti naknadne optimizacije redosljeda i položaja objekata dobivena rješenja lošija su od optimiranih rješenja dobivenih genetskim algoritmom. Nadalje, zbog svoje stohastičke prirode, korištenjem konstruktivnog algoritma u 12. iteraciji dobiveno je rješenje koje ima veću dobrotu od onog dobivenog pohlepnim pretraživanjem.

5. VALIDACIJA UPRAVLJAČKOG MODELA

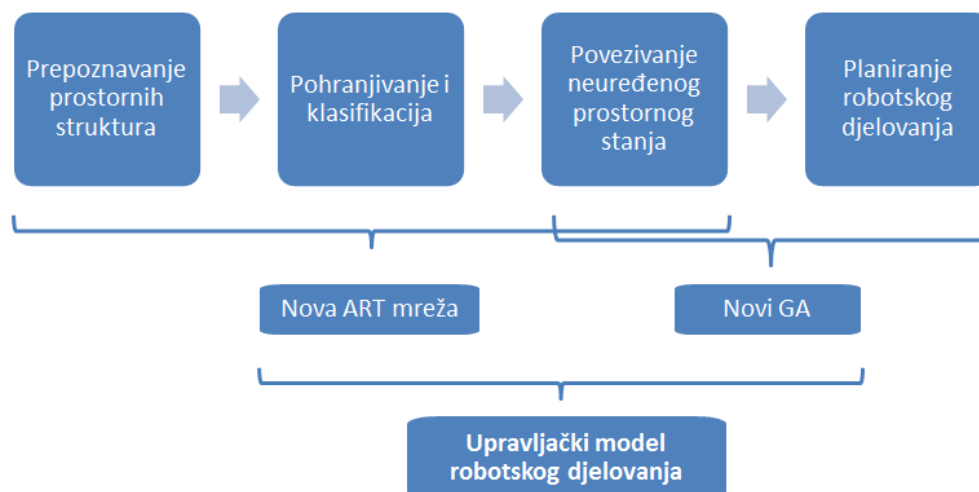
Kako bi se cjelokupan upravljači model, čije su osnovne cjeline ART neuronska mreža te genetski algoritam planiranja robotskog djelovanja, integrirao i verificirao na realnoj laboratorijskoj opremi, potrebno je kreirati robotske upravljačke programe koji će se koristiti navedenim cjelinama kao osnovnim upravljačkim modulima.

Eksperimentalni dio istraživanja proveden je na industrijskoj robotskoj ruci proizvođača Fanuc tipa M10-iA sa šest rotacijskih (6R) stupnjeva slobode gibanja. Upotrijebljen je industrijski vizijski sustav te upravljačko računalo na kojem je razvijen upravljački program s dvosmjernom komunikacijom preko TCP/IP protokola s robotom. Cijeli postav testiran je na opremi u Laboratoriju za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje pri Sveučilištu u Zagrebu (Slika 5-1.).



Slika 5-1. Prikaz laboratorijskog postava. I) robot, II) robotska upravljačka jedinica, III) računalo, IV) vizijski sustav (kamera), V) radni prostor robota

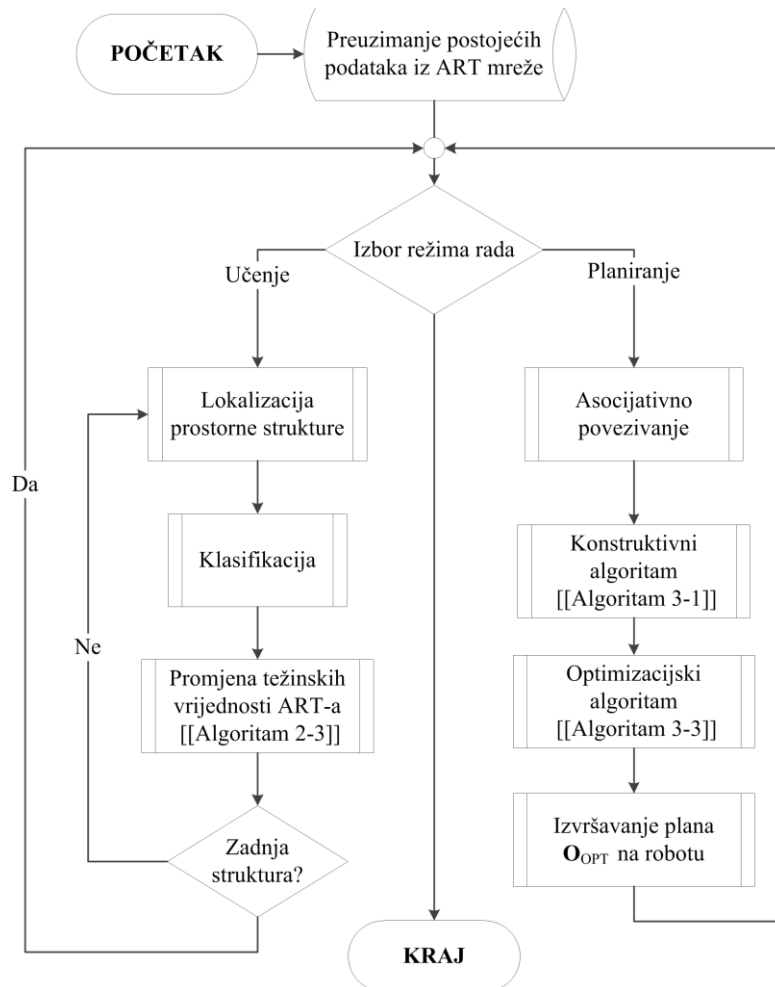
Na dijagramu (Slika 5-2.) prikazane su osnove cjeline upravljačkog modela koje su opisane u prethodnim poglavljima.



Slika 5-2. Osnovne cjeline upravljačkog modela.

Prvi korak u cijelom procesu eksperimentalne validacije sastoji se od prepoznavanja prostornih struktura, njihove klasifikacije prema obliku i rasporedu te povezivanja neuređenog prostornog stanja s poznatim klasificiranim prostornim strukturama. Detaljan opis razvijene ART mreže i funkcije preslikavanja opisan je u poglavlju 2.3. i 3. Funkcija preslikavanja kojoj je cilj povezivanje prepoznatih objekata u neuređenom stanju s prethodno klasificiranim strukturama povezuje ART mrežu s genetskim algoritmom planiranja robotskog djelovanja. Jedan od ciljeva upravljačkog modela (Slika 5-2.) je omogućiti izgradnju prostornih struktura iz neuređenog stanja objekata na temelju prethodnih spoznaja iz procesa klasifikacije uređenih prostornih struktura. Ulazni parametri algoritma planiranja su početno neuređeno stanje objekata \mathbf{P} te najbližnja prostorna struktura \mathbf{K} , koja se generira na temelju funkcije preslikavanja. Planiranje djelovanja kao osnovni cilj ima generiranje i određivanje diskretnog konačnog skupa aktivnosti \mathcal{A} , takvog da vrijedi $\mathcal{A} : \mathbf{P} \rightarrow \mathbf{K}$. Određivanje skupa aktivnosti provedeno je u dva koraka. Kao prvi korak generiran je inicijalni skup rješenja konstruktivnim algoritmom s usporednim kreiranjem i vrednovanjem rješenja. U drugom su koraku najbolja inicijalna rješenja optimirana novim genetskim algoritmom korištenjem modificiranih operatora mutacije i križanja.

Sa svrhom implementacije prethodno opisanih cjelina na realnoj istraživačkoj opremi potrebno je razviti dvije osnovne funkcionalne cjeline te pripadajuće robotske upravljačke programe: I. učenje i klasifikacija, II. prepoznavanje prostornih struktura i planiranje odgovarajućeg djelovanja robota. Dijagram toka cijelog postupka klasifikacije i planiranja djelovanja prikazan je na slici (Slika 5-3.).

