

საქართველოს ტექნიკური უნივერსიტეტი

პეტრე პეტაშვილი

კოლექტიური ინტელექტის მეთოდების გამოყენება მულტი-რობოტული
სისტემების მართვაში

წარმოდგენილია დოქტორის აკადემიური ხარისხის მოსაპოვებლად

სადოქტორო პროგრამა „ინფორმატიკა“, შიფრი 0401

საქართველოს ტექნიკური უნივერსიტეტი

თბილისი, 0175, საქართველო

ივლისი, 2016 წელი

საავტორო უფლება © 2016 წელი, პეტრე პეტაშვილი

თბილისი

2016 წელი

სამუშაო შესრულებულია საქართველოს ტექნიკური უნივერსიტეტში
ინფორმატიკისა და მართვის სისტემების ფაკულტეტი
მართვის ავტომატიზებული სისტემების დეპარტამენტი

ხელმძღვანელი: პროფ. ბადრი მეფარიშვილი

რეცენზენტები: -----

დაცვა შედგება ----- წლის ”-----” -----, ----- საათზე
საქართველოს ტექნიკური უნივერსიტეტის -----
----- ფაკულტეტის სადისერტაციო საბჭოს
კოლეგიის
სხდომაზე, კორპუსი -----, აუდიტორია -----
მისამართი: 0175, თბილისი, კოსტავას 77.

დისერტაციის გაცნობა შეიძლება სტუ-ს ბიბლიოთეკაში,
ხოლო ავტორეფერატისა - ფაკულტეტის ვებგვერდზე

სადისერტაციო საბჭოს მდივანი პროფ. თინათინ კაიშაური

საქართველოს ტექნიკური უნივერსიტეტი

ინფორმატიკისა და მართვის სისტემების ფაკულტეტი

ჩვენ, ქვემოთ ხელისმომწერნი ვადასტურებთ, რომ გავეცანით პეტაშვილი პეტრეს მიერ შესრულებულ სადისერტაციო ნაშრომს დასახელებით: „კოლექტიური ინტელექტის მეთოდების გამოყენება მულტი-რობოტული სისტემების მართვაში“ და ვაძლევთ რეკომენდაციას საქართველოს ტექნიკური უნივერსიტეტის ინფორმატიკისა და მართვის სისტემების ფაკულტეტის სადისერტაციო საბჭოში მის განხილვას დოქტორის აკადემიური ხარისხის მოსაპოვებლად.

თარიღი

ხელმძღვანელი: პროფ. ბადრი მეფარიშვილი

რეცენზენტი:

რეცენზენტი:

საქართველოს ტექნიკური უნივერსიტეტი

2016

ავტორი: პეტაშვილი პეტრე

დასახელება: „კოლექტიური ინტელექტის მეთოდების
გამოყენება მულტი-რობოტული სისტემების მართვაში“;

ფაკულტეტი : ინფორმატიკისა და მართვის სისტემების

ხარისხი: დოქტორი

სხდომა ჩატარდა:

ინდივიდუალური პიროვნებების ან ინსტიტუტების მიერ
ზემომოყვანილი დასახელების ნაშრომის გაცნობის მიზნით მოთხოვნის
შემთხვევაში მისი არაკომერციული მიზნებით კოპირებისა და გავრცელების
უფლება მინიჭებული აქვს საქართველოს ტექნიკურ უნივერსიტეტს.

ავტორის ხელმოწერა

ავტორი ინარჩუნებს დანარჩენ საგამომცემლო უფლებებს და არც
მთლიანი ნაშრომის და არც მისი ცალკეული კომპონენტების გადაბეჭდვა ან
სხვა რაიმე მეთოდით რეპროდუქცია დაუშვებელია ავტორის წერილობითი
ნებართვის გარეშე.

ავტორი ირწმუნება, რომ ნაშრომში გამოყენებული საავტორო
უფლებებით დაცული მასალებზე მიღებულია შესაბამისი ნებართვა (გარდა
ის მცირე ზომის ციტატებისა, რომლებიც მოითხოვენ მხოლოდ სპეციფიურ
მიმართებას ლიტერატურის ციტირებაში, როგორც ეს მიღებულია
სამეცნიერო ნაშრომების შესრულებისას) და ყველა მათგანზე იღებს
პასუხისმგებლობას.

რეზიუმე

ნაშრომის განხილვის საგანს წარმოადგენს ხელოვნური ინტელექტის ერთ-ერთი უახლესი მიმართულების - კოლექტიური ინტელექტის მეთოდების შემუშავება და მათი პრაქტიკული გამოყენების ასპექტები, კერძოდ, ტექნოგენური კატასტროფებით დაბინძურებული გარემოს მონიტორინგის თვალსაზრისით, რაც არის ძალიან რთული და ამასთანავე მნიშვნელოვანი პრობლემა. ეს უკანასკნელი განპირობებულია რადიაციით გარემოს ძლიერი დაბინძურებით გამოწვეული შედეგებით, განსაკუთრებით ატომური ენერგეტიკის და სამხედრო სფეროებში. რამაც წარმოიშვა მზარდი მოთხოვნა ისეთ საშუალებებზე, რომელიც უზრუნველყოფს დაბინძურების წყაროების ავტომატიზებულ მონიტორინგს, განსაზღვრავს რადიაქტიული წყაროების გამოსხივების, მომწამვლელი გაზების და ა.შ. ადამიანისთვის უსაფრთხო ზღვრებს. დღესდღეობით, არსებობს საჭიროება მაღალი სიზუსტისა და მგრძნობელობის მქონე მოწყობილობებზე დაბინძურების წყაროებისა და საშიში ზონების აღმოჩენა-გაზომვის თვალსაზრისით.

ადაპტური ოპტიმიზაციის აუცილებლობა გამომდინარეობს გარემოს არასტაციონალურობიდან, როდესაც მიზნობრივი ფუნქციის განაწილების ლანდშაფტი დინამიურად იცვლება. დასმული ამოცანის გადასაჭრელად საჭიროა მონიტორინგი განხორციელდეს განაწილებული მულტი-რობოტული ქსელიდან გარემოზე ინფორმაციის უწყვეტად შეკრების გზით. მულტი-რობოტული სისტემა არის ავტონომიური მოწყობილობების (რობოტების) რეკონფიგურირებადი სტრუქტურის ქსელი, რომელსაც შეუძლია მონიტორინგი გაუწიოს გარემო პირობების ცვალებადობას.

კვლევა ეფუძნება ხელოვნური ინტელექტის მეთოდების ფართო სპექტრს, ისეთ თანამედროვე მეთოდებს, როგორიცაა: ევოლუციური ალგორითმები და კოლექტიური ინტელექტი, კოლექტიური ქცევის მათემატიკური მოდელირება და მრავალმიზნობრივი მართვა ენტროპიაზე დაფუძნებული მეტრიკის ბაზაზე.

ევოლუციური ალგორითმები (EAs) წარმოადგენენ ბუნებრივ პოპულაციასა და მის ევოლუციაზე დაფუძნებულ მოდელებს, რომლებიც პოპულაციის ინდივიდების მრავალჯერადი ევოლუციის (წინა თაობის პოპულაციიდან ახალი თაობის გენერირება ევოლუციური ოპერატორებისა და ინდივიდების ფიტნეს ფუნქციის დახმარებით) გზით მიემართებიან სასურველი მიზნისაკენ. ევოლუციური ალგორითმების განსაკუთრებულობად ის ითვლება, რომ ამონახსნთა სიმრავლის წარმოდგენის ნებისმიერი სქემის დროს ის იძლევა არაჩვეულებრივ და საინტერესო შედეგს. ამდენად, ევოლუციური ალგორითმების შემდგომი განვითარება აქტუალური და პერსპექტიულია რეალურ ცხოვრებაში არსებულ ამოცანათა საიმედო და ეფექტური გადაჭრის თვალსაზრისით.

ხელოვნური ინტელექტის ერთ-ერთ უახლეს მიმართულებას წარმოადგენს მულტი-აგენტური სისტემები (მას), რომელიც განიხილება როგორც გადაწყვეტილების მიღების ეფექტური საშუალება ცალკეული თუ

ერთობლივი მიზნების მისაღწევად. ნაშრომი დაფუძნებულია მულტი-რობოტულ მიდგომაზე, რომელიც წარმოადგენს მას-ის კონკრეტულ შემთხვევას, რომლის აგენტები მობილური რობოტები არიან.

კვლევის მთავარ სიახლეს წარმოადგენს სოციალური სისტემების კოლექტიურ ქცევებთან დაკავშირებული კონცეფციების შემოტანა მულტი-აგენტურ სისტემებში. ჩვენს მიერ შემუშავებული მიდგომები შეიძლება განვიხილოთ როგორც ახალი საინჟინრო გამოთვლითი პარადიგმა, რომელიც ემყარება დინამიური სისტემებისათვის დამახასიათებელ ენტროპიისა და სინერგიის ცნებებს. თუ განვიხილავთ რთულ სისტემას, როგორც ინტერაქტიულ, მულტი-აგენტურ ჰეტეროგენულ ქაოტურ სისტემას მრავალგანზომილებიანი, რთული იერარქიული სტრუქტურით, მაშინ მისი მოდელირება წარმოადგენს ურთულეს პრობლემას. ეს განპირობებულია ადამიანის, როგორც არაწრფივი და არამკაფიო ფაქტორის არსებობით, რაც შესაბამისად დაკავშირებულია ქცევის თავისუფლების მეტად მაღალ ხარისხთან. მოცემულ კონტექსტში გროვის მოდელის აგება და მისი ოპტიმიზაცია შესაძლებელია ენტროპიის მინიმიზაციის კრიტერიუმის მიხედვით.

ჩვენი მიზანია, ხელოვნური ინტელექტის მეთოდებზე, განსაკუთრებით, გროვის ქცევის მოდელირებისა და კოლექტიური ინტელექტის კონცეფციაზე დაფუძნებული ახალი მიდგომების შემუშავება - ფიზიკური თუ გარემო რისკების მონიტორინგისა და შეფასების, რობოტების ჯგუფის ადაპტური მართვის (მათ შორის, უპილოტო მიწისზედა ტრანსპორტის სისტემის), განუსაზღვრელ პირობებში ნავიგაციის, საგანგებო სიტუაციებში სამაშველო თუ სხვა ოპერაციათა მხარდაჭერის, ასევე აგრესიულ თუ მტრულ გარემოში სამხედრო ამოცანების შესრულების თვალსაზრისით.

ჩვენ შევიმუშავეთ მულტი-რობოტული სისტემის მართვის ახლებური მიდგომა. ამისთვის გამოვიყენეთ კოლექტიური ინტელექტის მეთოდები, კერძოდ, ნაწილაკების გროვის ოპტიმიზაციის მეთოდი (PSO). ჩვენი მიდგომა, ორიგინალი მიდგომისგან განსხვავებით, გლობალური ოპტიმუმთან ერთად პოულობს ლოკალურ ოპტიმუმებსაც.

ალგორითმი არის იტერაციული და შედგება რამდენიმე ბიჯისგან: ინიციალიზაცია, კლასტერიზაცია, განახლება, კლასტერების შეერთება (არააუცილებელი ბიჯი) და დასრულების კრიტერიუმის შემოწმება.

სივრცულად განაწილებული მულტი-რობოტული სისტემის ქცევა რობოტების ინდივიდუალური, ავტონომიური ქმედებებით არის განპირობებული. საძიებო არეში ნაწილაკების მოძრაობა განისაზღვრება მათი საუკეთესო პოზიციით, რომელიც ამავდროულად მთლიანი სისტემის საუკეთესო პოზიციასაც განაპირობებს.

მიდგომაში შემოტანილია ლიდერი და აუტოსაიდერი რობოტების ცნებები და მათი შერჩევის კრიტერიუმები. აღსანიშნავია, რომ ალგორითმის მუშაობის პროცესში შესაძლებელია რობოტების როლების შეცვლა, ანუ ლიდერი რობოტი შეიძლება გახდეს აუტოსაიდერი და პირიქით.

დიდი ინფორმაციის სწრაფად დამუშავების მიზნით მოვახდინეთ საძიებო არის კლასტერიზაცია. კლასტერიზაცია წარმოადგენს დიდ

მონაცემთა სწრაფად მოპოვებისა და დამუშავების მძლავრ მექანიზმს. კლასტერიზაციას ვახდენთ ლიდერი რობოტებისა და K-Means ალგორითმის გამოყენებით.

კლასიკური ოპტიმიზაციის ალგორითმებისგან განსხვავებით, ჩვენს ალგორითმში დინამიურად გამოითვლება ალგორითმის დასრულების კრიტერიუმები. იგი დამოკიდებულია გროვის მდგომარეობასა და მოთხოვნილ სიზუსტეზე. პარამეტრების დინამიურად შერჩევა უმეტეს შემთხვევაში უფრო ეფექტურს ხდის ჩვენს ალგორითმს, რადგან ორი ურთიერთ საწინააღმდეგო ცნებიდან - დრო და სიზუსტე, საჭიროა დაბალანსება და კომპრომისზე წასვლა.

ჩვენ გვჯერა, რომ ჩვენი ალგორითმი შეძლებს სწრაფად მოძებნოს ოპტიმალური შედეგი, გარემო პირობების მუდმივად ცვლილების დროსაც კი.

ჩატარებული გამოთვლების შედეგებით დადასტურდა, რომ ჩვენს მიერ შემუშავებული ალგორითმის გამოყენების შედეგად, მნიშვნელოვნად შემცირდა გამოთვლითი ოპერაციების რაოდენობა, რაც საბოლოო ჯამში, ზოგავს დროსა და მანქანურ რესურსებს. ექსპერიმენტების შედეგების ანალიზმა ცხადყო შემუშავებული ალგორითმის ეფექტურობა, მაღალი სიზუსტე და კრებადობა კლასიკურ მეთოდებთან შედარებით. სწორედ ეს წარმოადგენს ჩვენი კვლევის და დისერტაციის მიღწევას.

Abstract

The subject of the present paper is one of modern directions of the artificial intelligence – the development of collective intelligence methods and their application for the monitoring of the environmental pollution caused by technogenic disasters, what is important and at the same time, very complex issue. Heavy effects of radiation pollution, especially in the nuclear power and military sectors, have created an increasing demand on instruments that can facilitate to the identification and automatic monitoring of pollution sources, and the establishment of safe areas in case of radiation, chemical and other hazards. Nowadays, the high precision and sensitive equipment that enable detection and monitoring of pollution sources and hazardous zones is demanded.

Necessity of Adaptive Optimization is caused by instability of the environment itself, where change in the distribution area of the Objective Function is dynamic. A Multi-Robot network should be used for a continuous environmental monitoring in order to solve the above defined problem. The Multi-Robot System can be interpreted as a network of autonomous, reconfigurable objects (robots), which are capable to monitor environmental changes.

The present research is based on a wide range of modern methods of the Artificial Intelligence (AI) such as: Evolutionary Algorithms (EAs) and Collective Intelligence, Mathematical Modelling of Collective Behaviour and entropy metrics in multi-objective management.

Evolutionary Algorithms represent models of a natural population and its evolution, where population inhabitants, through their continuous evolution (generation of new population from their predecessors with help of evolutionary operators and the fitness function of individuals), are striving to a desired objective. The distinctive feature of Evolutionary Algorithms is that they can achieve optimal results in any combination of solution candidates. Therefore, future development of Evolutionary Algorithms looks to be promising for effective solution of variety of practical issues.

The Multi-Agent System (MAS) is one of the modern approaches of the Artificial Intelligence, and it can be considered as an effective decision-making tool in the achievement of individual and combined (complex) goals. The paper is based on the Multi-Robot approach being a single case of MAS, which uses mobile robots as its agents.

The main novelty of the research is the introduction of ‘Collective Behaviour of Social Systems’-based concepts in the Multi-Agent Systems approach. Our approaches can be considered as a new computing paradigm, which is based on the Entropy and Synergy concepts relevant to the Dynamic Systems. If we consider a complex system as an interactive, multi-agent heterogeneous chaotic system with multidimensional, compound hierarchical structure, then, creation of the relevant model becomes a complicated issue. The modelling of a complex system, which is interactive, multi-agent heterogeneous, chaotic, multidimensional, with compound hierarchical structure, is an overwhelming task. This is due to the human being, as

a non-linear and uncertain factor, which is characterised with a high degree of freedom of behaviour. In this context, the swarm model can be build and optimized through the minimization of the entropy.

Our goal is the elaboration of new approaches based on artificial intelligence methods - especially swarm behaviour modelling and collective intelligence concept - to be applied in environmental monitoring and risks assessment, adaptive management of the robot swarm (pilotless ground transport system among others), navigation in uncertain conditions, support of emergency response operations and military operations in hostile environment. We have elaborated an innovative approach towards the Multi-Robot system management. The approach is based on the Collective Intelligence, and in particular the Particle Swarm Optimization (PSO) method. Versus to the original algorithm, which can cope only with the global optimum, the proposed one is capable to find both global and local optimums.

The algorithm is iterative and consists of several steps: initialization, clustering, updating, clusters merging (optional) and checking against algorithm completion criteria.

The behaviour of spatial multi-robot system is determined by autonomous actions of individual robots. The movement of elements in the search area is determined by their best possible positions, which on its turn determines the best possible position of the whole system.

For optimal solution of the problem we have introduced the concept of 'Leader' and 'Outsider' robots, and their selection criteria. It should be noted that robots can change their roles during the algorithm processing, i.e. a 'Leader' robot can become 'Outsider' and vice versa.

Further, search area has been clustered to speed-up large-scale data processing. Clustering is acknowledged powerful tool for prompt data mining and processing. We used 'Leader' robots and K-Means algorithms for clustering.

Different from traditional one, our algorithm dynamically computes completion criteria. They depend on a state of the swarm and requested accuracy. In the most cases, the dynamic selection of completion criteria makes our algorithm much more effective, as the logic is based on the balancing of two conflicting parameters such are time and accuracy, and finding of acceptable trade-off in this process.

We do believe, that our algorithm is capable to promptly find an optimal solution even in continuously changing environmental conditions.