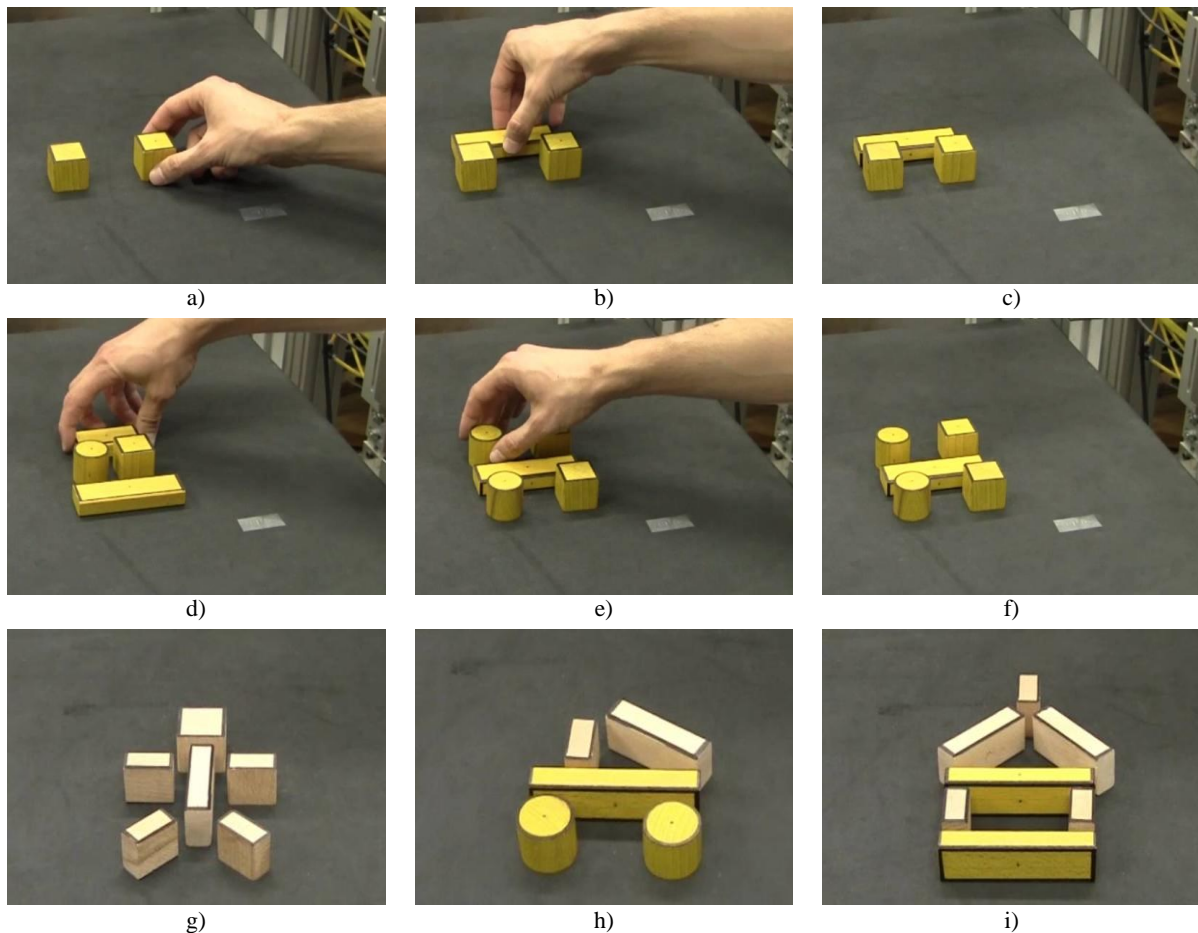


Slika 5-3. Dijagram toka validacije upravljačkog modela na laboratorijskoj opremi.

5.1. Učenje i klasifikacija

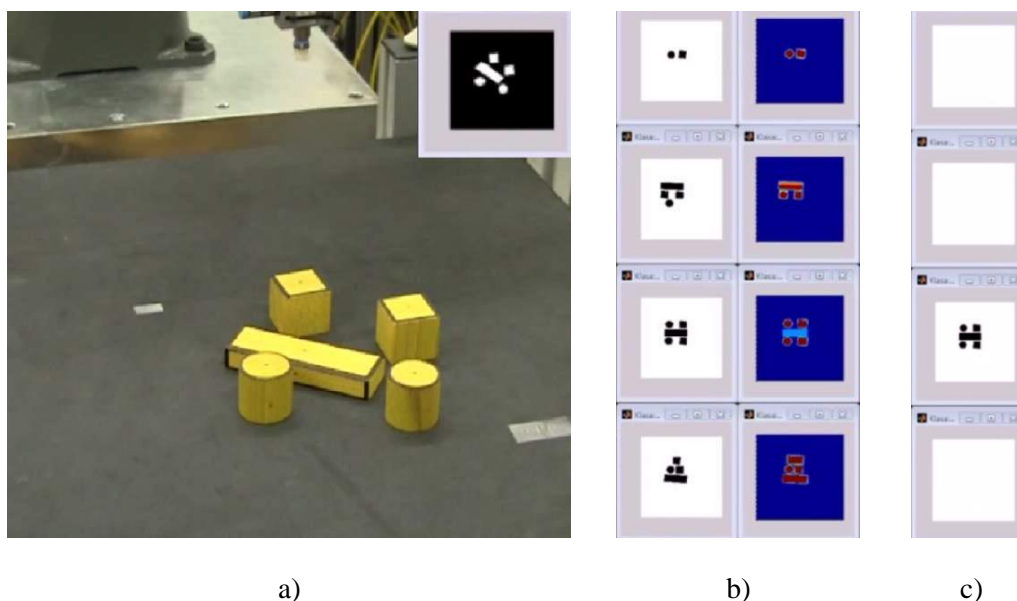
Kao osnovni način prijenosa informacija korisnika robotu u ovom je istraživanju izdvojena vizualna interpretacija radnog prostora robota, kao što je prikazano na slici (Slika 5-4.). Osnovni je cilj da krajnji korisnik kroz vizualnu demonstraciju, koja se ističe kao jedna od značajnijih komponenata interakcije robota i čovjeka [106], prikaže određen broj prostornih struktura. Ove prostorne strukture robot, odnosno razvijena ART mreža treba klasificirati i pohraniti. Svaku prostornu strukturu treba klasificirati na simboličkom nivou te na nivou koordinata u referentom koordinatnom sustavu robota. U ovom dijelu eksperimentalnog

istraživanja prikazan je niz različitih prostornih struktura sastavljenih od poznatih objekata. Pomoću vizijskog sustava i razvijene umjetne neuronske mreže predstavljene se strukture klasificiraju i pohranjuju. Podrazumijeva se da će se početni skup poznatih objekata moći nadopunjavati novim objektima u kasnijim fazama rada sustava. Na slici (Slika 5-4.) prikazan je proces vizualne demonstracije prostornih struktura, dok su strukture klasificirane korištenjem algoritma (Algoritam 2-2.).



Slika 5-4. Učenje prostornih struktura. Inicijalizacija robotskog sustava započinje demonstracijom različitih prostornih struktura koje se klasificiraju pomoću algoritma (Algoritam 2-2.).

Nakon postupka učenja i klasifikacije prostornih struktura potrebno je omogućiti asocijativno povezivanje lokaliziranog neuređenog prostornog stanja s postojećim klasificiranim strukturama. Ovaj postupak detaljno je opisan u poglavlju 3., dok je na slici (Slika 5-5.) prikazan jedan primjer lokalizirane strukture iz robotskog radnog prostora.



Slika 5-5. Povezivanje neuređenog prostornog stanja s prethodno klasificiranim prostornim strukturama: a) neuređeno stanje objekata u radnom prostoru robota i prikaz objekata lokaliziranih pomoću vizijskog sustava, b) postavljanje hipoteza o izgradnji poznatih klasa, c) izbor pobjedničke klase.

Za lokalizirane objekte u radnom prostoru robota generiraju se strukture na temelju funkcije asocijativnog povezivanja kao što je prikazano na slici (Slika 5-5.a). Prema funkciji (2.8) za svaku od postavljenih hipoteza (Slika 5-5.b) računa se vrijednost preklapanja te se izabire struktura s najvećom vrijednosti preklapanja. Pobjednička struktura, prikazana na slici (Slika 5-5.c) aplicira se u ART mrežu prema postupku iz algoritma (Algoritam 2-2.).

Cilj postupka tzv. logičkog i simboličkog prepoznavanja i povezivanja je pronaći adekvatno konačno stanje objekata u radnom prostoru tj. strukturu \mathbf{K} . Uz poznavanje početnog stanja objekata \mathbf{P} , sljedeći je cilj pronaći skup aktivnosti \mathcal{A} robota koji će omogućiti realnu promjenu radnog prostora te izvršiti prostornu transformaciju $\mathcal{A} : \mathbf{P} \rightarrow \mathbf{K}$. Taj je postupak opisan u narednom poglavlju.

5.2. Planiranje robotskog djelovanja

Planiranje djelovanja temelji se na prepoznavanju, odnosno asocijativnom povezivanju lokaliziranog neuređenog skupa objekata s poznatim naučenim i klasificiranim prostornim strukturama. Generiranje skupa aktivnosti \mathcal{A} koje transformiraju početno neuređeno stanje objekata \mathbf{P} u konačno stanje \mathbf{K} obavlja se u dva koraka:

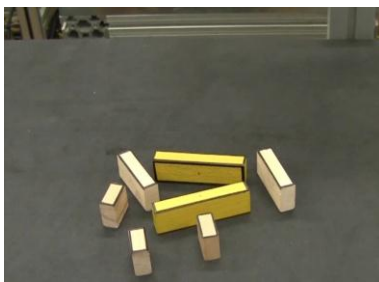
- konstruktivni algoritam (Algoritam 4-1.)

- genetski algoritam za optimizaciju inicijalnog skupa rješenja (Algoritam 4-2.).

Kako bi se ti algoritmi mogli iskoristiti za realno preslikavanje te generiranje plana koji je moguće izvršiti na eksperimentalnom postavu (Slika 5-1.), potrebno je kalibrirati postojeći robotski sustav i oba algoritma. Povezuju se sve dimenzije objekata kako bi se izvršilo preklapanje referentnih koordinatnih sustava [24] korištenih unutar razvijenih algoritama te koordinatnih sustava kojima se koristi robot. Nadalje, razvijen je robotski upravljački program koji ima sljedeće mogućnosti:

- dvosmjerna komunikacija preko TCT / IP protokola
- sinkronizacija radnji s aktivnostima generiranim u algoritmu planiranja djelovanja
- izbor odgovarajućih režima rada
- kalibracija koordinatnih sustava
- izvršavanje generiranog plana.

Osnovni cilj razvijenog robotskog upravljačkog programa je omogućiti izvođenje generiranog plana u realnim uvjetima. Na slici (Slika 5-6.a) prikazano je početno neuređeno stanje objekata za koje je prethodno generirano najsličnije konačno stanje (Slika 5-6.b). Generirani plan treba izvršiti na robotskom sustavu kako bi se početno stanje objekata transformiralo u konačno, kao što je prikazano na slici (Slika 5-6.c).



a)



b)



c)

Slika 5-6. a) početno stanje, b) generirano konačno stanje, c) izgrađeno konačno stanje.

5.2.1. Generiranje inicijalnog plana djelovanja

Za generiranje početnog skupa rješenja upotrebljava se konstruktivni algoritam i pripadajuća funkcija cilja (4.3). Usporedno se generira 30 rješenja koja se u svakoj iteraciji međusobno uspoređuju i vrednuju. Konstruktivni algoritam daje pojedince koji su relativno širokih raspona dobrote, što je vidljivo iz eksperimenata provedenih u fazi testiranja algoritma i

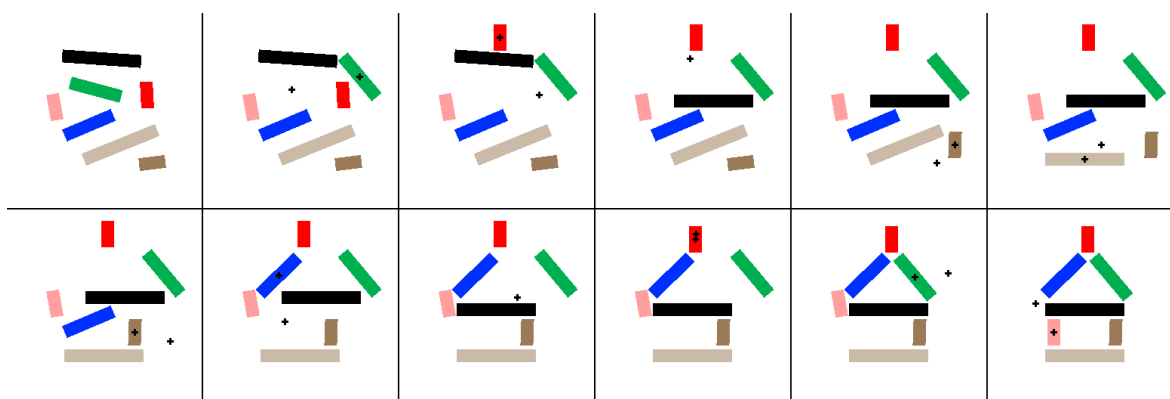
5. Validacija upravljačkog modela

prikazano na dijagramu (Slika 4-6.). Postupak inicijalnog kreiranja rješenja primjenjuje se za stohastičku pretragu prostora. Takav, široki raspon dobrote rješenja poželjan je kako bi se kasnijom primjenom modificiranih operatora mutacije i križanja mogao pretražiti još veći broj mogućih rješenja. Dobrota svih rješenja dobivena u postupku validacije prikazana je i grupirana u tablici (Tablica 5-1.).

Tablica 5-1. Raspodjela dobrote rješenja dobivene konstruktivnim algoritmom.

| | | | | | | | | | |
|-------------------------|------|------|------|------|------|------|------|------|------|
| $\Phi \leq 3500$ | 3163 | 3204 | 3205 | 3321 | 3344 | 3462 | | | |
| $3500 < \Phi \leq 4000$ | 3741 | 3809 | 3832 | 3835 | 3842 | 3853 | 3862 | 3894 | |
| $4000 < \Phi \leq 4500$ | 4002 | 4017 | 4033 | 4064 | 4102 | 4241 | 4343 | 4395 | 4402 |
| $\Phi > 4500$ | 4617 | 4631 | 4790 | 5670 | 6184 | 6420 | 6749 | | |

Nadalje, u tablici (Tablica 5-2.) prikazan je generirani plan djelovanja s najvećom dobrotom, dobiven pomoću konstruktivnog algoritma. Pripadajuća shema plana prikazana je na slici (Slika 5-7.). Objekt koji se pomiče ima označeno težište (+), dok se na pojedinoj slici nalazi prikaz prethodne pozicije zadanog objekta te njegova trenutna pozicija.



Slika 5-7. Najbolje rješenje (11 koraka) dobiveno konstruktivnim algoritmom – $\Phi_0=3163$.

Iz generiranog plana prikazanog na slici (Slika 5-7.) te pripadajućih pozicija robota prikazanih u tablici (Tablica 5-2.) vidljivo je da su tri objekta (2, 3 i 6) izravno postavljena na krajnji položaj iz **K**. Ostali objekti (1, 4, 5 i 7) su prije postavljanja na konačnu poziciju privremeno postavljeni na međupoložaje generirane konstruktivnim algoritmom. Zbog toga što je pomoću konstruktivnog algoritma tek inicijalno generirano rješenje, ono može sadržavati određene redundantne aktivnosti, odnosno radnje izuzimanja i odlaganja objekata.

5. Validacija upravljačkog modela

Tablica 5-2. Najbolje rješenje (11 koraka) dobiveno konstruktivnim algoritmom, $\{x_1, y_1, \delta_1\}$ – pozicija izuzimanja, $\{x_2, y_2, \delta_2\}$: pozicija odlaganja, $\Phi_0 = 3163$.

| Korak | Objekt | x1 | y1 | δ_1 | x2 | y2 | δ_2 |
|-------|--------|-----|-----|------------|-----|-----|------------|
| 1 | 7 | 137 | 153 | -15 | 120 | 243 | -50 |
| 2 | 5 | 144 | 221 | 93 | 68 | 169 | -89 |
| 3 | 1 | 96 | 161 | -4 | 152 | 192 | 0 |
| 4 | 4 | 234 | 228 | 7 | 210 | 252 | -92 |
| 5 | 2 | 210 | 186 | 22 | 229 | 164 | 0 |
| 6 | 4 | 210 | 252 | 7 | 198 | 205 | -92 |
| 7 | 6 | 184 | 144 | 23 | 123 | 136 | 44 |
| 8 | 1 | 152 | 192 | -4 | 168 | 164 | 0 |
| 9 | 5 | 68 | 169 | 93 | 75 | 168 | -89 |
| 10 | 7 | 120 | 243 | -15 | 125 | 199 | -50 |
| 11 | 3 | 160 | 99 | -80 | 198 | 123 | 91 |

5.2.2. Primjena genetskog algoritma

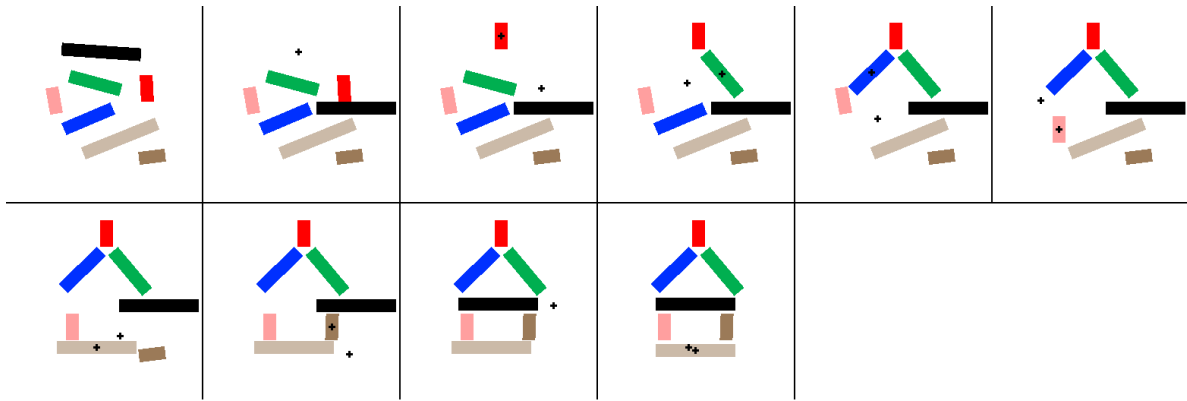
Svi inicijalno generirani planovi pomoću konstruktivnog algoritma daljnjim postupkom, detaljno objašnjenim u poglavlju 4.3.2., prolaze proces optimizacije. Uslijed uporabe operatora mutacije položaja i redoslijeda te operatora križanja, optimirana rješenja sadrže kombinaciju značajki većeg broja inicijalnih planova, kao i potpuno nove značajke (pozicije i redoslijede objekata u planu). U tablici (Tablica 5-3.) prikazana je raspodjela dobrote optimiranih rješenja.

Tablica 5-3. Raspodjela dobrote rješenja dobivene pomoću genetskog algoritma.

| | | | | | | | | | | |
|-------------------------|------|------|------|------|------|------|------|------|------|------|
| $\Phi \leq 2800$ | 2517 | 2630 | 2665 | 2698 | 2721 | 2735 | 2767 | 2777 | 2784 | 2799 |
| $2800 < \Phi \leq 3100$ | 2801 | 2806 | 2815 | 2816 | 2818 | 2824 | 2847 | 2873 | 2875 | 2876 |
| | 3007 | 3018 | 3024 | 3038 | 3085 | | | | | |
| $\Phi > 3100$ | 3231 | 3259 | 3322 | 3348 | 3376 | | | | | |

Na slici (Slika 5-8.) prikazano je najbolje rješenje dobiveno genetskim algoritmom nakon 300 iteracija, dok je pripadajući plan prikazan u tablici (Tablica 5-4.).