Our computations proved that the proposed algorithm substantially reduces the number processing operations, and thus enables to save time and resources. The analysis of the outcomes demonstrated that our algorithm is more effective and accurate than conventional methods. This is what we consider to be the main achievement of our research and the work.

შინაარსი

შესავალი.....	16
თავი 1 ლიტერატურის მიმოხილვა	19
1.1 კოლექტიური ინტელექტი.....	19
1.2 მულტი-რობოტული სისტემები.....	21
თავი 2 მულტი-რობოტული სისტემები.....	23
2.1 საგნობრივი სფერო. ამოცანის დასმა. კვლევის აქტუალურობა და მნიშვნელობა	23
2.1.1 უპილოტო მიწისზედა ტრანსპორტის ნავიგაციის მოდელირების და მართვის ამოცანა განუსაზღვრელ პირობებში.....	24
2.1.2 უპილოტო მიწისზედა ტრანსპორტის ნავიგაციის მოდელირების და მართვის ამოცანა გარემოს მონიტორინგის მიზნით	25
2.1.3 კვლევის მიზანი.....	26
2.1.4 კვლევის სიახლე	27
2.2 მულტი-რობოტული სისტემების მიმოხილვა	27
2.2.1 მულტი-რობოტული სისტემის არქიტექტურა.....	29
2.2.2 მულტი-რობოტული სისტემის მართვის სტრატეგიები.....	34
2.2.3 მულტი-აგენტური სისტემის პროგრამული არქიტექტურა	35
თავი 3 მულტი-რობოტული სისტემების თეორიული ასპექტები.....	39
3.1 ოპტიმიზაციის ამოცანის ფორმალიზაცია.....	39
3.1.1 ოპტიმიზაციის ალგორითმების კლასიფიკაცია.....	40
3.1.2 ერთი და მრავალმიზნობრივი ოპტიმიზაცია	43
3.1.3 პარეტოს ოპტიმიზაცია (Pareto Optimization).....	46
3.2 სოციალური ენტროპია და სინერგია	48
3.2.1 ენტროპია მულტი-რობოტულ სისტემებში	48
3.2.2 სინერგია.....	50
3.3 კლასტერიზაციის მიმოხილვა	52
3.3.1 ვორონოის დიაგრამა (Voronoi Diagram).....	56
3.3.2 კოჰონენის თვითორგანიზებადი რუკა (SOM)	61
3.3.3 კლასტერიზაცია K-Means - ის მეთოდით.....	66
თავი 4 ევოლუციური ალგორითმების მიმოხილვა და ანალიზი.....	72
4.1 გენეტიკური ალგორითმი (GA).....	73
4.1.1 გენეტიკური ალგორითმის კლასიკური მოდელი	74
4.2 გენეტიკური პროგრამირება (GP).....	79
4.3 ხელოვნური ნეირონული ქსელები (ANN)	83
4.3.1 ნეირონის მათემატიკური მოდელი	84
4.3.2 ხელოვნური ნეირონული ქსელის სწავლება	85
4.3.3 ხელოვნური ნეირონული ქსელის მოდელები	87
4.4 კოლექტიური ინტელექტი.....	88

4.5	ნაწილაკების გროვის ოპტიმიზაცია (PSO)	90
4.5.1	PSO-ს პარამეტრები	93
4.6	ჭიანჭველების კოლონიის ოპტიმიზაცია (ACO)	96
4.7	ამოწვის მეთოდი (ANNEALING)	98
4.8	იმპერიალისტური მეთოდი (ICA)	100
4.8.1	მეთოდის აღწერა	101
თავი 5	მულტი-რობოტული სისტემების მართვის ახლებური მიდგომა ...	106
5.1	ალგორითმის აღწერა	106
5.1.1	ინიციალიზაცია	107
5.1.2	კლასტერიზაცია	108
5.1.3	განახლება.....	112
5.1.4	კლასტერების გაერთიანება	113
5.1.5	ალგორითმის შეწყვეტა	114
5.2	ახლებური მიდგომის უპირატესობები	115
5.3	ექსპერიმენტის დაგეგმვა და რეალიზაცია	117
თავი 6	შედეგები და მათი განსჯა	125
თავი 7	დასკვნა.....	127
	გამოყენებული ლიტერატურა.....	129

ცხრილების ნუსხა

ცხრილი 1. რადიაციის წყაროები	118
ცხრილი 2. 1-20 ცდის დეტალური შედეგები	122
ცხრილი 3. ცდების ჯამური შედეგები	124

ნახაზების ნუსხა

სურათი 1. UGVs მართვა განუსაზღვრელ პირობებში.....	25
სურათი 2. UGVs მართვა გარემოს მონიტორინგის მიზნით	26
სურათი 3. მარტივი აგენტის სტრუქტურა	28
სურათი 4. მულტი-რობოტული სისტემა	29
სურათი 5. მულტი-რობოტული სისტემის თვითორგანიზება	31
სურათი 6. ცენტრალიზებული და დეცენტრალიზებული მართვა.....	35
სურათი 7. აგენტის შინაგანი არქიტექტურის მაგალითი.....	37
სურათი 8. ოპტიმიზაციის ალგორითმების კლასიფიკაცია.....	42
სურათი 9. გლობალური და ლოკალური ოპტიმუმები	45
სურათი 10. ოპტიმიზაცია პარეტოს ზღვრის მიდგომით.....	47
სურათი 11. კლასტერიზაციის ევოლუციური პროცესი	52
სურათი 12. ექსკლუზიური და იერარქიული კლასტერიზაცია	53
სურათი 13. კლასტერიზაციის ტიპი - „კარგად დაყოფილი“	54
სურათი 14. კლასტერიზაციის ტიპი - „პროტოტიპზე-დაფუძნებული“	54
სურათი 15. კლასტერიზაციის ტიპი - „სიმკვრივეზე-დაფუძნებული“	55
სურათი 16. კლასტერიზაციის ტიპი - „კონცეპტუალური“	55
სურათი 17. დეკარტის სივრცის დეკომპოზიცია	56
სურათი 18. ვორონოის დიაგრამა ორი წერტილით.....	58
სურათი 19. ვორონოის დიაგრამა სამი წერტილით.....	59
სურათი 20. მრავალი წერტილით აგებული ვორონოის რეგიონი.....	59
სურათი 21. ვორონოის დიაგრამა.....	60
სურათი 22. ვორონოის დიაგრამის დუალიზმი, დელონეს ტრიანგულაცია	61
სურათი 23. კოჰონენის ქსელი	63
სურათი 24. კოჰონენის რუკა.....	64

სურათი 25. კოჰონენის რუკა.....	65
სურათი 26. კლასტერი	66
სურათი 27. კლასტერიზაციის ამოცანა, საწყისი მდგომარეობა.....	67
სურათი 28. კლასტერიზაციის ამოცანა, შუალედური მდგომარეობა.....	68
სურათი 29. კლასტერიზაციის ამოცანა, საბოლოო მდგომარეობა.....	69
სურათი 30: ევოლუციური ალგორითმების ქრონოლოგია.....	72
სურათი 31. ევოლუციური ალგორითმების კლასიფიკაცია	73
სურათი 32. GA-ში ორობითი კოდირება	74
სურათი 33. GA-ში ალფაბეტური და რიცხვითი კოდირება	75
სურათი 34. GA-ში ხისებრი სტრუქტურის კოდირება.....	75
სურათი 35. გენეტიკური ალგორითმის მოქმედების ძირითადი ციკლი	75
სურათი 36. Ga-ში ქრომოსომების შეჯვარება	78
სურათი 37. Ga-ში ქრომოსომების მუტაცია	78
სურათი 38. გენეტიკური პროგრამირების ალგორითმის ბლოკ-სქემა	80
სურათი 39. გამოსახულების წარმოდგენა ხისებრი სტრუქტურით.....	81
სურათი 40. ინიციალიზაცია „სრული“ (Full) მეთოდით	81
სურათი 41. ინიციალიზაცია „გაზრდის“ (Grow) მეთოდით.....	82
სურათი 42. GP-ში შეჯვარება.....	82
სურათი 43. GP-ში მუტაცია.....	83
სურათი 44. ნეირონის მათემატიკური მოდელი	84
სურათი 45. კავშირების გლობალურობა ხელოვნურ ნეირონულ ქსელებში	85
სურათი 46. ქსელის სწავლება როგორც ოპტიმიზაციის ამოცანა	86
სურათი 47. ANN-ის მოდელის ტიპები.....	87
სურათი 48. PSO-ს ტოპოლოგია.....	91
სურათი 49. PSO-ს ტოპოლოგია.....	92
სურათი 50. ჭიანჭველების კოლონიის გზა საკვების მოსაპოვებლად.....	97
სურათი 51. ამოწვის იმიტაციის ალგორითმი	100
სურათი 52. იმპერიებისა და კოლონიების განაწილება.....	102
სურათი 53. კოლონიის მოძრაობა იმპერიალისტის მიმართ	103
სურათი 54. კოლონიის და იმპერიალისტის როლების შეცვლა	103
სურათი 55. იმპერიების პაექრობა კოლონიების დაპყრობის მიზნით.....	104
სურათი 56. ICA – ს ბლოკ-სქემა	105

სურათი 57. სენსორული რობოტების ჯგუფის მართვის ბლოკ-სქემა	106
სურათი 58. შემთხვევითად განაწილებული ნაწილაკები რადიაციის წყაროებთან	108
სურათი 59. ლიდერების არჩევა	109
სურათი 60. კლასტერიზაცია	111
სურათი 61. გროვის მდგომარეობა k -ურ ბიჯზე	113
სურათი 62. შეჯგუფების ცენტრალური წერტილი	114
სურათი 63. საბოლოო მდგომარეობა	115
სურათი 64. ალგორითმში გამოყენებული ტოპოლოგიის სქემა	116
სურათი 65. სამიეზო არის ლოკალური და გლობალური ოპტიმუმები	117
სურათი 66. ინიციალიზაცია	120
სურათი 67. კლასტერიზაცია	121
სურათი 68. ალგორითმის დასრულება	122
სურათი 69. კლასტერების გაერთიანების დიაგრამა	125
სურათი 70. გროვის ბიჯების რაოდენობის დიაგრამა	126
სურათი 71. გროვის ფიტნეს-ფუნქციის სიზუსტის დიაგრამა	126

დისერტაციაში გამოყენებული აბრევიატურები

GIS – გეოგრაფიული ინფორმაციული სისტემები (Geographic Information System);

AI – ხელოვნური ინტელექტი (Artificial intelligence);

EA – ევოლუციური ალგორითმი (Evolutionary algorithm);

GA – გენეტიკური ალგორითმი (Genetic algorithm);

GP – გენეტიკური პროგრამირება (Genetic programming);

ANN – ხელოვნური ნეირონული ქსელი (Artificial neural network);

MLP – მრავალდონიანი „პირდაპირი კავშირის“ ნეირონული ქსელი (Multi layer perceptron);

SI – გროვის ინტელექტი (Swarm intelligence);

MAS – მულტი-აგენტური სისტემა (Multi-agent system);

ACO – ჭიანჭველების კოლონიის ოპტიმიზაცია (Ant colony optimization);

ABC – მართვა ჭიანჭველის გამოყენებით (Ant Based Control);

PSO – ნაწილაკების გროვის ოპტიმიზაცია (Particle Swarm Optimization);

SOM – თვითორგანიზებადი რუკა (Self-Organizing Maps)

UGVs – უპილოტო მიწისზედა სატრანსპორტო სისტემა (Unmanned Ground Vehicle systems)

შესავალი

თანამედროვე ინფორმაციული ტექნოლოგიების განვითარებამ მნიშვნელოვნად გააფართოვა სხვადასხვა სფეროში ტექნოლოგიური კვლევის ჰორიზონტი. უკვე დიდი ხანია კომპიუტერი აღარ ისაზღვრება მხოლოდ ექსპერიმენტული მონაცემების დამუშავებისა და ანალიზის ფუნქციით. ძვირადღირებული და ძნელად რეალიზებადი ფუნდამენტური თეორიული თუ ექსპერიმენტული კვლევების ნაცვლად სულ უფრო მეტად გამოიყენება მანქანური მოდელირების პარადიგმა.

დღესდღეობით არსებული რეალური პრობლემების (ამოცანების) უმრავლესობა იყოფა ამოცანების ორ ტიპად, რომელთაც ახასიათებთ მრავალი კონფლიქტური მიზნობრიობა, ან ამონახსნთა საკმაოდ რთული საძიებო არე. მეორეს მხრივ, ასეთ ამოცანებში ერთი ოპტიმალური ამონახსნის მაგივრად წარმოდგენილია ამონახსნთა მთელი სიმრავლე (Pareto-optimal). ამონახსნთა ამ სიმრავლიდან, დამატებითი ცნობების გარეშე, შეუძლებელია შევავასოთ (ამოვირჩიოთ) ერთი ამონახსნი როგორც საუკეთესო. ამასთანავე, ეს სიმრავლე შეიძლება იყოს იმდენად დიდი, რომ ზუსტი მეთოდებით მისი ამოხსნა შეუძლებელი იყოს. ამიტომ საჭიროა ეფექტური ოპტიმიზაციის სტრატეგიები, რომელთაც შეუძლიათ ორივე სახის ამოცანებთან გამკლავება. სწორედ ასეთ მეთოდებს წარმოადგენენ ევოლუციური ალგორითმები.

ევოლუციური ალგორითმები (EAs) წარმოადგენენ ბუნებრივ პოპულაციასა და მის ევოლუციაზე დაფუძნებულ მოდელებს, რომლებიც პოპულაციის ინდივიდების მრავალჯერადი ევოლუციის (წინა თაობის პოპულაციიდან ახალი თაობის გენერირება ევოლუციური ოპერატორებისა და ინდივიდების ე.წ. ფიტნეს ფუნქციის მიხედვით) გზით მიემართებიან სასურველი მიზნისაკენ. ევოლუციური ალგორითმების განსაკუთრებულ მხარედ ითვლება ის, რომ ამონახსნთა სიმრავლის წარმოდგენის ნებისმიერი სქემის დროს იძლევა არაჩვეულებრივ და საინტერესო შედეგს. ამდენად,

ევოლუციური ალგორითმების შემდგომი განვითარება აქტუალური და პერსპექტიულია რეალურ ცხოვრებაში არსებულ ამოცანათა საიმედო და ეფექტური გადაჭრის თვალსაზრისით.

ხელოვნური ინტელექტის ერთ-ერთ უახლეს მიმართულებას წარმოადგენს *მულტი-აგენტური სისტემები (მას)*, რომელიც განიხილება როგორც გადაწყვეტილების მიღების ეფექტური საშუალება ცალკეული თუ ერთობლივი მიზნების მისაღწევად. ნაშრომი დაფუძნებულია მულტი-რობოტულ მიდგომაზე, რომელიც წარმოადგენს მას-ის კონკრეტულ შემთხვევას, რომლის აგენტები მობილური რობოტები არიან. ჩვენი მიზანია მულტი-რობოტული სისტემის მართვის ოპტიმალური ალგორითმის შემუშავება.

სადისერტაციო ნაშრომი შედგება შვიდი თავისგან:

პირველ თავში მოცემულია ლიტერატურის მიმოხილვა - არსებული წარმოდგენების და მოდელების განხილვა და ანალიზი, მკვლევარების ნაშრომები, მოსაზრებები და შეფასებები.

მეორე თავში აღწერილია საგნობრივი სფერო, კვლევის აქტუალურობა და მნიშვნელობა, კვლევის მიზანი და სიახლე. ასევე, აღწერილია მულტი-რობოტული სისტემა და მისი ოპტიმიზაციის ამოცანები, როგორც ჩვენი კვლევის მთავარი ინსტრუმენტი.

მესამე და მეოთხე თავებში განხილულია ნაშრომის თეორიული ასპექტები. კერძოდ, მოცემულია ოპტიმიზაციის ამოცანის ფორმალიზაცია, შემოტანილია ენტროპიის და სინერგიის ცნებები, როგორც სისტემის ქაოტურობის ზომა და განხილულია მონაცემთა კლასტერიზაციის ზოგადი და ჩვენს მიერ გამოყენებული მიდგომები. ასევე, განხილულია ევოლუციური ალგორითმები, მათი ისტორიული განვითარება, აგებისა და გამოყენების ხერხები. აღნიშნულ თავებში განსაკუთრებული ყურადღება ეთმობა ნაწილაკების გროვის ოპტიმიზაციის (PSO) მეთოდს, რადგან სწორედ ამ მიდგომას ვიყენებთ მულტი-რობოტული სისტემის მართვაში.

მეხუთე და მეექვსე თავებში განხილულია მულტი-რობოტული სისტემის მართვის ახლებური, ჩვენს მიერ შემუშავებული მიდგომა. დეტალურად არის აღწერილი მართვის ალგორითმის თითოეული ბიჯი, ამ ბიჯების შესრულების პირობები და ალგორითმის დასრულების კრიტერიუმები. ასევე, მოცემულია ჩვენი მიდგომის და კლასიკურ მეთოდების შედარებითი ანალიზი. აღწერილია ჩვენს მიერ ჩატარებული ექსპერიმენტი და წარმოდგენილია მიღებული შედეგების შეფასება და ანალიზი.

მეშვიდე თავი მოიცავს ნაშრომის დასკვნას.

ნაშრომის ბოლოს თანდართულია გამოყენებული ლიტერატურის სია.

თავი 1 ლიტერატურის მიმოხილვა

კვლევა შეეხება ტექნოგენური კატასტროფებით დაბინძურებული გარემოს მონიტორინგს, რაც წარმოადგენს უმნიშვნელოვანეს და ამასთან ძალზედ რთულ ამოცანას. იგი მოტივირებულია 2011 წელს ფოკუსიმაში მომხდარი რადიაციული კატასტროფით. ჩვენი მიზანია კოლექტიური ინტელექტისა და მულტი-აგენტური სისტემების გამოყენებით უპილოტო მიწისზედა ტრანსპორტის ქცევის მოდელირება და შემუავებული მიდგომების დანერგვა განუსაზღვრელ პირობებში ნავიგაციის და საგანგებო სიტუაციებში სამაშველო თუ სხვა ოპერაციათა მხარდაჭერისთვის.