5. Validacija upravljačkog modela



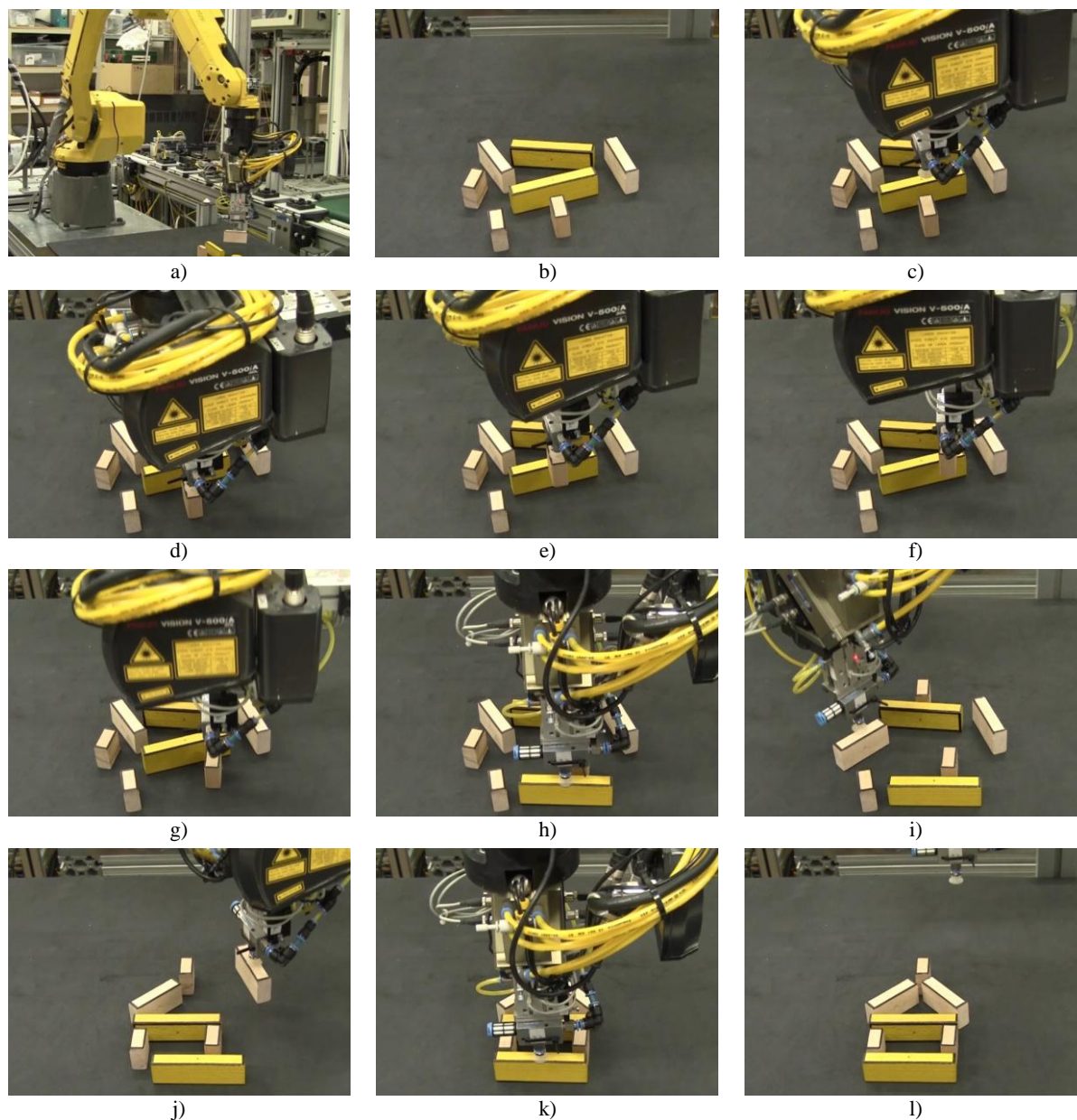
Slika 5-8. Najbolje rješenje (9 koraka) dobiveno genetskim algoritmom – $\Phi_K=2517$.

Tablica 5-4. Najbolje rješenje (9 koraka) dobiveno genetskim algoritmom, $\{x_1, y_1, \delta_1\}$: pozicija izuzimanja, $\{x_2, y_2, \delta_2\}$: pozicija odlaganja, $\Phi_K = 2517$.

| Korak | Objekt | x_1 | y_1 | δ_1 | x_2 | y_2 | δ_2 |
|-------|--------|-------|-------|------------|-------|-------|------------|
| 1 | 1 | 96 | 161 | -4 | 170 | 237 | 0 |
| 2 | 5 | 144 | 221 | 93 | 75 | 168 | -89 |
| 3 | 7 | 137 | 153 | -15 | 125 | 199 | -50 |
| 4 | 6 | 184 | 144 | 23 | 123 | 136 | 44 |
| 5 | 3 | 160 | 99 | -80 | 198 | 123 | 91 |
| 6 | 2 | 210 | 186 | 22 | 225 | 155 | 0 |
| 7 | 4 | 234 | 228 | 7 | 198 | 205 | -92 |
| 8 | 1 | 170 | 237 | 0 | 168 | 164 | 0 |
| 9 | 2 | 225 | 155 | 0 | 229 | 164 | 0 |

Sa slike (Slika 5-8.) te iz tablice (Tablica 5-4.) u kojoj su prikazane pripadajuće pozicije robota vidljivo je da je nakon procesa optimizacije dobiveno poboljšanje te je pet objekata (3, 4, 5, 6 i 7) izravno postavljeno na krajnji položaj **K**. Ostali objekti (1, 2) su prije postavljanja na konačnu poziciju privremeno postavljeni na međupoložaje, kao što je bio slučaj i kod konstruktivnog algoritma. Nadalje, najveća dobrota rješenja je s početnih $\Phi_0 = 3163$ optimirana na $\Phi_K = 2517$.

Na slici (Slika 5-9.) prikazani su isječci iz verifikacije navedenog plana na robotskom sustavu u Laboratoriju za projektiranje izradbenih i montažnih sustava. Svaki od objekata potrebno je lokalizirati pomoću vizijskog sustava, generirati točku prilaza (engl. *approach point*) (Slika 5-9.c), točku odmicanja (engl. *retract point*) (Slika 5-9.d) za poziciju izuzimanja. Nadalje, analogni postupak se ponavlja za poziciju odlaganja, što je prikazano na slici (Slika 5-9.f, Slika 5-9.g). Nakon izvođenja plana djelovanja izgrađena struktura odgovara onoj iz plana prema slici (Slika 5-8.).



Slika 5-9. Izvršavanje plana na realnom robotskom sustavu: a) prikaz robota u početnoj točki, b) neuređeno početno stanje objekata, c) – g) izuzimanje i odlaganje prvog objekta, h) – k) izuzimanje i odlaganje ostalih objekata, l) izgrađena uređena prostorna struktura.

5.3. Sažeti pregled postupka validacije

Validacija razvijenih algoritama na realnoj istraživačkoj opremi omogućila je uvid u mogućnosti povezivanja teorijski razvijenih algoritama s realnim robotskim sustavom. U procesu kalibracije, koji je preduvjet za dimenzijsko povezivanje prostora djelovanja robota i virtualnog prostora ART mreže i genetskog algoritma, te u kontinuiranoj komunikaciji između svih upravljačkih programa omogućeno je da se razvijeni algoritmi i metodologija iz

5. Validacija upravljačkog modela

poglavlja 2., 3. i 4. implementiraju i uspješno verificiraju na postojećoj opremi. Pseudokod robotskog upravljačkog programa prikazan je u algoritmu (Algoritam 5-1.).

Algoritam 5-1. *Robotski upravljački program.*

Algoritam: MAIN

```
Ulaz:  $\mathbf{I}, \mathbf{I}_{MTO}, \mathcal{W}^t, \beta_1, \beta_2, \varphi_1, \varphi_2, Pc, Pm, br\_generacija, faktor\_mut$   
Izlaz:  $Robot\_plan, err$   
1 // Centralni upravljački program koji služi za upravljanje rada  
  robota.  
2  
3 Dok izbor  $\neq$  „kraj“ ponavljaj  
4   izbor  $\leftarrow$  korisnik izabire vrstu rada: {učenje, povezivanje_i_planiranje, kraj}  
5   ako izbor = „učenje“ onda  
6     // Proces učenja.  
7      $\mathcal{W}^{t+1} \leftarrow$  Dvorazinski_ART( $\mathcal{W}^t, \mathbf{I}, \beta_1, \beta_2, \varphi_1, \varphi_2$ )  
8   kraj  
9   ako izbor = „planiranje“ onda  
10    // Povezivanje neuređenog prostornog  $\mathbf{I}_{MTO}$  stanja s poznatim  
    strukturama iz  $\mathcal{W}$ .  
11    ( $\mathbf{I}_{WIN}, klasa\_rb1, klasa\_rb2$ )  $\leftarrow$  Povezivanje_prostornih_stanja( $\mathbf{I}_{MTO}, \mathcal{W}$ )  
12    // Ažuriranje  $\mathcal{W}\{rb1\}\{rb2\}$  sa  $\mathbf{I}_{WIN}$ .  
13     $\mathcal{W}^{t+1} \leftarrow$  Dvorazinski_ART( $\mathcal{W}^t, \mathbf{I}_{WIN}, \beta_1, \beta_2$ )  
14    // Planiranje djelovanja 1.korak: kreiranje inicijalnog skupa  
    rješenja  $\mathcal{O}$ .  
15     $\mathcal{O} \leftarrow$  Konstruktivni_algoritam ( $\mathbf{P}, \mathbf{K}, br, u$ )  
16    // Optimizacija inicijalnog skupa rješenja.  
17     $\mathbf{O}_{OPT} \leftarrow$  Optimizacijski_GA ( $\mathcal{O}, Pc, Pm, br\_generacija, faktor\_mut$ )  
18    // Izvršavanje plana na robotu.  
19    za svaki  $O_i \in \mathbf{O}_{OPT}$   
20      | izuzmi predmet s položaja  $O_i(p1)$  i prenesi ga na  $O_i(p2)$   
21    kraj  
22  kraj  
23 kraj
```

6.ZAKLJUČAK

U doktorskom radu razvijen je upravljački model robota koji omogućuje klasifikaciju i prepoznavanje prostornih struktura prema obliku i rasporedu objekata. Nadalje, razvijena je nova funkcija koja povezuje neuređeno prostorno stanje objekata s prethodno klasificiranim strukturama sa svrhom asocijativnog povezivanja trenutnog stanja objekata s postojećim klasificiranim strukturama. Generirano konačno stanje objekata služi za planiranje robotskog djelovanja čiji je cilj izgradnja uređene prostorne strukture iz neuređenog stanja objekta lokaliziranih na sceni (radni prostor robota). U tu svrhu razvijen je novi genetski algoritam planiranja robotskog djelovanja.

Za prepoznavanje i klasifikaciju prostornih struktura koje su definirane kao uređeni skup konačnog broja poznatih objekata razvijena je nova neuronska mreža temeljena na teoriji adaptivne rezonancije. Osnovne karakteristike razvijene mreže očituju se u dvorazinskoj klasifikaciji i klasifikaciji struktura prema obliku i rasporedu pojedinačnih objekata unutar strukture. Na temelju toga razvijen je i implementiran novi podsustav prilagodbe vrijednosti rezonancije. Provedena je analiza brzine klasifikacije predložene ART mreže te je eksperimentalno utvrđen raspon parametra budnosti mreže $\varphi \in [\sim 0,85, 0,95]$ koji rezultira visokom brzinom klasifikacije. Nadalje, napravljena je statistička analiza točnosti klasifikacije, gdje su ispitani svi značajni parametri klasifikacije kao što su odziv, specifičnost i točnost te je napravljen Kappa statistički test. Svrha povezivanja neuređenog prostornog stanja objekata s postojećim klasificiranim strukturama je pronaći najbolje preslikavanje na razini objekata pomoću kojih je moguće izgraditi poznate (ili dovoljno slične) uređene strukture. Predložena je funkcija povezivanja koja traži najveću vrijednost kumulativne sume sličnosti objekata (tj. njihovih oblika) primjenom Kuhn-Munkresova algoritma i prethodno opisane ART neuronske mreže. Takvo povezivanje omogućava da se iz objekata na sceni koji su različiti od onih u postojećim klasificiranim strukturama izgradi najbližnja prostorna struktura.

Osim logičkog povezivanja prostornih stanja objekata, u okviru eksperimentalnog i primijenjenog dijela istraživanja, bitno je i realno preslikavanje. Realno tj. fizičko

preslikavanje uključuje generiranje uređenog skupa elemenata rada robota koji će na stvarnoj sceni tj. u radnom prostoru robota omogućiti izgradnju prethodno izabrane prostorne strukture bez kolizija između pojedinih objekata. Radi toga je razvijen novi genetski algoritam planiranja robotskog djelovanja sastavljen od dviju osnovnih komponenata: I) konstruktivne heuristike za generiranje inicijalnog skupa rješenja te II) genetskog algoritma za optimiranje inicijalnih rješenja. Razvijena konstruktivna heuristika temelji se na principu usporednog kreiranja planova djelovanja koji se međusobno uspoređuju i verificiraju. Međusobnom se evaluacijom planovi najveće parcijalne dobrote dalje razvijaju, dok se planovi s visokom razinom entropije ukidaju. Turnirskom se selekcijom najbolja rješenja u pojedinoj generaciji kloniraju. Konstruktivna heuristika koristi se isključivo operatorima mutacije bez operatora križanja jer rješenja u ovoj fazi još uvijek nisu dovršena. Nakon uspješno generiranih planova korištenjem konstruktivnog algoritma, u drugoj se fazi primjenjuju operatori mutacije i križanja kako bi se od početnih konstruktivnih rješenja (pojedinaca) u određenom broju iteracija generirali pojedinci veće dobrote. Primijenjeni su modificirani operatori mutacije prilagođeni postavljenom problemu, novi operator mutacije položaja objekata u planu te modificirani operator križanja pojedinaca (planova). U svrhu korištenja genetskog algoritma predloženo je kodiranje rješenja u obliku kromosoma primjenom dvaju međusobno povezanih vektora: vektora permutacija redoslijeda te vektora položaja objekata. Pojedini operatori mutacije $M1^1$ - $M3^1$ i operator križanja mijenjaju vektor permutacije redoslijeda, dok operator mutacije $M4$ mijenja vektor položaja. Međusobnom kombinacijom svih operatora dolazi do konvergencije rješenja, a samim time i generiranjem plana robotskog djelovanja veće dobrote. Usporedbom razvijenog genetskog algoritma s algoritmom koji se koristi pohlepnom heuristikom (pohlepno pretraživanje) dobiveno je prosječno poboljšanje dobrote najboljeg rješenja od ~25%.

Sa svrhom verifikacije predloženog upravljačkog modela izvršena je implementacija na industrijskom robotu sa šest stupnjeva slobode gibanja. Razvijeni su robotski upravljački programi koji omogućavaju da korisnik u stvarnom okruženju, koristeći poznate objekte, demonstrira određene prostorne strukture koje robot prepoznaje i klasificira pomoću ART mreže. Sljedeći korak uključuje demonstraciju neuređenog stanja objekata u radnom području robota koje se asocijativno povezuje s najsličnijim klasificiranim strukturama koristeći razvijenu funkciju povezivanja. Izabire se pobjednička struktura prema najvećoj razini sličnosti. Pobjednička struktura označava konačno stanje u koje lokalizirane objekte treba postaviti iz početnog stanja. Izgradnja strukture planira se primjenom prethodno opisanog

genetskog algoritma u dva koraka. Generirani plan najveće dobrote šalje se s upravljačkog računala na upravljačku jedinicu robota koja izvodi plan.

Zaključno, može se reći da je postavljeni cilj „razvoja i verifikacije upravljačkog modela koji će osposobiti robota za prepoznavanje i izgradnju uređenih prostornih struktura“ ispunjen. Nadalje, potvrđene su postavljene hipoteze „prepoznavanja i klasifikacije prostornih struktura interpretacijom položaja te rasporeda objekata“ kao i „planiranja izgradnje takvih struktura korištenjem novog genetskog algoritma“. S obzirom na stohastičku prirodu genetskih algoritama, ne postoji formalni dokaz o konvergenciji algoritma u globalno optimalno područje, što je čest slučaj prilikom razvoja ove vrste algoritama [105].

Kao izvorni znanstveni doprinosi ovog istraživanja ističu se:

- Samoorganizirajuća umjetna neuronska mreža za klasifikaciju i prepoznavanje prostornih struktura zasnovana na teoriji adaptivne rezonancije, koju u sklopu upravljačkog modela odlikuje nova dvorazinska klasifikacija po obliku i rasporedu objekata te mehanizam asocijativnog povezivanja neuređenog skupa objekata s uređenim.
- Novi genetski algoritam za planiranje robotskog djelovanja u nestrukturiranoj radnoj okolini karakteriziran usporednom evolucijskom strategijom za pronalaženje rješenja, s ciljem prostorne pretvorbe neuređenog stanja objekata u uređeno, ograničen na dvodimenzionalnu interpretaciju prostora.
- Upravljački model sastavljen od ART neuronske mreže, genetskog algoritma, te razvijenih robotskih upravljačkih programa.

Buduća istraživanja uključit će mogućnost implementacije prepoznavanja te klasifikacije objekata i struktura kroz trodimenzionalnu interpretaciju radnog prostora. Kao prvi korak planira se uvođenje parametra visine pojedinog objekta koji će karakterizirati komponentu z geometrije objekta. Sveobuhvatni prikaz profila visine pojedinih objekata, a samim time i prostorne strukture, omogućit će za detaljniju klasifikaciju različitih prostornih struktura koje tada neće biti isključivo karakterizirane dvodimenzionalnom interpretacijom.