1.1 კოლექტიური ინტელექტი

დიდი ხნის წინ, ადამიანებმა აღმოაჩინეს ბუნებაში მწერების, ფრინველებისა და ცხოველების ქცევის მრავალი საიტერესო ასპექტი, რომლიც კოლექტიურ ქცევასთან არის დაკავშირებული [1]. ბოლო დროს ბიოლოგები და კომპიუტერული მეცნიერების ინჟინრები იკვლევენ ბიოლოგიური არსებების ევოლუციასა და კოლექტიურ ქმედებას რათა მოახდინონ ამ პრინციპების გამოყენება ოპტიმიზაციის ამოცანებში, ისეთ სფეროებში როგორიცაა სატელეკომუნიკაციო სისტემები, რობოტოტექნიკა, სატრანსპორტო სისტემები, სამხედრო და თავდაცვითი სისტემები და სხვა [2, 3, 4, 5].

„გროვის ინტელექტი“ წარმოადგენს მარტივი ავტონომიური აგენტების ჯგუფის კოლექტიურ ინტელექტს, სადაც აგენტები არიან სრულიად დამოუკიდებელნი, ურთიერთქმედებენ მარტივი ინსტრუქციებით და კონკრეტული მიზანი მიიღწევა აღნიშნული აგენტების კოოპერაციის ხარჯზე [6]. მაგალითად, ფრინველი, რომელიც დაფრინავს გუნდთან ერთად, არეგულირებს თავის მოძრაობას მეზობელ ფრინველებთან მიმართებით. ის ცდილობს დაიჭიროს ისეთი დისტანცია, რომ რაც შეიძლება ახლოს იფრინოს თავის მეზობლებთან, მაგრამ ხელი არ შეუშალოს და არ დაეჯახოს მათ. ფრინველი გუნდში არის დამოუკიდებელი.

ის არ იღებს ბრძანებებს ლიდერი ფრინველისგან, რადგანაც გუნდში ასეთი არ არსებობს. ნებისმიერ ფრინველს შეუძლია გუნდის წინ, ცენტრში ან უკან ფრენა - შეზღუდვის გარეშე.

როგორც ზემოთ ავღნიშნე კოლექტიური ინტელექტის კვლევაში ჩართულია მრავალი მეცნიერი. მკვლევარები ცდილობენ დაადგინონ როგორ ახერხებს სხვადასხვა წარმონაქმნების გუნდი ასეთ ორგანიზებულ ქმედებას. დღესდღეობით შექმნილია მრავალი მოდელი, რომელიც წარმოადგენს ბუნებრივი კოლექტიური ქმედების ანალოგს. მაგალითად:

- რეინოლდსმა (Reynolds) 1987 წელს შექმნა იმიტაციური მოდელი, რომელიც აღწერს ფრინველების გუნდის მოძრაობის სიმულაციას [7].
- კუბემ (Kube) და ჩანგმა (Zhang) შექმნეს ჭიანჭველების კოლონიის ერთობლივი ტრანსპორტირების მოდელი. მოდელი აღწერს ჭიანჭველების გროვის ქმედებას, როცა ერთ ჭიანჭველას მარტო არ ძალუძს დიდი ზომის საკვების მოტანა და ჭიანჭველების გუნდს ერთობლივად მოაქვს იგი [2].
- რესნიკმა (Resnick) შექმნა „StarLogo“ - ობიექტზე ორიენტირებული ენა, სადაც ილუსტრირებულია სხვადასხვა სახის თვითორგანიზაციის და დეცენტრალიზაციის ნიმუშები - ხელოვნური ჭიანჭველების, საგზაო საცობების, ტერმიტების და ა.შ. მაგალითზე [8].
- ტერზოპულოსმა (Terzopoulos) და სხვებმა შექმნეს ხელოვნური თევზების სამგანზომილებიანი სამყარო. მათ შექმნეს სიმულატორი, სადაც თევზი როგორც დამოუკიდებელი აგენტი ემორჩილება ჰიდროდინამიკურ კანონებს და კუნთების გამოყენებით ახერხებს გადაადგილებას [9].
- სიმსმა (Sims) აღწერა ვირტუალური არსებების ევოლუცია რეალური ევოლუციის მაგალითზე. ხელოვნური არსებები ცდილობენ მოიპოვონ საერთო რესურსები. ამ ბრძოლაში გადარჩება ძლიერი ინდივიდები, რომლებიც შემდეგ ევოლუციას და რეპროდუქციას განიცდიან [10].

1.2 მულტი-რობოტული სისტემები

მულტი-რობოტული სისტემები ე.წ. „გროვის ინტელექტის“ ერთ-ერთ უმნიშვნელოვანეს დარგს წარმოადგენს დღესდღეობით. „გროვა“ უზრუნველყოფს ისეთ შესაძლებლობებს, მაგალითად დავალების შესრულების წარმადობა, მაღალი საიმედოობა, ერთეული რობოტის სიმარტივე და შესაბამისად წარმოების სიიაფე, რისი შესრულებაც შეუძლებელია ერთი რობოტის მიერ. ასეთი სისტემები შეიძლება გამოვიყენოთ მრავალ დარგში, ისეთში როგორიცაა მაგალითად მოქნილი წარმოების სისტემები, კოსმოსური აპარატები, ტექნიკური მომსახურების სისტემები, მშენებლობა, სოფლის მეურნეობა, მედიცინა და სხვა [11].

მულტი-რობოტული სისტემის მრავალი მოდელი არსებობს. მაგალითად, ბენიმ (Beni) წარმოადგინა ფიქური რობოტული სისტემის კონცეპცია. სისტემა შედგება ავტონომიური, არასინქრონიზებული, არაინტელექტუალური რობოტებისგან, რომლებიც ერთმანეთთან კოოპერირებენ სასრულ n - განზომილებიან უჯრედულ სივრცეში განაწილებული კონტოლის მექანიზმით. შეზღუდული კომუნიკაცია არსებობს მხოლოდ მეზობელ რობოტებს შორის. რობოტები მოქმედებენ დამოუკიდებლად და ამასთან ერთად შეთანხმებულად რათა უზრუნველყონ წინასწარ განსაზღვრული გლობალური დავალების შესრულება [4]. ჰაკვუდმა (Hackwood) და ბენიმ (Beni) შემოგვთავაზეს მოდელი, სადაც რობოტი არის ძალიან მარტივი, მაგრამ მისი ქმედება ნაკარნახებია „მიმთითებელი“ (Signpost Robots) რობოტების მიერ. მიმთითებელ რობოტებს შეუძლიათ იმოქმედონ და შეცვალონ ახლოს მდებარე გროვის ინდივიდების შინაგანი მდგომარეობა [12].

ქმედებაზე დაფუძნებული კონტროლის სტრატეგიას იყენებს მრავალი მკვლევარი. მაგალითად ბრუკსმა ის გამოიყენა მარტივი დამოუკიდებელი რობოტების მართვისთვის - მარტივი ამოცანების შესასრულებლად [13]. უიამამ (Ueyama) იგივე მიდგომით შემოგვთავაზა კომპლექსური რობოტების ორგანიზება ხის სტრუქტურის სახით, სადაც კომუნიკაცია იზღუდება ხის

ტოტების იერარქიის დონეზე [14]. მატარიკმა (Mataric) აღწერა ერთგვაროვანი რობოტების პოპულაციის ქმედება სხვადასხვა საკომუნიკაციო შეზღუდვების პირობებში. თავდაპირველად რობოტები მოქმედებენ ისე რომ არ იციან ერთმანეთის შესახებ, შემდეგ ინფორმირდებიან ერთმანეთზე და ბოლოს თანამშრომლობენ კიდევ. ნათელი გახდა რომ რობოტებს შორის კომუნიკაციის გაუმჯობესებასთან ერთად გროვას შეუძლია უფრო რთული ამოცანების შესრულება [15].

არსებობს მრავალი ოპტიმიზაციის მეთოდი ევოლუციური ალგორითმების გამოყენებით. მაგალითად, ფუკუდამ (Fukuda) წარმოადგინა განაწილებული გენეტიკური ალგორითმი განაწილებული დაგეგმარებისთვის ფიქური რობოტების სისტემაში. ეს მეთოდები მოიცავს თვითორგანიზების, გადაწყვეტილების მიღების, სწავლებისა და ადაპტაციის სტრატეგიებს [16, 17].

რობოტების გროვა არის უფრო მეტი ვიდრე უბრალოდ დამოუკიდებელი აგენტების ქსელი. ის წარმოადგენს პოტენციურად რეკონფიგურირებად ქსელს, რომელსაც ახასიათებს კოორდინაცია და კომუნიკაცია და შეუძლია ინტერაქცია და ადაპტაცია ცვალებად გარემოსთან.

თავი 2 მულტი-რობოტული სისტემები

2.1 საგნობრივი სფერო. ამოცანის დასმა. კვლევის აქტუალობა და მნიშვნელობა

საგნობრივ სფეროს წარმოადგენს ტექნოგენური კატასტროფებით გამოწვეული დაბინძურებული გარემო, რომლის მონიტორინგი შეადგენს ძალიან რთულ და ამასთანავე მნიშვნელოვან პრობლემას. რადიაციით გარემოს ძლიერი დაბინძურებით გამოწვეულმა შედეგებმა, განსაკუთრებით ატომური ენერგეტიკის და სამხედრო სფეროებში, წარმოშვა მზარდი მოთხოვნა ისეთ საშუალებებზე, რომელიც უზრუნველყოფს დაბინძურების წყაროების ავტომატიზებულ მონიტორინგს, განსაზღვრავს რადიაქტიული წყაროების გამოსხივების, მომწამვლელი გაზების და ა.შ. ადამიანისთვის უსაფრთხო ზღვრებს. დღესდღეობით, არსებობს საჭიროება მაღალი სიზუსტისა და მგრძნობელობის მქონე მოწყობილობებზე დაბინძურების წყაროებისა და საშიში ზონების აღმოჩენა-გაზომვის თვალსაზრისით [18, 19, 20].

მეორეს მხრივ, ადაპტური ოპტიმიზაციის აუცილებლობა გამომდინარეობს გარემოს არასტაციონალურობიდან, როდესაც მიზნობრივი ფუნქციის განაწილების ლანდშაფტი დინამიურად იცვლება. დასმული ამოცანის გადასაჭრელად საჭიროა მონიტორინგი განხორციელდეს განაწილებული მულტი-რობოტული ქსელიდან გარემოზე ინფორმაციის უწყვეტად შეკრების გზით. მულტი-რობოტული სისტემა არის ავტონომიური მოწყობილობების (რობოტების) რეკონფიგურირებადი სტრუქტურის ქსელი, რომელსაც შეუძლია მონიტორინგი გაუწიოს გარემო პირობების ცვალებადობას. გარემოზე ინფორმაციის შეგროვება ხდება მობილური უკაბელო სენსორებით. ჩვენ განვიხილავთ რადიაციის სენსორს, როგორც არხს გარემოსა და ავტომატურ მონიტორინგის სისტემას შორის. ეფექტური სენსორების და სენსორული სისტემების (ქსელების) განვითარებით და ამასთან ერთად, ხელოვნური ინტელექტის გამოყენებით,

მიიღწევა საჭირო ინფორმაციის შეგროვება-დამუშავება და განგაშის ფუნქციის რეალიზება. ასეთ სისტემებს მრავალი ფუნქცია შეიძლება ჰქონდეთ, მაგ: გარემოს ფიზიკური და ეკოლოგიური მონიტორინგი, ცოცხალი სამყაროს გარემოს მონიტორინგი (ფლორისა და ფაუნის სახეობების განსაზღვრა და მათი ქცევის შესწავლა), სეისმური აქტივობების დაკვირვება, სამხედრო დაზვერვა და სხვა [21, 22, 23].

პრობლემები, რომლებიც დაკავშირებულია უპილოტო მიწისზედა ტრანსპორტის ქცევის მოდელირებასა და მართვაზე, დაფუძნებულს კოლექტიური ინტელექტის ახალ მიდგომებზე, წარმოადგენს ამ კვლევების მთავარ საგანს.

კვლევა ეფუძნება ხელოვნური ინტელექტის მეთოდების ფართო სპექტრს, ისეთ თანამედროვე მეთოდებს, როგორიცაა: ნაწილაკთა გროვის ოპტიმიზაცია, ჭიანჭველების კოლონიის ოპტიმიზაცია, კოლექტიური ქცევის მათემატიკური მოდელირება და მრავალმიზნობრივი მართვა ენტროპიაზე დაფუძნებული მეტრიკის ბაზაზე.

2.1.1 უპილოტო მიწისზედა ტრანსპორტის ნავიგაციის მოდელირების და მართვის ამოცანა განუსაზღვრელ პირობებში

ჩვენი ამოცანის მიზანს წარმოადგენს უპილოტო მიწისზედა ტრანსპორტის მიერ საძიებო თუ სადაზვერვო დავალებების ავტონომიური შესრულებისათვის აუცილებელი ინფორმაციული ტექნოლოგიების, კერძოდ, პროგრამული უზრუნველყოფის განვითარება. თანამედროვე უპილოტო მიწისზედა ტრანსპორტი ფართოდ გამოიყენება სხვადასხვა სამხედრო, სახიფათო ნარჩენების ლიკვიდაციისა თუ პლანეტარული კვლევებისათვის. შესაბამისად, უპილოტო მიწისზედა ტრანსპორტი უნდა იყოს აღჭურვილი სპეციალური პროგრამებით, ბუნებრივი თუ ადამიანის მიერ შექმნილი დაბრკოლებების პირობებში ავტონომიური გადაადგილების მიზნით. ამისათვის, უპილოტო მიწისზედა ტრანსპორტს უნდა შეეძლოს

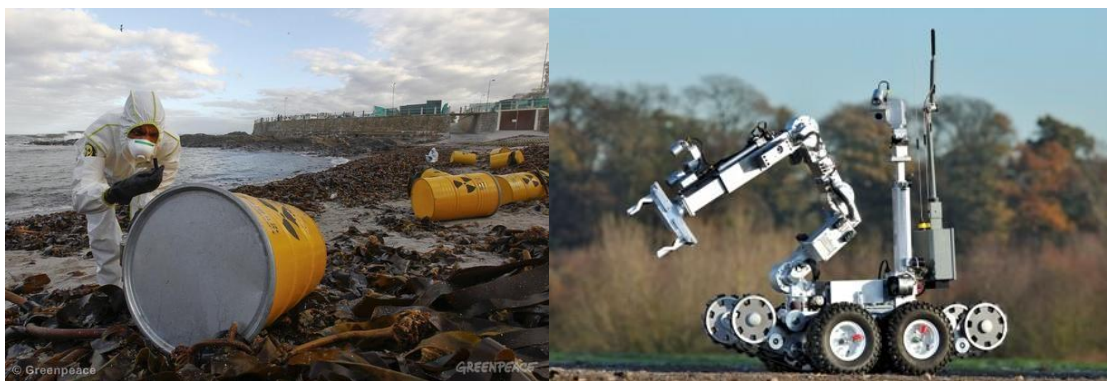
შეიცნოს მიწის ზედაპირის გეომეტრია, ნიადაგის საფარის მასალა (მიწის ტიპი), როგორც ღამით, ისე ნისლში.



სურათი 1. UGVs მართვა განუსაზღვრელ პირობებში

2.1.2 უპილოტო მიწისზედა ტრანსპორტის ნავიგაციის მოდელირების და მართვის ამოცანა გარემოს მონიტორინგის მიზნით

გარემოს მონიტორინგი, განსაკუთრებით, მაღალი რისკის შემცველი სამრეწველო პროექტების შემთხვევაში, შეიძლება განხორციელდეს ფართომასშტაბიანი, სივრცულად განაწილებული მობილური მულტი-სენსორული სისტემის მეშვეობით. იგი წარმოადგენს განაწილებულ მოწყობილობათა მოქნილ, რეკონფიგურირებად ქსელს, სადაც შესაძლებელი ხდება ფიზიკური თუ გარემო პირობებისათვის თითოეული მონაკვეთის შესახებ ერთობლივი გაზომვების ჩატარება და მონაცემთა შეგროვება, ასევე, გარემოს შეცნობა ან მონიტორინგი. მოქნილი, რეკონფიგურირებადი სენსორული ქსელის კონტროლი და მართვა წარმოადგენს ურთულეს პრობლემას, რომელშიც სისტემა, გარემოსთან თუ სამეცნიერო მოთხოვნათა ცვლილებებთან მიმართებითა და გარკვეული წონასწორობის დაცვით, უნდა ახორციელებდეს სიმძლავრეთა ხარჯვის, კომუნიკაციის მართვისა და წარმადობის ადაპტაციას.



სურათი 2. UGVs მართვა გარემოს მონიტორინგის მიზნით

2.1.3 კვლევის მიზანი

კვლევის მთავარ მიზანს წარმოადგენს თანამედროვე ხელოვნური ინტელექტის თეორიის - მულტი-აგენტური სისტემების ქცევის მოდელირების შესახებ გარკვეული წარმოდგენების განვითარება კოლექტიური ინტელექტის ჩათვლით და მიღწეული შედეგების დანერგვა უპილოტო მიწისზედა ტრანსპორტის (Unmanned Ground Vehicle systems - UGVs) სისტემებში, რომელიც იქმნება ავტონომიური რობოტების ჯგუფის სახით საგანგებო სიტუაციებში სხვადასხვა ოპერაციათა მხარდაჭერისათვის. უპილოტო მიწისზედა ტრანსპორტის სისტემის გამოყენება სხვადასხვა მრავალჯერადი დავალების შესრულებისას კოოპერაციის, კოორდინაციისა და მოქნილი კოლაბორაციის საშუალებას იძლევა. რობოტული ჯგუფისათვის, დავალების შესრულების ფარგლებში სტრუქტურული თუ ფუნქციური გადაწყობის მეშვეობით, უპილოტო მიწისზედა ტრანსპორტს შეუძლია უზრუნველყოს გარემო ფაქტორებისაგან სავარაუდო საშიშროების თავიდან არიდება ან მისი შემცირება.