Jedna od motivacija za ovo istraživanje bila je i mogućnost implementacije upravljačkog modela na robotima koji se nalaze u Laboratoriju za projektiranje izradbenih i montažnih sustava. U okviru ovog istraživanja verificirani su robotski upravljački programi na Fanuc M10iA robotu. Nadalje, moguće je generalizirati razvijene upravljačke programe sa svrhom

6. Zaključak

implementacije na širem spektru robotskih sustava različitih proizvođača koji se nalaze u laboratoriju a uključuju robote proizvođača Universal Robots, KUKA, IAI, kao i druge modele Fanuc robota.

LITERATURA

- [1] L. Steels. When are robots intelligent autonomous agents? , *Robotics and Autonomous Systems*, vol. 15, no. 1–2, p. 3–9, Jul. 1995.
- [2] A. Knoll. The dawn of the age of autonomy, Technische Universität München, Technical Report, 2014.
- [3] B. Siciliano, O. Khatib, Springer handbook of robotics. Berlin: Springer, 2008.
- [4] B. Jerbić, G. Nikolić, D. Chudy, M. Švaco, B. Šekoranja. Robotic application in neurosurgery using intelligent visual and haptic interaction, *International Journal of Simulation Modelling*, vol. 14, no. 1, p. 71–84, 2015.
- [5] G. H. Lim, I. H. Suh, H. Suh. Ontology-Based Unified Robot Knowledge for Service Robots in Indoor Environments, *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 3, p. 492–509, May 2011.
- [6] Tang Li, Fu Shuangfei, Fang Lijin, Wang Hongguang. Obstacle-navigation strategy of a wire-suspend robot for power transmission lines, 2004, p. 82–87.
- [7] B. D. Argall, S. Chernova, M. Veloso, B. Browning. A survey of robot learning from demonstration, *Robotics and Autonomous Systems*, vol. 57, no. 5, p. 469–483, May 2009.
- [8] G. Konidaris, S. Kuindersma, R. Grupen, A. Barto. Robot learning from demonstration by constructing skill trees, *The International Journal of Robotics Research*, p. 0278364911428653, 2011.
- [9] D. H. Grollman, O. C. Jenkins. Incremental learning of subtasks from unsegmented demonstration, *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, p. 261–266.
- [10] L. Jun, T. Duckett. Some practical aspects on incremental training of RBF network for robot behavior learning, *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, 2008, p. 2001–2006.
- [11] M. F. Martins, Y. Demiris. Learning multirobot joint action plans from simultaneous task execution demonstrations, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 2010, p. 931–938.

- [12] M. Sridharan, J. Wyatt, R. Dearden. Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs, *Artificial Intelligence*, vol. 174, no. 11, p. 704–725, Jul. 2010.
- [13] N. E. Du Toit, J. W. Burdick. Robot Motion Planning in Dynamic, Uncertain Environments, *IEEE Transactions on Robotics*, vol. 28, no. 1, p. 101–115, Feb. 2012.
- [14] P. Raja. Optimal path planning of mobile robots: A review, *International Journal of the Physical Sciences*, vol. 7, no. 9, Feb. 2012.
- [15] J. Bidot, L. Karlsson, F. Lagriffoul, A. Saffiotti. Geometric Backtracking for Combined Task and Path Planning in Robotic Systems, *Artificial Intelligence*, 2013.
- [16] F. Zacharias, C. Borst, G. Hirzinger. Bridging the gap between task planning and path planning, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, p. 4490–4495.
- [17] A. Amedi. Convergence of Visual and Tactile Shape Processing in the Human Lateral Occipital Complex, *Cerebral Cortex*, vol. 12, no. 11, p. 1202–1212, Nov. 2002.
- [18] R. Dillmann. Teaching and learning of robot tasks via observation of human performance, *Robotics and Autonomous Systems*, vol. 47, no. 2–3, p. 109–116, Jun. 2004.
- [19] D. Domijan. Neuronske mreže za percepciju svjetlina i oblika, *Filozofski fakultet, Sveučilište u Zagrebu*, 2000.
- [20] G. A. Carpenter, S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 1, p. 54–115, Jan. 1987.
- [21] S. Grossberg. Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world, *Neural Networks*, vol. 37, p. 1–47, Jan. 2013.
- [22] S. Grossberg. From brain synapses to systems for learning and memory: Object recognition, spatial navigation, timed conditioning, and movement control, *Brain Research*, Nov. 2014.
- [23] G. A. Carpenter, W. D. Ross. ART-EMAP: A neural network architecture for object recognition by evidence accumulation, *IEEE Transactions on Neural Networks*, vol. 6, no. 4, p. 805–818, Jul. 1995.
- [24] P. I. Corke, *Robotics, vision and control: fundamental algorithms in MATLAB*. Berlin: Springer, 2011.
- [25] V. R. Kompella, M. Luciw, M. Stollenga, L. Pape, J. Schmidhuber. Autonomous learning of abstractions using curiosity-driven modular incremental slow feature

- analysis, Development and Learning and Epigenetic Robotics (ICDL), 2012 IEEE International Conference on, 2012, p. 1–8.
- [26] P. Rouanet, P.-Y. Oudeyer, F. Danieau, D. Filliat. The Impact of Human-Robot Interfaces on the Learning of Visual Objects, *IEEE Transactions on Robotics*, vol. 29, no. 2, p. 525–541, Apr. 2013.
- [27] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, A. Stentz. Perceiving, learning, and exploiting object affordances for autonomous pile manipulation, *Autonomous Robots*, vol. 37, no. 4, p. 369–382, Dec. 2014.
- [28] J. Leitner, S. Harding, M. Frank, A. Forster, J. Schmidhuber. Learning Spatial Object Localization from Vision on a Humanoid Robot, *International Journal of Advanced Robotic Systems*, vol. 9, p. 1–10, 2012.
- [29] Yang Yang, I. Saleemi, M. Shah. Discovering Motion Primitives for Unsupervised Grouping and One-Shot Learning of Human Actions, Gestures, and Expressions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, p. 1635–1648, Jul. 2013.
- [30] S. Kirstein, H. Wersing, E. Körner. A biologically motivated visual memory architecture for online learning of objects, *Neural Networks*, vol. 21, no. 1, p. 65–77, Jan. 2008.
- [31] Y. Wu, Y. Su, Y. Demiris. A morphable template framework for robot learning by demonstration: Integrating one-shot and incremental learning approaches, *Robotics and Autonomous Systems*, vol. 62, no. 10, p. 1517–1530, Oct. 2014.
- [32] Motonobu Hattori. *Avoiding Catastrophic Forgetting by a Biologically Inspired Dual-Network Memory Model*, vol. 7664, Springer Berlin Heidelberg, 2012.
- [33] M. Hattori. A biologically inspired dual-network memory model for reduction of catastrophic forgetting, *Neurocomputing*, vol. 134, p. 262–268, Jun. 2014.
- [34] G. A. Carpenter, S. Grossberg. ART 2: Self-organization of stable category recognition codes for analog input patterns, *Applied optics*, vol. 26, no. 23, p. 4919–4930, 1987.
- [35] G. A. Carpenter, S. Grossberg, D. B. Rosen. ART 2-A: An Adaptive Resonance Algorithm Category Learning and Recognition.
- [36] G. A. Carpenter, S. Grossberg. ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures, *Neural networks*, vol. 3, no. 2, p. 129–152, 1990.
- [37] G. A. Carpenter, S. Grossberg, D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system., *Neural Networks*, vol. 4, no. 6, p. 759–771, 1991.

- [38] A. Akhbardeh, A. Varri. Novel supervised fuzzy adaptive resonance theory (SF-ART) neural network for pattern recognition, *Intelligent Signal Processing, 2005 IEEE International Workshop on*, 2005, p. 149–154.
- [39] A.-H. Tan, G. A. Carpenter, S. Grossberg. *Intelligence through interaction: Towards a unified theory for learning*, *Advances in Neural Networks–ISNN 2007*, Springer, 2007, p. 1094–1103.
- [40] J. C. Castuera, I. Lopez-Juarez. *Intelligent Task Level Planning for Robotic Assembly: Issues and Experiments*, *MICAI 2004: Advances in Artificial Intelligence*, vol. 2972, R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, H. Sossa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, p. 872–881.
- [41] I. Lopez-Juarez, M. Howarth. *Learning manipulative skills with art*, *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, 2000, vol. 1, p. 578–583.
- [42] L. Gu, J. Su. *Humanoid Robot Behavior Learning Based on ART Neural Network and Cross-Modality Learning*, *Advances in Natural Computation*, vol. 4221, L. Jiao, L. Wang, X. Gao, J. Liu, F. Wu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, p. 447–450.
- [43] P. K. Park, J. H. Lee, C. W. Shin, H.-S. Ryu, B.-C. Kang, G. A. Carpenter, S. Grossberg. *Gesture recognition system based on Adaptive Resonance Theory*, *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, p. 3818–3822.
- [44] M. Yavaş, F. N. Alpaslan. *Hierarchical behavior categorization using correlation based adaptive resonance theory*, *Neurocomputing*, vol. 77, no. 1, p. 71–81, Feb. 2012.
- [45] S., Seyit Sabri. *Simple and complex behavior learning using behavior hidden markov model and CobART*, Middle East Technical University, 2013.
- [46] E. Berghöfer, D. Schulze, C. Rauch, M. Tscherepanow, T. Köhler, S. Wachsmuth. *ART-based fusion of multi-modal perception for robots*, *Neurocomputing*, vol. 107, p. 11–22, May 2013.
- [47] D. C. Wunsch. *ART properties of interest in engineering applications*, *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 2009, p. 3380–3383.
- [48] W. Fung, Y. Liu. *Adaptive categorization of ART networks in robot behavior learning using game-theoretic formulation*, *Neural Networks*, vol. 16, no. 10, p. 1403–1420, Dec. 2003.
- [49] S. S. Seyhan, F. N. Alpaslan, M. Yavaş. *Simple and complex behavior learning using behavior hidden Markov model and CobART*, *Neurocomputing*, vol. 103, p. 121–131, Mar. 2013.