ჩვენი მიზანია ხელოვნური ინტელექტის მეთოდებზე, განსაკუთრებით, გროვის ქცევის მოდელირებისა და კოლექტიური ინტელექტის კონცეფციაზე დაფუძნებული ახალი მიდგომების შემუშავება - ფიზიკური თუ გარემო რისკების მონიტორინგისა და შეფასების, რობოტების ჯგუფის ადაპტური მართვის (მათ შორის, უპილოტო მიწისზედა ტრანსპორტის სისტემის), განუსაზღვრელ პირობებში ნავიგაციის, საგანგებო სიტუაციებში სამაშველო

თუ სხვა ოპერაციათა მხარდაჭერის, ასევე აგრესიულ თუ მტრულ გარემოში სამხედრო ამოცანების შესრულების თვალსაზრისით.

2.1.4 კვლევის სიახლე

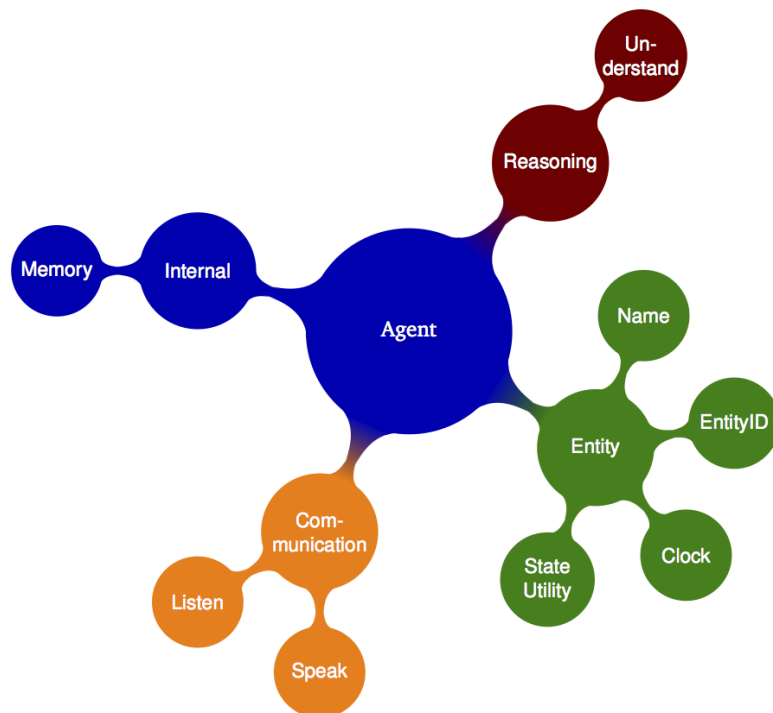
კვლევის მთავარ სიახლეს წარმოადგენს სოციალური სისტემების კოლექტიურ ქცევებთან დაკავშირებული სამეცნიერო კონცეფციების შემუშავება და განვითარება. მრავალმიზნობრივი ევოლუციური ალგორითმები (გენეტიკური ალგორითმები, ნაწილაკთა გროვის ოპტიმიზაცია, ჭიანჭველების კოლონიის ოპტიმიზაცია და სხვ.), აგრეთვე პარალელური გამოთვლების მიდგომები და გადაწყვეტილების მიღების თანამედროვე პროცედურები ეფუძნება მულტი-აგენტური მოდელირებისა და დისკრეტულ ხდომილებათა მოდელირების მეთოდებს. ჩვენს მიერ შემუშავებული მიდგომები შეიძლება განვიხილოთ როგორც ახალი საინჟინრო გამოთვლითი პარადიგმა, რომელიც ემყარება დინამიური სისტემებისათვის დამახასიათებელ ენტროპიისა და სინერგიის ცნებებს. თუ განვიხილავთ რთულ სისტემას, როგორც ინტერაქტიულ, მულტი-აგენტურ ჰეტეროგენულ ქაოტურ სისტემას მრავალგანზომილებიანი, რთული იერარქიული სტრუქტურით, მაშინ მისი მოდელირება წარმოადგენს ურთულეს პრობლემას. ეს განპირობებულია ადამიანის, როგორც არაწრფივი და არამკაფიო ფაქტორის არსებობით, რაც შესაბამისად დაკავშირებულია ქცევის თავისუფლების მეტად მაღალ ხარისხთან. მოცემულ კონტექსტში გროვის მოდელის აგება და მისი ოპტიმიზაცია შესაძლებელია ენტროპიის მინიმიზაციის კრიტერიუმის მიხედვით.

2.2 მულტი-რობოტული სისტემების მიმოხილვა

ხელოვნური ინტელექტის ერთ-ერთ უახლეს მიმართულებას წარმოადგენს *მულტი-აგენტური სისტემების (მას) მეთოდი*, რომელიც განიხილება როგორც გადაწყვეტილების მიღების ეფექტური საშუალება ცალკეული თუ ერთობლივი მიზნების მისაღწევად. ყველგან, სადაც კი

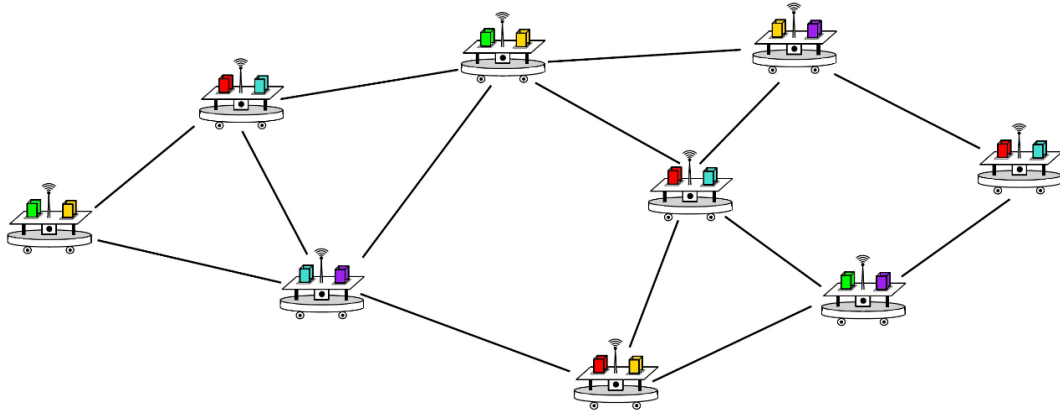
გვაქვს ცოცხალი ორგანიზმებისა თუ ტექნიკური ობიექტების გარკვეული ერთობლიობა, რომლის ამოცანასაც წარმოადგენს გარკვეული ჯგუფური სამუშაოს შესრულება, ან ამა თუ იმ ამოცანის გადაწყვეტა, შეიძლება ითქვას, რომ არსებობს ჯგუფის მართვის ამოცანა.

ზოგადად, აგენტი არის რაიმე მიზნით აღჭურვილი ობიექტი (არსება), კონკრეტულად ამ სისტემისთვის კი, იგი წარმოადგენს მოაზროვნე, გადაწყვეტილების მიმღებ და მის შემსრულებელ ობიექტს, რომელიც გამოიყენება სხვადასხვა რთული ამოცანების გადასაწყვეტად მოცემულ გარემოში (სურათი 3). გროვა კი, თავის მხრივ, წარმოადგენს საერთო მიზნით გაერთიანებული აგენტების სიმრავლეს. მულტი-აგენტური სისტემები გამოიყენება ისეთი ამოცანების გადასაწყვეტად რომელთა ამოხსნაც ძალიან რთულია ან შეუძლებელია ერთი აგენტის მიერ. აღნიშნული სისტემები შედგება ავტონომიური აგენტებისგან, რომლებსაც არ გააჩნიათ სისტემის გლობალური ცოდნა და საერთო მიზნის მიღწევისთვის ოპერირებენ შეზღუდული ცოდნით და შესაძლებლობებით. სისტემა დაფუძნებულია კოლექტიური ინტელექტის მეთოდებზე, რომელიც თავის მხრივ ეფუძნება ბიოლოგიური არსებების კოლექტიური ქცევის თვისებას.



სურათი 3. მარტივი აგენტის სტრუქტურა

მულტი-რობოტული სისტემა არის მულტი-აგენტური სისტემა, რომელიც შედგება ფიზიკური აგენტ-რობოტებისგან. იგი მთლიანად იზიარებს მას-ის მიდგომებსა და კონცეფციებს და მათ განიხილავს ფიზიკური რობოტების ჭრილში (სურათი 4).



სურათი 4. მულტი-რობოტული სისტემა

ზოგადად, მულტი-აგენტური სისტემების მიდგომა, რომელიც მიეკუთვნება ხელოვნური ინტელექტის ალგორითმების კლასს, ასევე ფართოდ გამოიყენება ძიებისა და მანქანური სწავლების მეთოდებში.

2.2.1 მულტი-რობოტული სისტემის არქიტექტურა

მულტი-რობოტული სისტემები ხასიათდება ისეთი ატრიბუტებით, როგორიცაა: ზომა, კომპოზიცია, თვითორგანიზება და თვითმართვა, კომუნიკაცია, კოოპერაცია, ცოდნის განმტკიცება, რობოტების (აგენტების) რაოდენობა და კოლექტიური ინტელექტი. ჩამოთვლილი თვისებებიდან გამოვყოფთ რამდენიმე კრიტიკულ თვისებას, რომლებიც განსაზღვრავს მთელი სისტემის ეფექტურ მუშაობას. მათ დეტალურად ქვემოთ განვიხილავთ.

მულტი-რობოტულ სისტემებს გააჩნია სამი ძირითადი უპირატესობა. პირველი, მასშტაბირება - არქიტექტურა, რომელიც რობოტების რაოდენობის მარტივად ზრდის საშუალებას იძლევა რამდენიმე ერთეულიდან ათასობით რობოტამდე. მეორე, თვითორგანიზება - სისტემა არის ტოლერანტული

ცვლილებების მიმართ და სწრაფი ორგანიზების საშუალებას იძლევა. მესამე, სიზუსტე და მდგრადობა - მიიღწევა არა მარტო რობოტების რაოდენობის ზრდით, არამედ სისტემის არქიტექტურისა და თითოეული ინდივიდის კომპონენტების სწორად შერჩევით [24].

თავის მხრივ, სისტემაში შემავალ რობოტებს გააჩნიათ შემდეგი თვისებები:

- ავტონომიურობა - რობოტები სისტემაში სრულად ან ნახევრად ავტონომიურები არიან.
- ლოკალური ხედვა - არც ერთ რობოტს არ აქვს სისტემაზე სრული წარმოდგენა, ან სისტემა იმდენად რთულია რობოტისთვის, რომ ამ ცოდნის პრაქტიკულად გამოყენების საშუალება არ ეძლევა.
- ცენტრალიზაცია - სისტემაში არსებობს ერთი, ძირითადი, მაკონტროლებელი რობოტი და სწორედ იგი იღებს ძირითად გადაწყვეტილებებს.
- დეცენტრალიზაცია - სისტემაში არ არსებობს მაკონტროლებელი რობოტი, ანუ ყველა რობოტი თანაბარია.

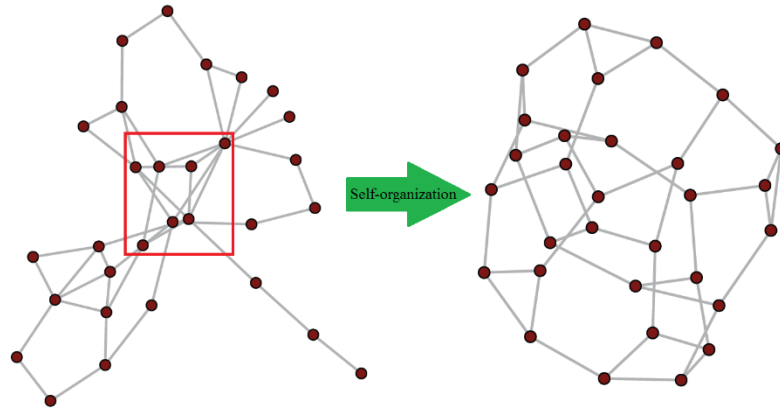
სისტემის კლასიფიკაცია შეიძლება სამი ძირითადი თვისებით [25]:

1. ფუნქციონალური - აღწერს სისტემის ფუნქციას და გამოყენების არეს.
2. სტრუქტურული - აღწერს სისტემის სტრუქტურას, ანუ რა კომპონენტებისგან შედგება იგი.
3. ქმედითი - აღწერს სისტემის კომპონენტების ფუნქციონალურ და დროზე დამოკიდებულ ქმედებებს.

როგორც ზემოთ იყო აღნიშნული, მულტი-რობოტულ სისტემებს გააჩნია რამდენიმე კრიტიკული თვისება, რომლებიც განსაზღვრავს მთელი სისტემის ეფექტურ მუშაობას, კერძოდ:

თვითორგანიზება. მულტი-რობოტულ სისტემებს ახასიათებთ თვითორგანიზებულობა, თვითმართვა და კონტროლის სხვა პარადიგმები, რომლებიც დაკავშირებულია რთულ ქცევით ქმედებებთან, მიუხედავად

აგენტების სიმარტივისა. ეს თვისებები სისტემას გაუჩნდება მას შემდეგ, რაც აგენტები გაცვლიან ცოდნას წინასწარ განსაზღვრული საკომუნიკაციო არხითა და პროტოკოლით (სურათი 5).



სურათი 5. მულტი-რობოტული სისტემის თვითორგანიზება

კომუნიკაცია. იგი მულტი-რობოტული სისტემის ერთ-ერთი მნიშვნელოვანი კომპონენტია. რთული ამოცანების გადასაჭრელად, სისტემა უნდა იყოს დინამიურად რეკონფიგურირებადი და ადაპტური, ეს კი მიიღწევა სისტემის აგენტებს შორის ინფორმაციის გაცვლის გზით. კომუნიკაცია შეიძლება იყოს პირდაპირი, არაპირდაპირი, ან ამ ორის კომბინაცია. არაპირდაპირი კომუნიკაციის დროს, ინფორმაცია გადაეცემა ორ ან მეტ ინდივიდს ერთდროულად, მაშინ როცა პირდაპირი კომუნიკაციის დროს - კონკრეტულ ინდივიდს. კომუნიკაციის მექანიზმის ეფექტურობა ყოველ კონკრეტულ შემთხვევაში დამოკიდებულია დასმულ ამოცანაზე. აღსანიშნავია, რომ რობოტების კოოპერაციისათვის კომუნიკაციის ორივე მეთოდი მისაღებია, ანუ მისი ეფექტურობის თვალსაზრისით, არ არის აუცილებელი, რომ ცალკეულმა რობოტმა ცოდნა გაუზიაროს გროვის ყველა ინდივიდს [26].

ცოდნის განმტკიცება. სწავლება მულტი-რობოტულ სისტემებში ხორციელდება ისეთივე სახით, როგორც ადამიანთა საზოგადოებაში. ანუ, თითოეული რობოტი თავად არ სწავლობს ყველაფერს ნულიდან, ცოდნის ნაწილს იღებს სხვა რობოტებთან ინფორმაციის გაცვლის გზით.

ცოდნის განმტკიცება არის ინტერაქტიული მიდგომა, რომელიც უახლოვდება ოპტიმალური მართვის სტანდარტულ ტექნიკას, რომელიც ცნობილია დინამიური პროგრამირების სახით. მისი მოდელი აღიწერება შემდეგნაირად: გარე სამყარო წარმოდგენილია როგორც დისკრეტული დროის, სასრული რაოდენობის მდგომარეობის და მარკოვის გადაწყვეტილების მიღების პროცესის ერთობლიობა. თითოეული ქმედება ფასდება ჯილდოთი. მიდგომის მიზანია თითოეული ქმედების ჯილდოს შემცირების მაქსიმიზაცია. თითოეული შემეცნებითი აგენტი იყენებს ერთ-ბიჯიან Q – *learning* ალგორითმს. სწავლება მიმდინარეობს ქმედების განმსაზღვრელი მდგომარეობის/ქმედების ფუნქციით Q , რომელიც ყოველი წყვილისთვის აფასებს ჯილდოს შემცირებას. მოცემულია x მდგომარეობა და ხელმისაწვდომი ქმედებები a_i . Q – *learning* აგენტი ირჩევს ყოველ a ქმედებას ალბათობით, რომელიც მოიცემა ბოლცმანის განაწილებით:

$$p(a_i|x) = \frac{e^{Q(x,a_i)/T}}{\sum_{k \in \text{actions}} e^{Q(x,a_i)/T}} \quad (1)$$

სადაც, T - არის ტემპერატურული პარამეტრი, რომელიც არეგულირებს გადაწყვეტილების ქაოტურობას (შემთხვევითობას).

ამის შემდეგ, აგენტი ასრულებს მოქმედებას, იღებს შუალედურ ჯილდოს - r და გადადის შემდეგ, y - მდგომარეობაზე.

თითოეულ ბიჯზე, აგენტი აახლებს $Q(x, a)$ - ს დადებითი შემეცნების ტემპით β :

$$Q(x, a) \leftarrow Q(x, a) + \beta(r + \gamma V(y) - Q(x, a)) \quad (2)$$

სადაც, γ ($0 \leq \gamma < 1$) ჯილდოს შემცირების პარამეტრი, და $V(x)$ მოცემულია შემდეგი ფორმულით:

$$V(x) = \max_{b \in \text{actions}} Q(x, b) \quad (3)$$

როდესაც აგენტის კვლევის პროცესშია, Q თანდათან უმჯობესდება $V(x)$ მიდგომით:

$$E\{\sum_{n=1}^{\infty} \gamma^{n-1} r_{t+1}\} \quad (4)$$

სადაც, r_t არის ჯილდო, მიღებული t დროის მომენტში, $t - 1$ დროში არჩეული ქმედებისთვის.

ვოტკინსმა და დაიანმა (Watkins and Dayan) დაამტკიცეს, რომ Q – *learning* ალგორითმი პოულობს ოპტიმალურ ამონახსნს მარკოვის გადაწყვეტილების მიღების სასრული პროცესისთვის [27].

კოოპერაცია. რობოტების კოოპერაცია (შეთანხმებული მოქმედება) არის ძალიან მნიშვნელოვანი ასპექტი, რადგან ავტონომიური რობოტების მიერ დავალების შესრულება მიიღწევა მათი კავშირებით, ცოდნისა და რესურსების გაზიარებით. ბუნებრივი მიდგომა იმაში მდგომარეობს, რომ ეს ქმედებები უნდა დაიყოს პატარ-პატარა შეთანხმებულ ქმედებებად, რომელთაგან თითოეულს შეასრულებს რობოტების ჯგუფი, რომელშიც თითოეულ რობოტს მკაცრად განსაზღვრული როლი გააჩნია.

იმისათვის, რომ მივაღწიოთ წინასწარ განსაზღვრულ კოოპერაციის დონეს მულტი-რობოტულ სისტემებში, საჭიროა წინასწარ განისაზღვროს სისტემის არქიტექტურა და კონტროლის სტრატეგიები. ასეთი რთული სისტემების დიზაინზე ბევრი ასპექტი მოქმედებს, მაგალითად: ამოცანების (ქმედებების) დეკომპოზიცია, ამოცანების განაწილება, როლების განაწილება, რობოტებს შორის კავშირის და ურთიერთობის ფორმა, რობოტების კოოპერაცია, კოორდინაცია, კონფლიქტების გადაჭრა.

კოოპერაცია მულტი-აგენტურ სისტემებში უზრუნველყოფს წარმადობას, რადგან ამ დროს ხდება აგენტების შესაძლებლობების მაქსიმალურად ეფექტური გამოყენება. კოოპერაცია სამი კომპონენტის ერთობლიობას წარმოადგენს:

- კოლაბორაცია - ორიენტირებულია ამოცანების განაწილებაზე. ურთიერთობის პროტოკოლი გამოიყენება იმისთვის, რომ აგენტს დაევალოს ამა თუ იმ სპეციფიური ამოცანის შესრულება. ამ ინტერაქციით (დიალოგით) დგინდება აგენტის ხელმისაწვდომობა, ისევე როგორც აგენტის შესაძლებლობები.

- კოორდინაცია - აგვარებს სინქრონიზაციის და მართვის პრობლემებს. ამ შემთხვევაში აუცილებელია განისაზღვროს დროის რა მომენტში უნდა შესრულდეს ესა თუ ის ქმედება. მთელი გუნდის წარმადობა დამოკიდებულია თითოეული აგენტის დროით სიზუსტეზე.
- კონფლიქტების გადაჭრა - როგორც წესი, აგენტები ცვლიან რესურსებს პარალელურ რეჟიმში, რაც კონფლიქტების და შეფერხებების მიზეზი ხდება. ამიტომ, აუცილებელია ამ პრობლემების აღმოფხვრის მექანიზმების შემუშავება და გატარება [26].

2.2.2 მულტი-რობოტული სისტემის მართვის სტრატეგიები

მულტი-რობოტული სისტემის მართვაში არსებობს ორი მიდგომა: ცენტრალიზებული და დეცენტრალიზებული.

ცენტრალიზებული მართვის სტრატეგია გულისხმობს, რომ ყველა რობოტი მოქცეულია ერთი, ლიდერი რობოტის კონტროლის ქვეშ. ეს გულისხმობს იმას, რომ თითოეული რობოტის მოძრაობა დამოკიდებულია ლიდერის პოზიციასა და ლიდერსა და i -ურ მიმდევარს შორის დისტანციაზე. ამ ფორმირების (რობოტული ქსელის) შემდეგი, სასურველი პოზიცია განისაზღვრება ლიდერის მიმდინარე პოზიციითა და ასევე მთელი სისტემის ქცევის განმსაზღვრელი რამდენიმე გეომეტრიული კონსტანტით. ცენტრალიზებული მართვის კანონი განისაზღვრება ლიდერი რობოტის მდგომარეობით, თავის მხრივ კი, იგი განსაზღვრავს დანარჩენი რობოტების გადაადგილების ვექტორს (სიჩქარე და მიმართულების კუთხე) (სურათი 6 (ა)).

დეცენტრალიზებული მართვის სტრატეგია დაფუძნებულია ლიდერ-მიმდევარ სტრუქტურაზე. ასეთ სისტემაში, კვლევის დასაწყისში აირჩევა ერთი ლიდერი, ხოლო დანარჩენი რობოტები ავტომატურად ხდებიან მიმდევრები. მიების პროცესში თუ აღმოჩნდა ისე, რომ რომელიმე მიმდევარი რობოტის პოზიცია უკეთესია, ვიდრე ლიდერი რობოტის, მაშინ მიმდევარი რობოტი ავტომატურად გახდება ლიდერი და პირიქით. ლიდერი რობოტის

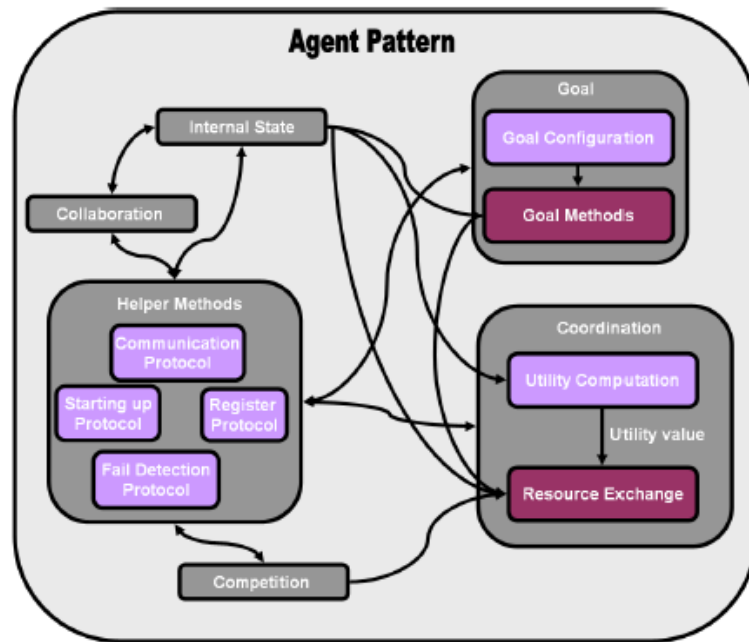
Figure 1 consists of two network diagrams. Diagram (a) shows a star topology with a central node connected to 20 peripheral nodes. Diagram (b) shows a hierarchical topology with a central node connected to 4 intermediate nodes, which are further connected to 20 peripheral nodes.

2.2.3 მულტი-აგენტური სისტემის პროგრამული არქიტექტურა

35

სისტემის პროგრამულად აღწერისთვის, თავდაპირველად საჭიროა მასში გამოსაყენებელი ტერმინების განსაზღვრა/აღწერა:

- აგენტის არქიტექტურა - აღწერს მოდულებს და ამ მოდულების ერთმანეთთან ინტერაქციას, რომლითაც ერთი აგენტი არის წარმოდგენილი. მაგალითად: როგორც წესი, სისტემას გააჩნია კომუნიკაციის მოდული, რომლითაც იგი კომუნიკაციას ამყარებს როგორც მომხმარებელთან, ასევე სხვა აგენტებთან (სურათი 7).
- მულტი-აგენტური სისტემის მოწყობა (ორგანიზაცია) - აღწერს თუ როგორ არიან აგენტები ორგანიზებული მულტი-აგენტურ სისტემაში. კონკრეტულად, იგი განსაზღვრავს აგენტების როლებს და აგენტებს შორის ურთიერთობის ტიპებს.
- მულტი-აგენტური ინფრასტრუქტურა - იგი განსაზღვრავს აგენტის არქიტექტურისა და მულტი-აგენტური სისტემის ორგანიზების დამოკიდებულებას (თუ ასეთი დამოკიდებულება არსებობს), რაც საშუალებას იძლევა აიგოს კონკრეტულ ამოცანაზე ორიენტირებული სისტემა.
- მულტი-აგენტური ინფრასტრუქტურული სერვისები - ეს არის სერვისები, როგორიცაა, მაგალითად: სისტემური დიზაინის და მოდელების აგების ხელსაწყოები, აგენტების ადგილმდებარეობის განსაზღვრის მექანიზმები, სისტემის ეფექტურობის და რესურსების უტილიზაციის ხელსაწყოები, მონაცემთა გაცვლის პროტოკოლები, მონაცემთა დაცვის პროტოკოლები და სერტიფიკატები და სხვა [30, 31].



სურათი 7. აგენტის შინაგანი არქიტექტურის მაგალითი

სისტემის თითოეული აგენტი სამი მოდელის ერთობლიობას წარმოადგენს, რომლებიც აღწერენ აგენტის ინფორმაციულ და მოტივაციულ მდგომარეობას და პოტენციურ ქმედებებს [32]:

- ე.წ. „ნდობითი მოდელი“ (Belief Model) – აღწერს იმ გარემოს და რობოტის შიდა მდგომარეობას, რომლითაც შეიძლება ხასიათდებოდეს კონკრეტული კლასის აგენტი და ქმედებებს რაც შეიძლება განახორციელოს. ეს ინფორმაცია ინახება „ნდობით სიმრავლეში“ (Belief Set). ერთი ან რამდენიმე „ნდობითი მდგომარეობით“ (Belief State), რომელიც არის „ნდობითი სიმრავლის“ კონკრეტული წარმომადგენელი, შეიძლება განისაზღვროს აგენტის საწყისი მენტალური მდგომარეობა (Mental State).
- ე.წ. „მიზნობრივი მოდელი“ (Goal Model) – აღწერს მიზნებს, რომელიც შეიძლება დაისახოს აგენტმა და ხდომილებებს რომელსაც შეიძლება მოჰყვეს აგენტის საპასუხო ქმედება. ასეთ მოდელში ინფორმაცია ინახება „მიზნობრივ სიმრავლეში“ (Goal Set), რომელიც განსაზღვრავს მიზნებისა და ხდომილებების დომენს. ერთი, ან რამდენიმე

„მიზნობრივი მდგომარეობით“ (Goal State) შეიძლება განისაზღვროს აგენტის საწყისი მენტალური მდგომარეობა (Mental State).

- ე.წ. „გეგმის მოდელი“ (Plan Model) – აღწერს გეგმებს, რომელიც აგენტმა შეიძლება გამოიყენოს მიზნის მისაღწევად ან ხდომილებებზე შესაბამისი საპასუხო ქმედებისთვის. ინფორმაცია ინახება „გეგმის სიმრავლეში“ (Plan Set), რომელიც განსაზღვრავს ინდივიდუალური გეგმის თვისებებსა და ქმედების სტრუქტურას.

ობიექტზე ორიენტირებული მიდგომებით აღიწერება სისტემის მთავარი ობიექტები, მათი ქცევები, კავშირები და ურთიერთობები სხვა ტიპის ობიექტებთან. აღწერისას გამოიყენება სამი ტიპის მოდელის ერთობლიობა:

1. ობიექტის მოდელი - აღწერს სისტემაში არსებულ ობიექტებს, მათ მონაცემთა სტრუქტურებს, კავშირებს და ქმედებებს.
2. დინამიური მოდელი - განსაზღვრავს სისტემის მდგომარეობას, გადასვლას ერთი მდგომარეობიდან მეორეში, ხდომილებებს, აქტივობებს და ურთიერთქმედებებს, რომლითაც ხასიათდება სისტემის ქცევა.
3. ფუნქციონალური მოდელი - აღწერს მონაცემთა ნაკადებს სისტემის აქტივობის დროს, როგორც სისტემის კომპონენტებს შორის, ასევე კომპონენტების შიგნით.

თავი 3 მულტი-რობოტული სისტემების თეორიული ასპექტები

3.1 ოპტიმიზაციის ამოცანის ფორმალიზაცია

ჩვენი სამყაროს ერთ-ერთი ფუნდამენტური პრინციპი არის ოპტიმალური მდგომარეობის ძიება. ის ემყარება ფიზიკის კანონებს, იწყება მიკრო სამყაროდან (ატომი) და მთავრდება მაკრო სამყაროში. ზუსტად იგივე წესებით და მიზნებით ხელმძღვანელობს ბიოლოგიური პრინციპი „ძლიერის გადარჩენა“ (survival of the fittest), რომელიც ბიოლოგიურ ევოლუციასთან ერთად იძენს გარემო პირობებთან უკეთ ადაპტირების უნარს. აქ, ლოკალურ ოპტიმუმს წარმოადგენს გარემოსთან კარგად შეგუებული სახეობები, რომლებიც დომინირებენ სხვა სახეობებზე მოცემულ გარემოში. განვითარების იგივე დონეს მიაღწია როგორც ჰომო საპიენსმა, ისე გონებრივად ნაკლებად განვითარებულმა სახეობებმა, ჭიანჭველებმა, ბაქტერიებმა, ფუტკრებმა და სხვ.

ეს მიდგომები, როგორც ყველა მნიშვნელოვანი, ზოგადი და აბსტრაქტული რამ სამყაროში, აღიწერება მათემატიკური მოდელით. გლობალური ოპტიმიზაცია არის მეთოდი, რომელიც ორიენტირებულია ოპტიმიზაციის ამოცანების გადაწყვეტაზე და ამ მიზნით, იგი იყენებს აღნიშნულ მათემატიკურ მოდელს და რიცხვით ანალიზს. მისი მიზანია იპოვოს საუკეთესო x^* ელემენტი X სიმრავლიდან შემდეგ კრიტერიუმებზე დაყრდნობით $F = \{f_1, f_2 \dots f_n\}$. ეს კრიტერიუმები წარმოადგენილია მათემატიკური ფუნქციების საშუალებით და მათ ეწოდებათ „მიზნობრივი ფუნქციები“ (Objective functions).

მიზნობრივი ფუნქცია $f : X \rightarrow Y$, სადაც $Y \subseteq R$ არის მათემატიკური ფუნქცია, რომლის საგანიც არის ოპტიმიზაცია. მიზნობრივი ფუნქციის Y არე უნდა წარმოადგენდეს ნამდვილი რიცხვების ქვესიმრავლეს ($Y \subseteq R$). X არე კი წარმოადგენს ე.წ. საპრობლემო არეს და იგი შეიძლება იყოს ნებისმიერი ტიპის ელემენტი, როგორც მათემატიკური, ასევე სხვა ტიპის, მაგალითად:

რიცხვი, მასივი, მშენებლობის გეგმა, ქარხნის კონვეიერის დამგეგმავი (scheduler) და სხვა. ის დამოკიდებულია ამოცანაზე, რომელსაც უნდა მიუყენოთ ოპტიმიზაციის პროცესი. ასევე, მიზნობრივი ფუნქციაც შესაძლოა იყოს მათემატიკური გამოსახულება. ის შეიძლება წარმოადგენდეს რთულ ალგორითმს, რომელიც საჭიროებს მრავალ იტერაციულ სიმულაციას. გლობალური ოპტიმიზაცია მოიცავს ყველა მეთოდს, რომელიც შეიძლება იქნას გამოყენებული X სიმრავლიდან საუკეთესო x^* ელემენტების საპოვნელად, $f \in F$ კრიტერიუმების გათვალისწინებით.

ოპტიმიზაციის ამოცანებისთვის აქტუალურია სამი ძირითადი ცნება:

- „ამონახსნი კანდიდატი“ (Solution Candidate) – x კანდიდატი, რომელიც წარმოადგენს კონკრეტული საპრობლემო არის პოტენციურ ამონახსნს X სიმრავლიდან, გარკვეული კრიტერიუმების გათვალისწინებით.
- „ამონახსნთა სიმრავლე“ (Solution Space) – S სიმრავლე, რომელიც წარმოადგენს პოტენციურ ამონახსნთა სიმრავლეს: $X^* \subseteq S \subseteq X$.
- „ფიტნეს ფუნქცია“ (Fitness Function) - ფუნქცია, რომლის მნიშვნელობა $v(x) \in V$ ყველა x ელემენტისთვის S საპრობლემო სიმრავლიდან, არის გადაწყვეტილების გამოსადეგობა, ან პრიორიტეტი შემდეგი ბიჯებისთვის, ოპტიმიზაციის პროცესში.

ოპტიმიზაციის მეთოდები აქტუალურია ხელოვნურ ინტელექტში, ბიოინსპირირებულ ევოლუციურ და კოლექტიური ინტელექტის მიდგომებში, გამოყენების სფეროები კი - შეუზღუდავი, ისეთი როგორიცაა: თამაშის თეორია, ინჟინერია, სოციალური მეცნიერებები, ზუსტი თუ საბუნებისმეტყველო დარგები და სხვა [59].