- [50] M. Tscherepanow, M. Kortkamp, M. Kammer. A hierarchical ART network for the stable incremental learning of topological structures and associations from noisy data, *Neural Networks*, Jun. 2011.
- [51] M. J.-Y. Chung, M. Forbes, M. Cakmak, R. P. N. Rao. Accelerating imitation learning through crowdsourcing, 2014, p. 4777–4784.
- [52] R. Dillmann, T. Asfour, M. Do, R. Jäkel, A. Kasper, P. Azad, A. Ude, S. R. Schmidt-Rohr, M. Lösch. *Advances in Robot Programming by Demonstration*, *KI - Künstliche Intelligenz*, vol. 24, no. 4, p. 295–303, Nov. 2010.
- [53] J. Corona-Castuera, R. Rios-Cabrera, I. Lopez-Juarez, M. Peña-Cabrera. An Approach for Intelligent Fixtureless Assembly: Issues and Experiments, *MICAI 2005: Advances in Artificial Intelligence*, vol. 3789, A. Gelbukh, Á. de Albornoz, H. Terashima-Marín, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, p. 1052–1061.
- [54] R. Dearden, C. Burbridge. Manipulation planning using learned symbolic state abstractions, *Robotics and Autonomous Systems*, vol. 62, no. 3, p. 355–365, Mar. 2014.
- [55] S. Ekvall, D. Kragic. Robot learning from demonstration: a task-level planning approach, *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, p. 223–234, 2008.
- [56] B. Subagdja, A.-H. Tan. iFALCON: A neural architecture for hierarchical planning, *Neurocomputing*, vol. 86, p. 124–139, Jun. 2012.
- [57] M. Rickert, M. E. Foster, M. Giuliani, T. By, G. Panin, A. Knoll. Integrating language, vision and action for human robot dialog systems, *Universal Access in Human-Computer Interaction. Ambient Interaction*, Springer, 2007, p. 987–995.
- [58] Jianwei Zhang, A. Knoll. A two-arm situated artificial communicator for human-robot cooperative assembly, *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, p. 651–658, Aug. 2003.
- [59] C. Lenz, S. Nair, M. Rickert, A. Knoll, W. Rosel, J. Gast, A. Bannat, F. Wallhoff. Joint-action for humans and industrial robots for assembly tasks, 2008, p. 130–135.
- [60] C. Lenz. Context-aware human-robot collaboration as a basis for future cognitive factories, Technische Universität München, 2011.
- [61] E. Guizzo, E. Ackerman. The rise of the robot worker, *Spectrum, IEEE*, vol. 49, no. 10, p. 34–41, 2012.
- [62] M. Stenmark, J. Malec. Knowledge-based instruction of manipulation tasks for industrial robotics, *Robotics and Computer-Integrated Manufacturing*, Sep. 2014.

- [63] K. Puljić. An Evolutionary Algorithm Based on Repeated Mutations for Solving the Capacitated Vehicle Routing Problem, *Journal of Computing and Information Technology*, vol. 20, no. 1, 2012.
- [64] P. T. Zacharia, N. A. Aspragathos. Optimal robot task scheduling based on genetic algorithms, *Robotics and Computer-Integrated Manufacturing*, vol. 21, no. 1, p. 67–79, Feb. 2005.
- [65] B. Jerbić, K. Grolinger, B. Vranješ. Autonomous agent based on reinforcement learning and adaptive shadowed network, *Artificial intelligence in engineering*, vol. 13, no. 2, p. 141–157, 1999.
- [66] B. Jerbić. Autonomous robotic assembly using collaborative behavior based agents, *International journal of Smart engineering system design*, vol. 4, no. 1, p. 11–20, 2002.
- [67] M. Švaco, B. Jerbić, F. Šuligoj. Autonomous Robot Learning Model Based On Visual Interpretation Of Spatial Structures, *Transactions of FAMENA*, vol. 38, no. 4, p. 13–28, 2014
- [68] M. Švaco, B. Jerbić, F. Šuligoj. ARTgrid: A Two-Level Learning Architecture Based on Adaptive Resonance Theory, *Advances in Artificial Neural Systems*, vol. 2014, p. 1–9, 2014.
- [69] M. Švaco, B. Šekoranja, B. Jerbić. Autonomous Planning Framework for Distributed Multiagent Robotic Systems, *Technological Innovation for Sustainability*, Springer, 2011, p. 147–154.
- [70] M. Švaco, B. Šekoranja, B. Jerbić. Industrial Robotic System with Adaptive Control, *Procedia Computer Science*, 2012, vol. 12, p. 164–169.
- [71] D. D. Hoffman. What do we mean by the structure of the world?, *Perception as Bayesian inference*, D. C. Knill, W. Richards, Eds. Cambridge, U.K. ; New York: Cambridge University Press, 1996, p. 219–221.
- [72] F. Crick, *The astonishing hypothesis: the scientific search for the soul*. New York: Simon & Schuster, 1995.
- [73] D. R. Hofstadter. Analogy as the core of cognition, *The analogical mind: Perspectives from cognitive science*, p. 499–538, 2001.
- [74] G. A. Carpenter, S. Grossberg. Adaptive Resonance Theory, *Encyclopedia of machine learning*, Springer, 2011.
- [75] J. He, A.-H. Tan, C.-L. Tan. ART-C: A neural architecture for self-organization under constraints, *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2002, p. 2550–2555.

- [76] M. Georgiopoulos, I. Dagher, G. L. Heileman, G. Bebis. Properties of learning of a fuzzy ART variant, *Neural networks*, vol. 12, no. 6, p. 837–850, 1999.
- [77] Wing Yee Sit, Lee Onn Mak, Gee Wah Ng. Managing Category Proliferation in Fuzzy ARTMAP Caused by Overlapping Classes, *IEEE Transactions on Neural Networks*, vol. 20, no. 8, p. 1244–1253, Aug. 2009.
- [78] A. Ben-David. About the relationship between ROC curves and Cohen's kappa, *Engineering Applications of Artificial Intelligence*, vol. 21, no. 6, p. 874–882, Sep. 2008.
- [79] N. Bogunović. Otkrivanje znanja u skupovima podataka.
http://www.zemris.fer.hr/predmeti/kdisc/Priprema_Evaluacija.ppt (preuzeto 14. rujna 2014.).
- [80] V. Labatut, H. Cherifi. Analysis of accuracy measures for classifiers comparison, *Ubiquitous Computing and Communication*, vol. Special Issue on ICIT 2011, p. 21–34, 2011.
- [81] M. Sokolova, G. Lapalme. A systematic analysis of performance measures for classification tasks, *Information Processing & Management*, vol. 45, no. 4, p. 427–437, Jul. 2009.
- [82] E. Alpaydin, *Introduction to machine learning*, 2nd ed. Cambridge, Mass: MIT Press, 2010.
- [83] B. Dalbelo Bašić, J. Šnajder. Vrednovanje klasifikatora. FER, 2011.
- [84] K. Krippendorff, *Content analysis: an introduction to its methodology*, 2nd ed. Thousand Oaks, Calif: Sage, 2004.
- [85] H. W. Kuhn. The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, p. 83–97, Mar. 1955.
- [86] Mordecai J. Golin. *Bipartite Matching and the Hungarian Method*. Hong Kong University of Science and Technology, 2006.
- [87] T. Lipić. Problemi pridruživanja i Mađarska metoda, *Kombinatorički algoritmi*, FER, 2011.
- [88] S. J. Russell, P. Norvig, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River, N.J: Prentice Hall/Pearson Education, 2010.
- [89] H. A. Simon, *The sciences of the artificial*. Cambridge, Mass.: MIT Press, 1996.
- [90] M. Ghallab, *Automated planning: theory and practice*. Amsterdam: Elsevier/Morgan Kaufmann, 2004.