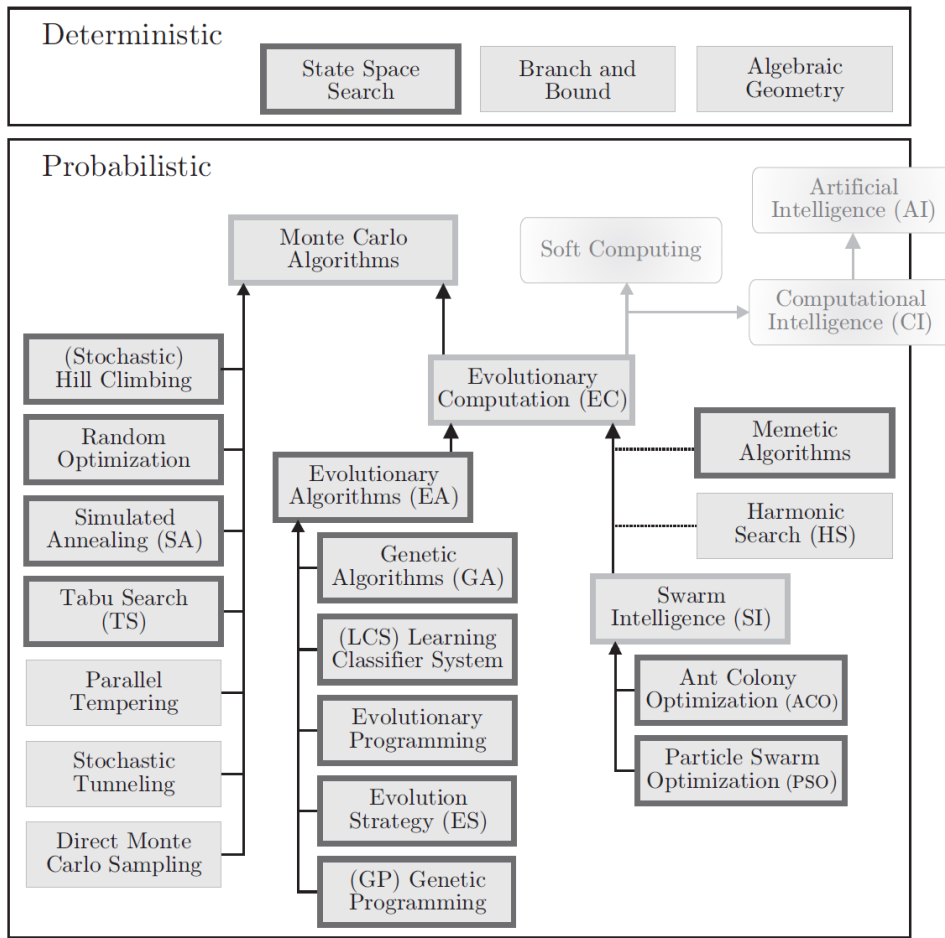
3.1.1 ოპტიმიზაციის ალგორითმების კლასიფიკაცია

ზოგადად, ოპტიმიზაციის ალგორითმები იყოფა ორ ძირითად კლასად: „დეტერმინისტულ“ (Deterministic) და „ალბათურ“ (Probabilistic) ალგორითმებად (სურათი 8).

დეტერმინისტული ალგორითმები გამოიყენება იმ შემთხვევაში, როდესაც მოცემული პრობლემისთვის ნათელი კავშირი არსებობს მის მახასიათებლებსა და შესაძლო ამონახსნებს შორის. ამ შემთხვევაში, საძიებო არე შეიძლება ადვილად იქნას შესწავლილი, მაგალითად, „დაყავი და იბატონე“-ს (divide and conquer) პრინციპით. მაგრამ, იმ შემთხვევაში, თუ საძიებო არე არის კომპლექსური და მრავალგანზომილებიანი, ან ამონახსნ კანდიდატსა და მის ფიტნეს ფუნქციას შორის არ არსებობს ნათელი კავშირი, ან ეს კავშირი ძალიან რთულია, ამ პრობლემის დეტერმინისტული ალგორითმით გადაჭრაც რთულდება ან შეუძლებელი ხდება, რადგან ყველაზე მარტივ შემთხვევაშიც კი, საჭიროა ამოუწურავი გამოთვლებისა და იტერაციების რაოდენობა.

ასეთ შემთხვევებში გამოიყენება **ალბათური** ალგორითმები. ალბათური ალგორითმების ერთ-ერთ გამორჩეულ ოჯახს წარმოადგენს ე.წ. „მონტე კარლო“-ზე (Monte Carlo) დაფუძნებული მიდგომები. ამ ალგორითმების მთავარი დადებითი მხარე ის არის, რომ მათ შეუძლიათ იპოვონ ამონახსნი დროის მოკლე შუალედში. შესაძლოა ეს ამონახსნი არ იყოს გლობალური ოპტიმუმი, თუმცა იქნება მასთან მიახლოებული.

ალგორითმში ევრისტიკული მიდგომები გამოიყენება იმის დასადგენად, თუ შესაძლო ამონახსნთა სიმრავლეებიდან რომელი უნდა გადამოწმდეს შემდეგ ეტაპზე. ევრისტიკა არის ოპტიმიზაციის ალგორითმების ნაწილი, რომელიც იყენებს ალგორითმის მიერ ახლად შეგროვებულ ინფორმაციას, რათა დაეხმაროს მას შემდეგი ეტაპის გადაწყვეტილების მიღებაში, ანუ დაადგინოს, თუ რომელი კანდიდატები უნდა იქნას განხილული, ან შექმნილი შემდეგ ეტაპზე. ერთის მხრივ, დეტერმინისტული ალგორითმები იყენებენ ევრისტიკულ მიდგომებს, რათა განსაზღვრონ ამონახსნ კანდიდატთა დამუშავების თანმიმდევრობა. მეორეს მხრივ, კი ალბათურმა ალგორითმებმა საძიებო არის შემდეგი გამოთვლებისათვის შეიძლება გამოიყენონ მხოლოდ ის კანდიდატები, რომლებიც აირჩა ევრისტიკული მეთოდების გამოყენებით.



სურათი 8. ოპტიმიზაციის ალგორითმების კლასიფიკაცია

ალგორითმისთვის, გარდა ზოგადი მიდგომისა და სტრუქტურისა, უმნიშვნელოვანესი მახასიათებელია სიზუსტე და შესრულების დრო (სისწრაფე). სიზუსტე და სისწრაფე, თავის მხრივ, ურთიერთ საწინააღმდეგო თვისებებია, რადგან ზოგადად ალბათური ოპტიმიზაციის ალგორითმებში სიზუსტის გაზრდა შეიძლება შესრულების დროის გაზრდის ხარჯზე. ასეთი პრობლემების მინიმიზაციისთვის მკვლევარები ცდილობენ ე.წ. „პარეტოს საზღვრის“ (Pareto frontier) მიდგომის დანერგვას.

ოპტიმიზაციის ალგორითმები დროითი შეზღუდვებით იყოფა ორ, ონლაინ და ოფლაინ ჯგუფად:

ონლაინ ოპტიმიზაციის ამოცანები საჭიროა გადაიჭრას სწრაფად, რამდენიმე მილიწამიდან რამდენიმე წუთამდე ინტერვალში. იმისთვის, რომ ამონახსნის პოვნა მოხერხდეს ასეთ მცირე დროში, უნდა დაბალანსდეს ოპტიმალურობა და სისწრაფე. ონლაინ ოპტიმიზაციის მაგალითებია

რობოტების ლოკალიზაციის სინქრონიზაცია (robot localization synchronization), დატვირთვის გადანაწილება ქსელებში (load balancing), სერვისების კომპოზიცია ბიზნეს ამოცანებში და სხვა.

ოფლაინ ოპტიმიზაციის ამოცანებში დრო არც ისე მნიშვნელოვანია, რადგან მომხმარებელს შეუძლია დაიცადოს თუნდაც რამდენიმე დღე. მთავარია, მან მიიღოს ოპტიმალური, ან მასთან მიახლოებული შედეგი. ოფლაინ ოპტიმიზაციის მაგალითებია, მონაცემთა შეგროვება (data mining), გრძელვადიანი გეგმის შედგენა მაგ. სატრანსპორტო კომპანიებისთვის და სხვა. ამ შემთხვევაში, დროის ფაქტორი უმნიშვნელოა იმდენად, რამდენადაც ასეთი გამოთვლები კეთდება იშვიათად ხანგრძლივი პერიოდის განმავლობაში [59].

3.1.2 ერთი და მრავალმიზნობრივი ოპტიმიზაცია

იმის მიხედვით თუ რა მიზანია დასახული მოცემული პრობლემური არისთვის (ანუ, რას ვეძებთ), ოპტიმიზაციის ალგორითმებს ყოფენ ორ ჯგუფად - ერთ და მრავალ მიზნობრივ ალგორითმებად.

ერთმიზნობრივი ოპტიმიზაცია (Single Objective Optimization).

ერთმიზნობრივი ოპტიმიზაციის დროს, კრიტერიუმი f წარმოადგენს სამიზნო არის ოპტიმუმს, მინიმუმს, ან მაქსიმუმს, იმის მიხედვით, თუ რას ვეძებთ. მაგალითად, პროდუქციის წარმოებისას, წარმოების დროის მინიმიზაციაა საჭირო. მეორეს მხრივ, კი შემოსავლების მაქსიმიზაციისთვის, მუშა ხელის აყვანა და რეკლამის გაკეთება. ოპტიმიზაციის ამოცანები, როგორც წესი, განმარტებულია მინიმიზაციის კუთხით, ხოლო იმ შემთხვევაში თუ საჭიროა f კრიტერიუმის მაქსიმიზაცია, ხდება ამ კრიტერიუმის უარყოფის ($-f$) მინიმიზაცია. სურათი 9 - ზე მოცემულია f ფუნქციის გამოსახულება ორ განზომილებიან სივრცეში $X = (X_1, X_2)$. სურათზე გამოსახულია ლოკალური და გლობალური ოპტიმუმები.

გლობალური ოპტიმუმი წარმოადგენს მთელი X არის ოპტიმუმს, ხოლო ლოკალური - მისი ქვე-არის ოპტიმუმს.

ლოკალური მაქსიმუმი (Local Maximum) $\hat{x}_l \in X$ მიზნობრივი ფუნქციისთვის $f : X \rightarrow R$ არის შემავალი ელემენტები, სადაც $f(\hat{x}_l) \geq f(x)$, ყველა \hat{x}_l - ის მეზობელი x - თვის. თუ $X \subseteq R^n$, მაშინ:

$$\forall \hat{x}_l \exists \varepsilon > 0 : f(\hat{x}_l) \geq f(x) \forall x \in X, |x - \hat{x}_l| < \varepsilon \quad (5)$$

ლოკალური მინიმუმი (Local Minimum) $\check{x}_l \in X$ მიზნობრივი ფუნქციისთვის $f : X \rightarrow R$ არის შემავალი ელემენტები, სადაც $f(\check{x}_l) \leq f(x)$, ყველა \check{x}_l - ის მეზობელი x - თვის. თუ $X \subseteq R^n$, მაშინ:

$$\forall \check{x}_l \exists \varepsilon > 0 : f(\check{x}_l) \leq f(x) \forall x \in X, |x - \check{x}_l| < \varepsilon \quad (6)$$

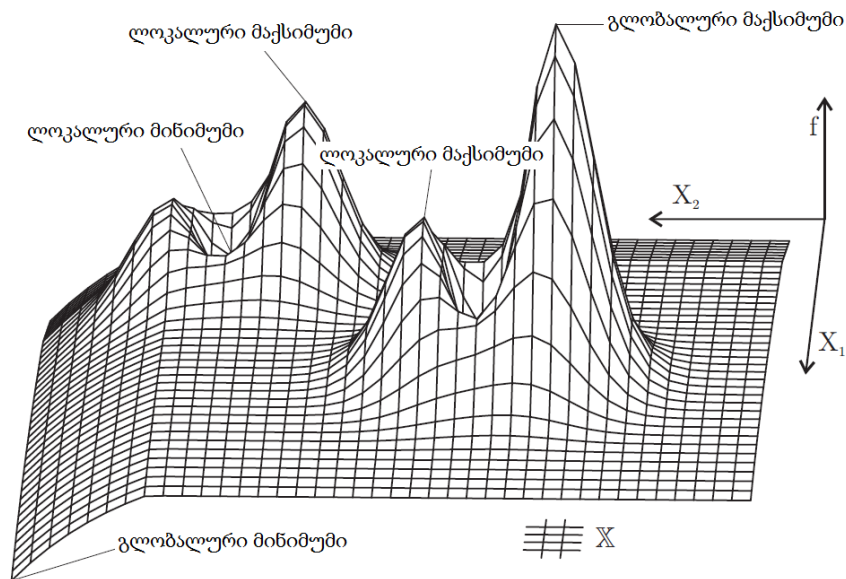
ლოკალური ოპტიმუმი (Local Optimum) $x_l^* \in X$ მიზნობრივი ფუნქციისთვის $f : X \rightarrow R$ წარმოადგენს ან ლოკალურ მაქსიმუმს, ან ლოკალურ მინიმუმს.

გლობალური მაქსიმუმი (Global Maximum) $\hat{x} \in x$ მიზნობრივი ფუნქციისთვის $f : X \rightarrow R$ არის შემავალი ელემენტები, სადაც $f(\hat{x}) \geq f(x) \forall x \in X$.

გლობალური მინიმუმი (Global Minimum) $\check{x} \in x$ მიზნობრივი ფუნქციისთვის $f : X \rightarrow R$ არის შემავალი ელემენტები, სადაც $f(\check{x}) \leq f(x) \forall x \in X$.

გლობალური ოპტიმუმი (Global Optimum) $x^* \in X$ მიზნობრივი ფუნქციისთვის $f : X \rightarrow R$ წარმოადგენს ან გლობალურ მაქსიმუმს, ან გლობალურ მინიმუმს.

ოპტიმუმების სიმრავლე (Optimal Set) X^* წარმოადგენს ყველა ოპტიმალური ელემენტისგან შედგენილ სიმრავლეს.



სურათი 9. გლობალური და ლოკალური ოპტიმუმები

მოცემულ X არეში, ერთ განზომილებიან ფუნქციას $f : X = R \rightarrow R$ შეიძლება ქონდეს ერთზე მეტი, როგორც გლობალური მაქსიმუმი, ასევე გლობალური მინიმუმი. ამ შემთხვევაში, სწორი ამონახსნი იქნება ყველა ოპტიმალური ელემენტების X^* სიმრავლე X სიმრავლიდან და არა ერთი მაქსიმუმი, ან მინიმუმი. აქედან გამომდინარე, ოპტიმუმის ზუსტი მნიშვნელობა დამოკიდებული პრობლემური არის სპეციფიკაზე.

მრავალმიზნობრივი ოპტიმიზაცია (Multiple Objective Optimization).

გლობალური ოპტიმიზაციის ამოცანა მხოლოდ ერთი მინიმუმის ან მაქსიმუმის პოვნაში არ მდგომარეობს. დაგეგმვის და გადაწყვეტილების მიღების მრავალ პრაქტიკულ ამოცანებში განისაზღვრება f_i კრიტერიუმების F სიმრავლე, რომელიც უნდა დაოპტიმიზირდეს.

$$F = \{f_i : X \rightarrow Y_i : 0 < i \leq n, Y_i \subseteq R\} \quad (7)$$

მრავალმიზნობრივი ოპტიმიზაცია ორიენტირებულია კონფლიქტური მიზნების აღმოფხვრისკენ. ზემოთ ნახსენებ პროდუქციის წარმოების მაგალითში, ასეთ მრავალმიზნობრივ ფუნქციას წარმოადგენს [59, 60]:

- დროის მინიმიზაცია შემოსულ შეკვეთასა და პროდუქტის მიწოდებას შორის.

- შემოსავლების მაქსიმიზაცია.
- რეკლამის, პერსონალის და ნედლეულზე გაწეული ხარჯების მინიმიზაცია.
- პროდუქტის ხარისხის მაქსიმიზაცია.
- გარემოზე ზემოქმედების მინიმიზაცია და სხვ.

3.1.3 პარეტოს ოპტიმიზაცია (Pareto Optimization)

უილფრედო პარეტომ (Vilfredo Pareto) შემოგვთავაზა მრავალმიზნობრივი ოპტიმიზაციის მათემატიკური აღწერა, რომელიც ეფუძნება კონფლიქტური კრიტერიუმების ოპტიმიზაციას. ის წარმოადგენს პრობლემის ამონახსნთა ზღვრებს, რომლებიც შეიძლება იქნას მიღწეული კონფლიქტური მიზნებისთვის ოპტიმალური კომპრომისული გადაწყვეტილების მიღების გზით. ამ ზღვრებიდან, გადაწყვეტილების მიღებისას, შეირჩევა ისეთი კონფიგურაცია, რომელიც ყველაზე მეტად არის ახლოს ოპტიმალურთან. ოპტიმალურობა პარეტოს მიდგომაში დამოკიდებულია დომინირების განსაზღვრაზე. ელემენტი x_1 არის დომინანტი (პრივილეგირებული) x_2 ($x_1 \vdash x_2$) ელემენტზე, თუ x_1 არის უკეთესი x_2 - ზე მინიმუმ ერთი მიზნობრივი ფუნქციით მაინც და არ არის უარესი არცერთი სხვა მიზნობრივი ფუნქციით. შესაბამისად, f მიზნობრივი ფუნქციების F სიმრავლისთვის შეიძლება დავწეროთ:

$$x_1 \vdash x_2 \Leftrightarrow \forall i : 0 < i \leq n \Rightarrow \omega_i f_i(x_1) \leq \omega_i f_i(x_2) \wedge \exists j : 0 < j \leq n : \omega_j f_j(x_1) < \omega_j f_j(x_2) \quad (8)$$

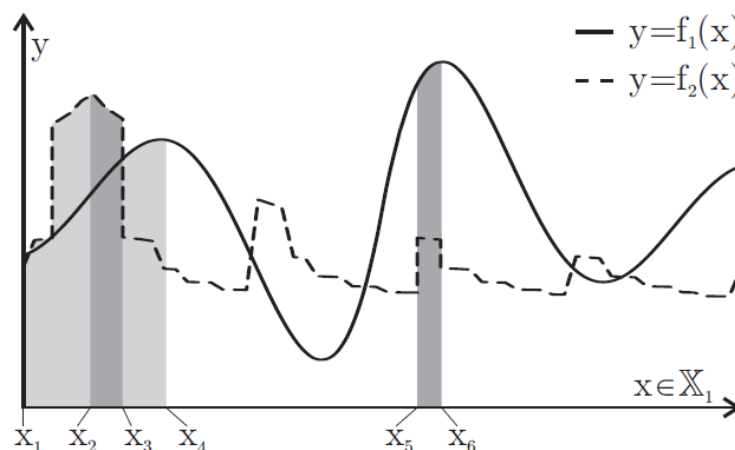
$$\omega_i = \begin{cases} 1 & \text{if } f_i - \text{minimized} \\ -1 & \text{if } f_i - \text{maximized} \end{cases} \quad (9)$$

ელემენტი $x^* \in R$ არის პარეტოს ოპტიმუმი და შესაბამისად, შედის X^* სიმრავლეში, მაშინ და მხოლოდ მაშინ, თუ მასზე არ დომინირებს სხვა რომელიმე ელემენტი საპრობლემო X - სფეროდან. პარეტოს ოპტიმიზაციის ტერმინებით, X^* წარმოადგენს პარეტოს სიმრავლეს ან პარეტოს ზღვარს (Pareto Frontier).

$$x^* \in X^* \Leftrightarrow \exists x \in X : x \vdash x^* \quad (10)$$

სურათი 10-ზე ნაჩვენებია ოპტიმიზაცია პარეტოს ზღვრის მიდგომით. f_1 და f_2 ფუნქციებისთვის ხდება მაქსიმიზაცია, შესაბამისად, $\omega_1 = \omega_2 = 1$. არეალი, რომელიც მონიშნულია მუქი ნაცრისფერით წარმოადგენს პარეტო ოპტიმუმს და შესაბამისად, სიმრავლეს $X^* = [x_2, x_3] \cup [x_5, x_6]$, რომელიც თავის მხრივ, შეიცავს უსასრულოდ ბევრ ელემენტს. სხვა დანარჩენ წერტილებზე ხდება დომინირება, ანუ ისინი არ წარმოადგენენ ოპტიმუმებს. x_1 და x_2 წერტილები დომინირებენ დანარჩენ წერტილებზე $[x_1, x_2]$ არეში, შესაბამისად f_1 და f_2 ფუნქციები შეიძლება გავაუმჯობესოთ x - ის გაზრდით. თუ, დავიწყებთ ყველაზე მარცხენა წერტილიდან აღნიშნულ არეში, რომელიც შეესაბამება x_1 წერტილს და გადავინაცვლებთ მცირე Δ ბიჯით მარჯვნივ, აღმოვჩნდებით $x_1 + \Delta$ წერტილზე, რომელიც დომინირებს x_1 - ზე, რადგან $f_1(x_1 + \Delta) > f_1(x_1)$ და $f_2(x_1 + \Delta) > f_2(x_1)$. თუ, გავაგრძელებთ ამ პროცესს, ვიპოვით ახალ-ახალ დომინანტ წერტილებს, მანამ, სანამ არ მივაღწევთ x_2 წერტილს. x_2 წარმოადგენს f_2 - ის ყველაზე დიდ შესაძლო მნიშვნელობას, ანუ გლობალურ მაქსიმუმს, რომელზეც არც ერთი წერტილი X სიმრავლიდან არ დომინირებს [59].

აღნიშნული მიდგომით პარეტოს ყველა ზღვრის პოვნა შეიძლება, რომელიც საბოლოო ჯამში მიმართულია მოცემული ამოცანის კონფლიქტური პარამეტრების დაბალანსება-ოპტიმიზაციისკენ.



სურათი 10. ოპტიმიზაცია პარეტოს ზღვრის მიდგომით

3.2 სოციალური ენტროპია და სინერგია

3.2.1 ენტროპია მულტი-რობოტულ სისტემებში

ჩვენ განვიხილავთ მობილური სენსორული ჯგუფის ქცევის ხარისხის შეფასების რამდენიმე კრიტერიუმს, როგორებიცაა მოწესრიგებულობა (სინერგია) და მოუწესრიგებლობა (ენტროპია), ოპტიმიზაციის კუთხით. ეს პარამეტრები დაგვეხმარება შევაფასოთ ჯგუფის ქცევის წარმადობის ხარისხი.

სისტემის ნებისმიერი ქმედება შეიძლება აღიწეროს როგორც ერთი მდგომარეობიდან მეორეში გადასვლა, რაც მჭიდროდ არის დაკავშირებული ენტროპიის ცვლილებასთან. თერმოდინამიკურ სისტემებში ენტროპია ასოცირდება მოწესრიგებულობის, მოუწესრიგებლობის, ან/და ქაოსის რაოდენობასთან (ხარისხთან). მოუწესრიგებლობა შეიძლება წარმოვიდგინოთ, როგორც დეზორიენტირებული აგენტის ქმედების ვექტორი [35].

ამ საკითხთან მიმართებაში ჩვენ გვაინტერესებს რობოტების გროვის ინდივიდების ერთობლივი ქმედება, რომელიც უზრუნველყოფს მთლიანი სისტემის საერთო მიზნისკენ სწრაფვას. რეალურ სისტემებში დიდი მნიშვნელობა ენიჭება გროვის ზომას (რაც ფინანსურ ხარჯებთან არის დაკავშირებული), გროვის მიერ მოხმარებულ ენერგიას და დროს. აღნიშნული პარამეტრების სწორად შერჩევით შესაძლებელია დასმული ამოცანის ოპტიმალური გადაწყვეტა მივიღოთ. ენტროპიის, რიგის და საშუალო კუთხური სიჩქარის პარამეტრების შემოტანით სისტემაში, შეიძლება განისაზღვროს ჯგუფის პოზიციური განლაგება და ენერგიის მოხმარება. ენერგიის მოხმარების განსაზღვრისათვის ზოგიერთ შემთხვევაში ასევე გამოიყენება დამატებითი პარამეტრი - გადაადგილების საშუალო სიჩქარე [36].

ენტროპია (5) ზომავს გროვის პოზიციურ მოუწესრიგებლობას. იგი გამოიყენება კლასტერიზაციის კლასიკურ მიდგომებში, როგორც კლასტერიზაციის პროცესის მართვის საშუალება. იგი გამოითვლება ყველა

შესაძლო კლასტერის მოძებნის, ამ კლასტერებისთვის შენონის ინფორმაციული ენტროპიის გამოთვლის და მათი შეჯამების გზით [37].

ინფორმაციული ენტროპია $H(l)$ კლასტერისთვის მაქსიმალური დისტანციით l გამოითვლება შემდეგნაირად:

$$H(l) = - \sum_{k=1}^K P_k \log_2(P_k) \quad (11)$$

სადაც, P_k არის ინდივიდების პროპორცია k -ურ კლასტერში და K არის კლასტერების რაოდენობა მოცემული l -ისთვის.

ენტროპიის ცვლილების სიჩქარეც (dS/dt) განიხილება როგორც სისტემის ერთ-ერთი მნიშვნელოვანი პარამეტრი. საერთო ენტროპიის (S) დასადგენად ხდება ყველა ამ მნიშვნელობების ინტეგრალების გამოთვლა:

$$S = \int_0^{\infty} H(l) dl \quad (12)$$

სენსორების მოწესრიგებულობა (სინერგია) განისაზღვრება 0-1 დიაპაზონში, ფორმულით:

$$\psi(t) = \frac{1}{M} \left| \sum_{k=1}^M e^{i\theta_k} \right| \quad (13)$$

სადაც, M არის სენსორების რაოდენობა კლასტერში, ხოლო θ_k - k - ური სენსორის მიმართულება t დროში.

კიდევ ერთი მნიშვნელოვანი პარამეტრი - გროვის გეომეტრიული ცენტრის საშუალო სიჩქარე მოძრაობის მთელი ციკლის დროს, გამოითვლება:

$$\vec{V}_s(t) = \frac{1}{N} \left| \sum_{i=1}^N \vec{V}_i(t) \right| \quad (14)$$

მობილური სენსორების ჯგუფის ქმედების ხარისხის შეფასების ზემოთ აღწერილ მეთოდებზე დაყრდნობით, ჩვენ ვამტკიცებთ, რომ გროვის კლასტერის ენტროპია, როგორც მოუწესრიგებლობის (ქაოსის) ხარისხი, შესაძლებელია გამოითვალოს სენსორებიდან შეგროვებული ინფორმაციით, კერძოდ, შეფარდებითი პოზიციებით ან საშუალო კუთხური სიჩქარით. ჩვენ

შემოვიტანეთ ენტროპიის კონცეფცია, როგორც შიდა ქცევითი შეუთავსებლობა ან ანტაგონიზმი ზოგიერთ დისორიენტირებული კომპონენტების ქმედით ვექტორებს შორის. შესაბამისად, რობოტების გროვის ქმედების სიდიდეები (პარამეტრები), რომლებიც შეიცავენ დისორიენტირებული რობოტების ქმედების ვექტორებს შეიძლება მივიღოთ ქვემოთ მოყვანილი წესის შესაბამისად. ეს მიდგომა დაფუძნებულია მობილური რობოტებისთვის სიჩქარის ვექტორების $\vec{V}_i(t)$ ალგებრულ ჯამზე t მომენტში. ენტროპია მთლიანი ჯგუფისთვის t დროში კი იზომება [38]:

$$H(t) = \sum_{k=1}^K P_k \log_2(P_k(t)) \quad (15)$$

სადაც:

$$P_k(t) = \frac{\sum_i^N \vec{V}_i(t)}{\sum_i^N |\vec{V}_i(t)|} \quad (16)$$

3.2.2 სინერგია

ყველა სისტემა ხასიათდება სტრუქტურით, კომპოზიციით და მდგომარეობით. სისტემის მდგომარეობა აღიწერება შეუთავსებლობის სხვადასხვა ხარისხით. სოციალური პროცესების ნებისმიერი მაშტაბირება ხასიათდება ძლიერი კონფრონტაციული ფონით, რაც ხშირად მიყვავართ დისბალანსის კრიტიკულ ზღვრამდე. ეგრეთ წოდებული ძლიერი სოციალური კლასტერი, რომელიც ცდილობს გაფართოვდეს სუსტი ჰომეოსტაზის მქონე კლასტერების შევიწროვებით (ჩაგვრით) და მიიღტვის ლიდერობის მოპოვებისკენ, იწვევს კლასტერის დისბალანსს. თავის მხრივ, პატარა კლასტერები ცდილობენ მოძებნონ გარე მხარდაჭერა, სოციალური ჰეტეროსტაზი, საკუთარი ჰომეოსტაზის გასაძლიერებლად.

ზემოთ აღნიშნულიდან გამომდინარე შესაძლებელია აიგოს ისეთი სისტემა, რომელიც დაფუძნებულია ტვინის მოდელის მუშაობის პრინციპზე და შედგება ნერვული უჯრედებისგან, რომლებიც აქტიურდებიან

სუბიექტის ადგილმდებარეობის მიხედვით. ასეთი მოდელი შედგება ორი ნეირონული ფენისგან (დონისგან), შემავალი და გამომავალი, და ასევე, როგორც წესი, ორგანოზომილებიანი ნეირონული ცხრილისგან, რომელიც წარმოადგენს კლასტერის ტოპოლოგიურ მოდელს. ყველა ასეთი სისტემა, მათ შორის მულტი-აგენტური სისტემებიც, შეიძლება განვიხილოთ როგორც ე.წ. „სინერგიული გრაფი“, სადაც აგენტები ასოცირდება ნეირონებთან. სხვა სიტყვებით რომ ვთქვათ, სისტემა წარმოადგენს აქსონ-დენდრიტულ მოდელს, სინაპსური კავშირებით. სინაპსების თავსებადობა წარმოადგენს გრაფის მთლიანობის აუცილებელ პირობას. სისტემის ქმედება განისაზღვრება გარე და შიდა ფაქტორების თავისუფლების ხარისხით. სისტემის სტაბილურობა ან ჰომეოსტაზი დროის მოცემულ მომენტში განისაზღვრება სინერგიისა და ენტროპიის სხვაობით:

$$M_h = S - H \quad (17)$$

თითოეული სინაპსი, ან ნებისმიერ ორ, ან მეტ კლასტერს შორის ინტერაქცია რეკურსიულად აფორმირებს ახალ ობიექტს - ახალ გაერთიანებულ კლასტერს, სადაც ერთობლივი ძალისხმევით ხდება სინერგია-ენტროპიის, ბალანსის და ჰომეოსტაზის ფორმირება. კლასტერის ფორმირება ხდება, როდესაც ენტროპია გადადის სინერგიაში და პირიქით, სინერგიის დანაწევრება ზრდის ენტროპიას. ზოგადად, კლასტერიზაციის პროცესი მიმდინარეობს შემდეგი თანმიმდევრობით:

Confrontation → Cooperation → Consolidation.

1. კონფრონტაცია გამოწვეულია სუბიექტებს შორის ინტერესთა კონფლიქტით:

$$\text{synergy} < \text{entropy and } \sum_{i=1}^n \mu_i > 0 \quad (18)$$

2. კოოპერაცია, იგივე თანამშრომლობა (პეტეროსტაზისის დაბალი ხარისხი) გამოწვეულია სუბიექტებს შორის ინტერესების დამთხვევით და მხოლოდ შიდა ანტაგონიზმით:

$$synergy > entropy \text{ and } \sum_{i=1}^n \mu_i > 0 \quad (19)$$

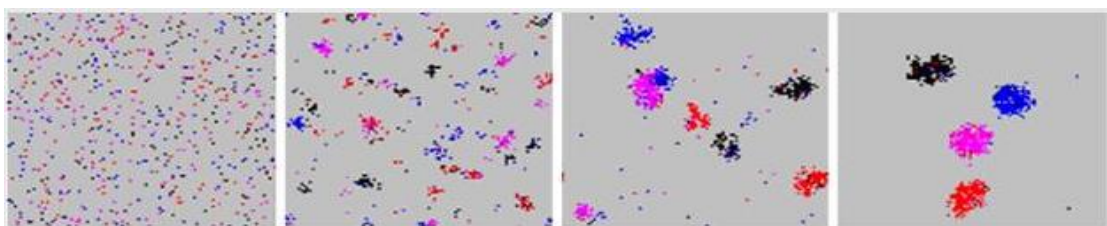
3. კონსოლიდაცია, იგივე ჰარმონიული თანაცხოვრება (ჰეტეროსტაზისის მაღალი ხარისხი) გამოწვეულია სუბიექტებს შორის ინტერესების დამთხვევით და შიდა ანტაგონიზმის დაძლევით. ეს არის საზოგადოების იდეალური მდგომარეობა:

$$synergy > entropy \text{ and } \sum_{i=1}^n \mu_i = 0 \quad (20)$$

განხილულ საკითხში წარმოდგენილია დესტრუქციული და კონსტრუქციული ურთიერთქმედების ფორმები, მათი ცვლილების და ერთმანეთში გადასვლის კრიტერიუმები. ზოგადად, ნებისმიერი სინერგიული ქმედების არსი იმაში მდგომარეობს, რომ ყველა მონაწილე მხარისთვის იყოს მომგებიანი. ყოველ კონკრეტულ სიტუაციაში ოპტიმიზაცია მიიღწევა სტაბილურობის მაქსიმიზაციის და არამკაცრი ენტროპიის მინიმიზაციის გზით [39].

3.3 კლასტერიზაციის მიმოხილვა

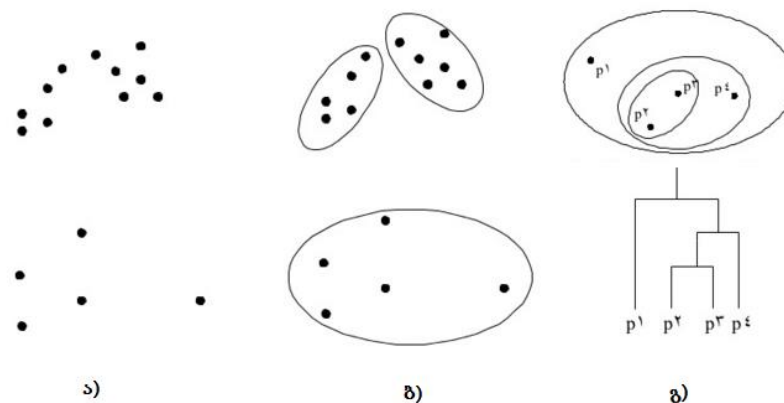
კლასტერიზაცია არის მონაცემების დაჯგუფება (დაყოფა) ობიექტების მსგავსობის მიხედვით, რომლის ევოლუციური პროცესი წარმოდგენილია სურათი 11-ზე. თითოეული ჯგუფი, ანუ კლასტერი, შეიცავს მსგავს ობიექტებს ჯგუფის შიგნით და სხვა ჯგუფებისგან განსხვავებულ ობიექტებს.



სურათი 11. კლასტერიზაციის ევოლუციური პროცესი

კლასტერული ანალიზის როლი მნიშვნელოვანია სხვადასხვა სფეროში, ინჟინერია (მანქანური სწავლება, ხელოვნური ინტელექტი, გამოსახულების ამოცნობა), კომპიუტერული მეცნიერებები (ინფორმაციის ძიება ინტერნეტში, ტექსტური და გრაფიკული მონაცემების მოძიება-ფრაგმენტაცია), მედიცინა, საბუნებისმეტყველო და სოციალური მეცნიერებები. ასევე აქტუალურია სტატისტიკაში, გრაფთა თეორიაში, ხელოვნურ ნეირონულ ქსელებში, ევოლუციურ გამოთვლებსა და სხვა ოპტიმიზაციის ამოცანებში [40].

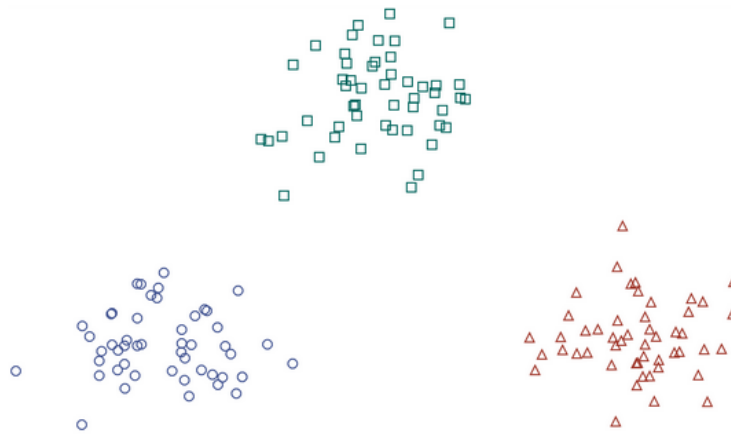
კლასტერიზაციის ორი მიდგომა არსებობს: ექსკლუზიური და იერარქიული. ექსკლუზიური დაყოფის დროს ერთი ობიექტი ეკუთვნის მხოლოდ კლასტერს (სურათი 12 (ბ)), ხოლო იერარქიული მიდგომის დროს - ერთმანეთში ჩადგმული კლასტერები ქმნიან იერარქიულ ხეს (სურათი 12 (გ)) [41, 42].