- [91] P. Čurković, B. Jerbić, T. Stipančić. Coordination of Robots with Overlapping Workspaces Based on Motion Co-evolution, *International Journal of Simulation Modelling*, vol. 12, no. 1, p. 27–38, 2013.
- [92] D. Sidobre, W. He. Online task space trajectory generation, *Workshop on Robot Motion Planning: Online, Reactive, and in Real-time 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, 2012*, p. 121–128.
- [93] M. Švaco, B. Šekoranja, B. Jerbić. A multiagent framework for industrial robotic applications, *Procedia Computer Science*, 2011, vol. 6, p. 291–296.
- [94] D. Miklić. Grid-based coordinated control of mobile robot formations, *Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu*, 2011.
- [95] E. Kreyszig, *Advanced engineering mathematics*, 10th ed. Hoboken, NJ: John Wiley, 2011.
- [96] C. Blum, A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys*, vol. 35, no. 3, p. 268–308, Sep. 2003.
- [97] F. Glover, G. A. Kochenberger, Eds., *Handbook of metaheuristics*. Boston: Kluwer Academic Publishers, 2003.
- [98] K. Puljić, R. Manger. A distributed evolutionary algorithm with a superlinear speedup for solving the vehicle routing problem, *Computing and informatics*, vol. 31, no. 3, p. 675–692, 2012.
- [99] G. K. Gultekin, M. O. Ogucu, E. Sayginer, A. Saranli. A genetic algorithm based solution to constrained component placement of a mobile robot, 2012, p. 1–6.
- [100] P. Larranaga, C.M. H. Kuijpers, I. Inza, S. Dizdarevic. Genetic algorithms for the travelling salesman problem: a review of representations and operators, *Artificial Intelligence Review*, vol. 13, p. 129–170, 1999.
- [101] P. Pongcharoen, D. J. Stewardson, C. Hicks, P. M. Braiden. Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry, *Journal of Applied Statistics*, vol. 28, no. 3–4, p. 441–455, Mar. 2001.
- [102] H.-S. Hwang. An improved model for vehicle routing problem with time constraint based on genetic algorithm, *Computers & Industrial Engineering*, vol. 42, no. 2–4, p. 361–369, Apr. 2002.
- [103] M. Mitchell, *An introduction to genetic algorithms*. Cambridge, Mass.: MIT Press, 1996.

- [104] C. Rego, D. Gamboa, F. Glover, C. Osterman. Traveling salesman problem heuristics: Leading methods, implementations and latest advances, *European Journal of Operational Research*, vol. 211, no. 3, p. 427–441, Jun. 2011.
- [105] P. Čurković. *Evolutivni algoritam za upravljanje višeagentskim robotskim sustavom*, Fakultet strojarstva i brodogradnje, Sveučilište u Zagrebu, 2010.
- [106] C. Breazeal, B. Scassellati. Robots that imitate humans, *Trends in cognitive science*, vol. 6, no. 11, p. 481–487, 2002.

PRILOG A

Ovaj prilog sadržava video materijale validacije upravljačkog modela. Svi isječci snimljeni su u Laboratoriju za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje pri Sveučilištu u Zagrebu.

- Implementacija i validacija konstruktivnog algoritma i genetskog algoritma planiranja robotskog djelovanja: <http://youtu.be/gVMb8v12zU8>
- Validacija cjelokupnog upravljačkog modela: <http://youtu.be/83M4F7cJes4>
- Izbor objekata za koje se planira izgradnja: <http://youtu.be/RBSXNvyIbws>

ŽIVOTOPIS

Marko Švaco rođen je u Zagrebu 14.03.1986. godine. Osnovnu školu Dobriše Cesarića u Zagrebu pohađao je od 1992. do 2000. Godine 2004. završio je XV. gimnaziju u Zagrebu, matematičkog smjera. Studij na Fakultetu strojarstva i brodogradnje upisuje 2004. godine. Za izvrstan uspjeh na studiju nagrađen je nagradom „Davorin Bazjanac” 2005. i 2007. godine te medaljom fakulteta 2009. godine. Primao je stipendiju grada Zagreba za darovite studente u akademskoj godini 2007./2008. Diplomirao je 2009. godine, kada upisuje i Poslijediplomski studij smjera Robotika i automatizacija, a od veljače 2010. godine radi kao asistent te znanstveni novak pri Zavodu za robotiku i automatizaciju proizvodnih sustava na Katedri za projektiranje izradbenih i montažnih sustava Fakulteta strojarstva i brodogradnje, Sveučilištu u Zagrebu. Aktivno je sudjelovao u znanstvenim projektima: „IGRAMO – Improving grasping methods based on observations“, „ACRON – A new concept of Applied Cognitive Robotics in clinical Neuroscience“, znanstvenom projektu MZOS-a „Autonomna višeagentska automatska montaža“, te tehnologijskim projektima: „Automatska montaža dvoručnim robotom“ i „Primjena robota u neurokirurgiji – RONNA“ te strukturnom projektu „RONNA – Robotska neuronavigacija“. Sudjelovao je u pripremi strukturnog projekta „RONNA – Robotska neuronavigacija“. U sklopu stručnog usavršavanja boravio je mjesec dana na Sveučilištu KTH u Švedskoj 2011. godine. Aktivno se služi engleskim jezikom u govoru i pismu.

Objavio je tri rada u međunarodno recenziranim časopisima, te 12 radova na konferencijama s međunarodnom recenzijom.

Popis javno objavljenih radova:

[1] Bojan Jerbić, Gojko Nikolić, Darko Chudy, **Marko Švaco**, Bojan Šekoranja, „Robotic application in neurosurgery using intelligent visual and haptic interaction, International Journal of Simulation Modelling, vol. 14, no. 1, p. 71–84, 2015.

[2] **Marko Švaco**, Bojan Jerbić, Filip Šuligoj. ARTgrid: A Two-Level Learning Architecture Based on Adaptive Resonance Theory, Advances in Artificial Neural Systems, vol. 2014, p. 1–9, 2014.

- [3] **Marko Švaco**, Bojan Jerbić, Filip Šuligoj. Autonomous Robot Learning Model Based On Visual Interpretation Of Spatial Structures, Transactions of FAMENA, vol. 38, no. 4, p. 13-28, 2014
- [4] Bojan Jerbić, Filip Šuligoj, **Marko Švaco**, Bojan Šekoranja. Robot Assisted 3D Point Cloud Object Registration, Procedia Engineering, 2015, vol. 100, p. 847–852.
- [5] Bojan Šekoranja, Denis Bašić, **Marko Švaco**, Filip Šuligoj, Bojan Jerbić. Human-Robot Interaction Based on Use of Capacitive Sensors, Procedia Engineering, 2014, vol. 69, p. 464–468.
- [6] Filip Šuligoj, Bojan Šekoranja, **Marko Švaco**, Bojan Jerbić. Object Tracking with a Multiagent Robot System and a Stereo Vision Camera, Procedia Engineering, 2014, vol. 69, p. 968–973.
- [7] **Marko Švaco**, Bojan Šekoranja, Filip Šuligoj, Bojan Jerbić. Calibration of an Industrial Robot Using a Stereo Vision System, Procedia Engineering, 2014, vol. 69, p. 459–463.
- [8] **Marko Švaco**, Denis Bašić, Bojan Šekoranja, Bojan Jerbić. A capacitive sensor for human-robot interaction, Annals of DAAAM for 2012. & Proceedings of the 23rd International DAAAM Symposium, Vienna, 2012, p. 819–822.
- [9] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. Industrial Robotic System with Adaptive Control, Procedia Computer Science, 2012, vol. 12, p. 164–169.
- [10] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. Service Oriented Robotic Assembly Architecture, Proceedings of 13th International Scientific Conference on Production Engineering, Biograd, Croatia, 2011, p. 121–124.
- [11] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. Autonomous Planning Framework for Distributed Multiagent Robotic Systems, Technological Innovation for Sustainability, Springer, 2011, p. 147–154.
- [12] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. A multiagent framework for industrial robotic applications, Procedia Computer Science, 2011, vol. 6, p. 291–296.

[13] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. Multiagent Robotic Collaborative Framework, Annals of DAAAM for 2011 & proceedings of the 22nd International DAAAM symposium Intelligent manufacturing & Automation, Vienna, 2011, p. 731–732.

[14] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. Heuristic action planning algorithm for distributed multiagent systems, Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Zadar, Croatia, 2010, p. 1067–1069.

[15] **Marko Švaco**, Bojan Šekoranja, Bojan Jerbić. A Multiagent Approach for Development of a Flexible and Adaptive Robotic Assembly Work Cell, The 3rd International Conference on Computational Intelligence and Industrial Application (PACIIA 2010), 2010, p. 64–67.

CURRICULUM VITAE

Marko Švaco was born on March 14th 1986 in Zagreb. He finished elementary school “Dobriša Cesarić” in 2000 and high school “XV. gimnazija” in 2004. He enrolled the Faculty of Mechanical Engineering and Naval Architecture (FSB), University of Zagreb in 2004. He was awarded by the “Davorin Bazjanac” award in 2005 and 2007 and the "Medal of the Faculty of Mechanical Engineering and Naval Architecture" in 2009 for the best students. He received the scholarship of city of Zagreb in academic year 2007/2008 for gifted students. In 2009 he graduated from the Faculty of Mechanical Engineering and Naval Architecture (FSB) and enrolled the postgraduate program of Robotics and automation. From February 2010 he works as a research and teaching assistant at the Department of Robotics and Production System Automation, Chair for Manufacturing and Assembly System Planning, at FSB. He actively participated in research projects: „IGRAMO – Improving grasping methods based on observations“, „ACRON - A new concept of Applied Cognitive Robotics in clinical Neuroscience“, „Autonomous multiagent automatic assembly“ ,and technological projects: „Automatic dual-arm robotic assembly“, „Application of robots in neurosurgery“ and „RONNA –robotic neuronavigation“. In January-February 2011 he was on a research visit at KTH Stockholm, Sweden. He is fluent in written and spoken English.

He published three scientific journal papers and 12 scientific conference papers with international peer-review.