სურათი 12. ექსკლუზიური და იერარქიული კლასტერიზაცია

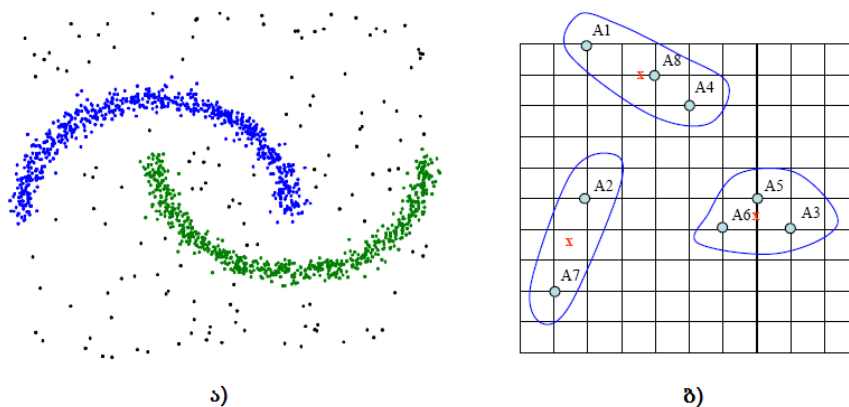
კლასტერული ანალიზის ამოცანებში, გარდა ამ ორი ძირითადი მიმართულებისა არსებობს კიდევ რამდენიმე ტიპი, რომელთა შერჩევა და მოკიდებულია კონკრეტული ამოცანის ან კონკრეტული მონაცემთა მასივების თვისებებზე (ზომა, ფორმა, მსგავსება, განლაგება და ა.შ.). ქვემოთ, მოკლედ მიმოვიხილავთ რამდენიმე მათგანს.

ე.წ. „კარგად დაყოფილი“ (well-separated): თითოეული წერტილი კლასტერის შიგნით უფრო ახლოსაა ერთმანეთთან, ვიდრე სხვა კლასტერის რომელიმე წერტილთან (სურათი 13).



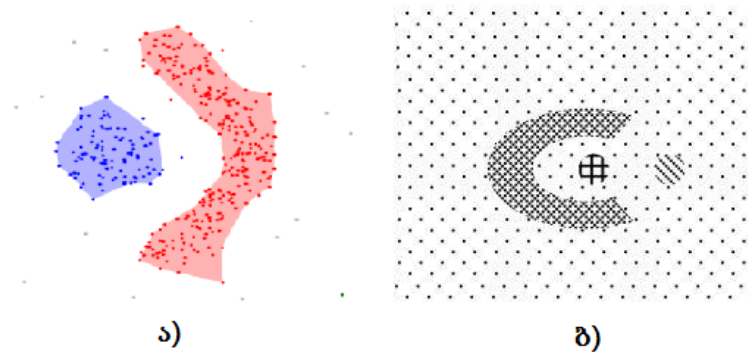
სურათი 13. კლასტერიზაციის ტიპი - „კარგად დაყოფილი“

ე.წ. „პროტოტიპზე-დაფუძნებული“ (prototype-based): კლასტერში შემავალი ობიექტები არის უფრო მსგავსი კლასტერის შიგნით შერჩეულ პროტოტიპთან, ვიდრე სხვა კლასტერების პროტოტიპებთან (სურათი 14 (ა)). მაგალითად, უწყვეტი მონაცემთა მასივებისთვის პროტოტიპი არის ცენტროიდი (ცენტრი, ანუ ყველა წერტილის საშუალო მნიშვნელობა) (სურათი 14 (ბ)).



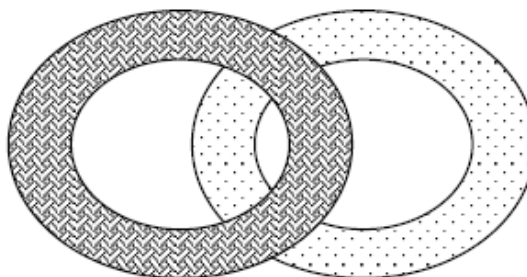
სურათი 14. კლასტერიზაციის ტიპი - „პროტოტიპზე-დაფუძნებული“

ე.წ. „სიმკვრივეზე-დაფუძნებული“ (density-based): კლასტერს წარმოადგენს მაღალი სიმკვრივით გამოყოფილი რეგიონი, რომელსაც გარს არტყია დაბალი სიმკვრივის ზონა (სურათი 15 (ა)). ასეთი განსაზღვრება გამოიყენება მაშინ, როცა ინფორმაციაში შეიმჩნევა ხმაური და გამოფრქვევები (სურათი 15 (ბ)).



სურათი 15. კლასტერიზაციის ტიპი - „სიმკვრივე-დაფუძნებული“

ე.წ. „კონცეპტუალური“ (conceptual): კლასტერი შედგება ობიექტების ჯგუფისგან, რომლებსაც აქვთ საერთო თვისებები. მათ თანაკვეთაზე არსებულ წერტილებს გააჩნიათ ორივე კლასტერისთვის საერთო თვისებები (სურათი 16). კლასტერიზაციის ეს მიდგომა მოიცავს ყველა ზემოთ აღწერილ მიდგომებსაც.



სურათი 16. კლასტერიზაციის ტიპი - „კონცეპტუალური“

სადისერტაციო ნაშრომში კლასტერიზაციას უდიდესი როლი ენიჭება, რადგან სწორედ მისი გამოყენებით ვახდენთ მულტი-რობოტული სისტემის ეფექტურ მართვას. კერძოდ, კლასტერებად იყოფა საძიებო არე და მიება ხდება კლასტერების შიგნით და არა მთელს არეალზე, რაც, თავის მხრივ, სისწრაფესა და ეფექტურობას მათებს ჩვენს ალგორითმს. ამისათვის, ჩვენ განვიხილეთ კლასტერიზაციის რამდენიმე მეთოდი, კერძოდ, ვორონოის დიაგრამა, კოჰონენის თვითორგანიზებადი რუკა და K-Means ალგორითმი. გავაკეთეთ შედარებითი ანალიზი და ავირჩიეთ K-Means-ი, რადგან აღნიშნულ ალგორითმებთან შედარებით გააჩნია შემდეგი უპირატესობები:

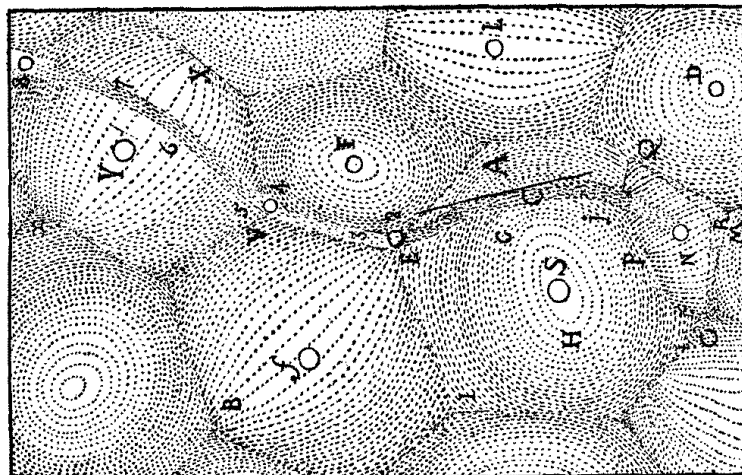
- მარტივი და ადგილად რეალიზებადია

- სწრაფად მუშაობს მცირე განზომილებიან მონაცემთა სტრუქტურებთან
- არ საჭიროებს კლასტერების რაოდენობის წინასწარ განსაზღვრას
- სტაბილური და ადვილად გაფართოებადია

3.3.1 ვორონოის დიაგრამა (Voronoi Diagram)

ვორონოის დიაგრამა (Voronoi diagram) არის სიბრტყის რეგიონებად დაყოფის, ანუ კლასტერიზაციის ერთ-ერთი მეთოდი, რომელიც დაფუძნებულია წერტილებს შორის მანძილზე კონკრეტულ რეგიონში (ქვესიბრტყეში). ამ მეთოდში თითოეული რეგიონი შეესაბამება ერთ წერტილს მოცემულ სიბრტყეზე, რომელსაც მთავარ წერტილს, „მარცვალს“ (seed) ან „გენერატორს“ (generator) უწოდებენ. ყველა წერტილი რეგიონის შიგნით არის უფრო ახლოს ამავე რეგიონის მთავარ წერტილთან (გენერატორთან) ვიდრე სხვა რეგიონის გენერატორთან. წერტილები რომლებიც თანაბრად დაშორებულნი არიან ორი გენერატორიდან ქმნიან სეგმენტებს (segment), ხოლო სამი გენერატორიდან კვანძებს (node) [43, 44].

იდეა წამოვიდა მე - 17 საუკუნიდან. რენე დეკარტი (René Descartes) თავის წიგნში „ფილოსოფიის პრინციპები“ ამტკიცებს, რომ მზის სისტემა შედგება მორევებისგან, რომლებიც თავის მხრივ შედგებიან ერთი უძრავი ვარსკვლავის გარშემო მოძრავი მატერიისგან (სურათი 17).



სურათი 17. დეკარტის სივრცის დეკომპოზიცია

მოგვიანებით, ეს კონცეფცია, განვითარდა და დაამტკიცა თავისი სარგებლიანობა მეცნიერების სხვადასხვა სფეროში. ამ მეთოდებს ვიცნობთ სხვადასხვა სახელით, იმის მიხედვით თუ რომელ სფეროში და რა სპეციფიკური საქმისთვის გამოიყენება. მაგალითად, შუათანა ღერძების ტრანსფორმაცია (ბიოლოგია, ფიზიოლოგია), ვინგერ-ზეიტცის ზონები (ქიმია, ფიზიკაში), კრისტალოგრაფიაში ქმედების არეები და ტისენის მრავალკუთხედი (მეთოდოლოგია, გეოგრაფია).

მათემატიკაში კონცეფციის პირველი წარმომდგენები იყვნენ ვორონოი (Voronoi) და დირიხლე (Dirichlet). ისინი ამ კონცეფციას იყენებდნენ კვადრატული ფორმების შესწავლაში. დღესდღეობით ეს მეთოდები ცნობილია როგორც ვორონოის დიაგრამა (Voronoi diagram) ან დირიხლეს ტესელაცია (Dirichlet tessellation). ვორონოი იყო პირველი ვინც აღმოაჩინა ამ სტრუქტურის დუალიზმი - ნებისმიერი ორ, დაკავშირებულ გენერატორის რეგიონებს აქვთ საერთო საზღვარი. მოგვიანებით, დელონე (Delaunay) მივიდა იგივე შედეგამდე - ორი გენერატორი შეიძლება დაკავშირებული იყოს ერთმანეთთან მაშინ და მხოლოდ მაშინ თუ ისინი განთავსებულია წრეზე, რომელიც არ შეიცავს არც ერთ წერტილს S გენერატორების სიმრავლიდან. სხვაგვარად, ვორონოის დიაგრამის დუალიზმს ჰქვია დელონეს ტრიანგულაცია (Delaunay triangulation) [45].

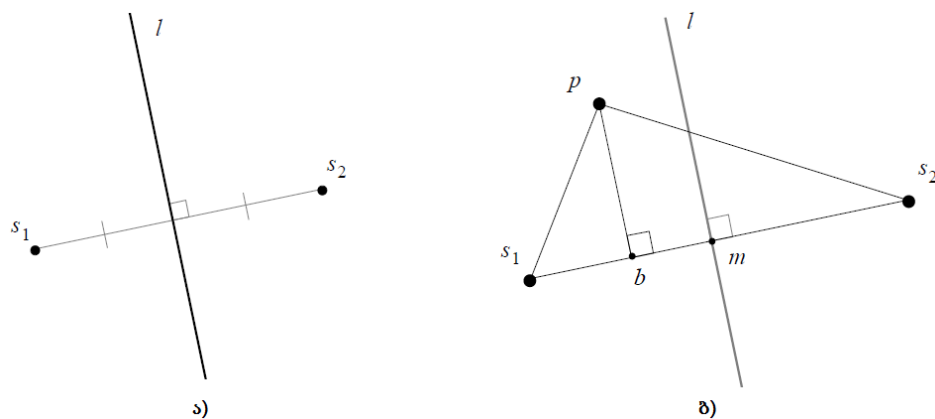
3.3.1.1 მარტივი მოდელი

ვორონოის დიაგრამას იყენებენ შემდეგ ოპტიმიზაციის ამოცანებში: უახლოესი წყვილი (Closest Pair), ყველა უახლოესი მეზობელი (All Nearest Neighbors), ევკლიდეს მინიმალური გადამფარავი ხე (Euclidean Minimum Spanning Tree), ტრიანგულაცია (Triangulation) და უახლოესი მეზობლის ძიება (Nearest Neighbor Search). ზოგადად ამოცანა იმაში მდგომარეობს რომ, სიბრტყეზე მოცემულია n წერტილების S სიმრავლე და საჭიროა თითოეული s წერტილი მივაკუთვნოთ რეგიონს, რომელშიც სიბრტყის ყველა წერტილი უფრო ახლოსაა s - თან ვიდრე სხვა ნებისმიერ s' -თან S სიმრავლიდან.

$$Vor(s) = \{p: distance(s, p) \leq distance(s', p), \forall s' \in S\} \quad (21)$$

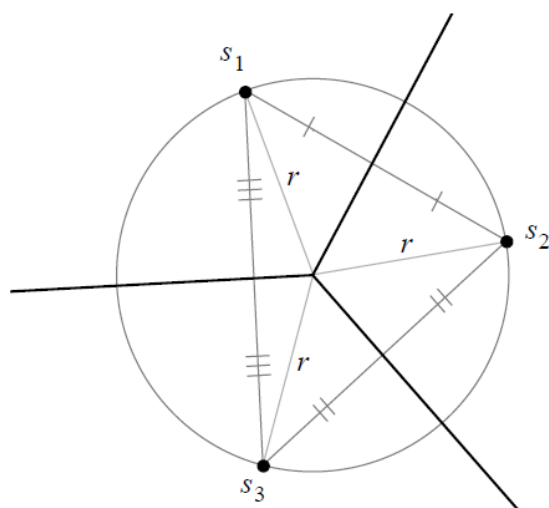
სადაც $Vor(s)$ არის ვორონოის რეგიონი s წერტილისთვის.

თუ S სიმრავლე შეიცავს ერთ წერტილს, მაშინ ვორონოის დიაგრამა ამ წერტილისთვის იქნება მთელი სიბრტყე. თუ სიმრავლე შეიცავს ორ წერტილს $S = \{s_1, s_2\}$, მაშინ სიბრტყე l წრფით ორ ნახევარსიბრტყედ გაიყოფა, სადაც l არის $\overline{s_1 s_2}$ – ის ბისექტრისა (სურათი 18 (ა)). აღსანიშნავია რომ მიღებული ნახევარსიბრტყეები გადაფარვა ხდება მხოლოდ იმ წერტილებში რომლებიც ერთნაირი მანძილით არიან დაშორებული s_1 და s_2 წერტილებიდან ანუ მდებარეობენ l წრფეზე. ყველა წერტილი რომელიც მდებარეობს s_1 წერტილის შესაბამის ნახევარსიბრტყეზე არის უფრო ახლოს s_1 წერტილთან ვიდრე s_2 წერტილთან (სურათი 18 (ბ)).



სურათი 18. ვორონოის დიაგრამა ორი წერტილით

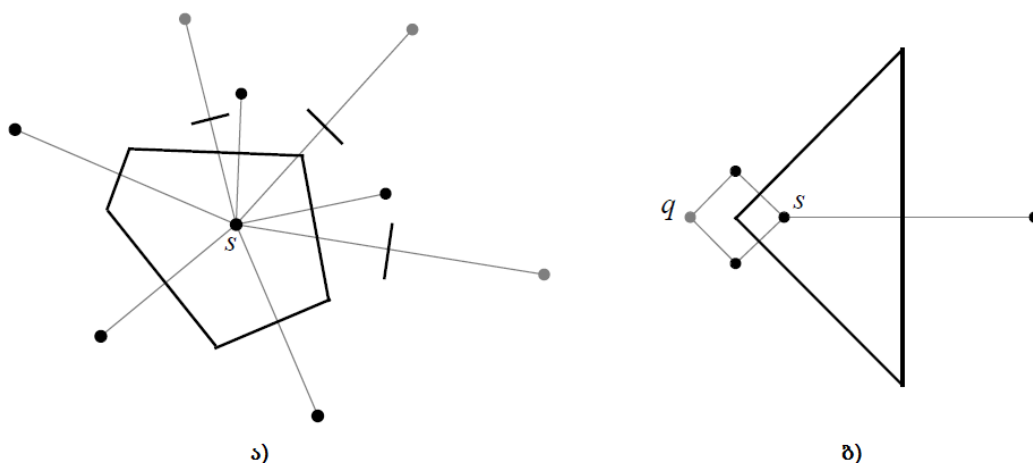
სამი წერტილისგან შემდგარ სიმრავლეზე ვორონოის დიაგრამა აიგება შემდეგნაირად: სიმრავლეში შემავალი s წერტილების ყოველი წყვილი უნდა შეერთდეს ერთმანეთთან, რის შემდეგადაც მივიღებთ სამკუთხედს $\Delta s_1 s_2 s_3$. შემდეგ სამკუთხედის თითოეული გვერდისთვის უნდა გავავლოთ ბისექტრისები. ეს ბისექტრისები ერთმანეთს გადაკვეთენ ერთ წერტილში, რადგანაც სიბრტყეზე სამი წერტილი განსაზღვრავს წრეწირს, მიღებული თანაკვეთის წერტილი იქნება ამ წრეწირის ცენტრი. ბისექტრისებით და მათი თანაკვეთის წერტილით შემოსაზღვრული ნახევარსიბრტყეები წარმოადგენენ ვორონოის რეგიონებს. (სურათი 19).



სურათი 19. ვორონოის დიაგრამა სამი წერტილით

ვორონოის რეგიონის საზღვრებს ვორონოის წიბოები ეწოდებათ, ხოლო ამ წიბოების დაბოლოებებს ვორონოის წვეროები [46].

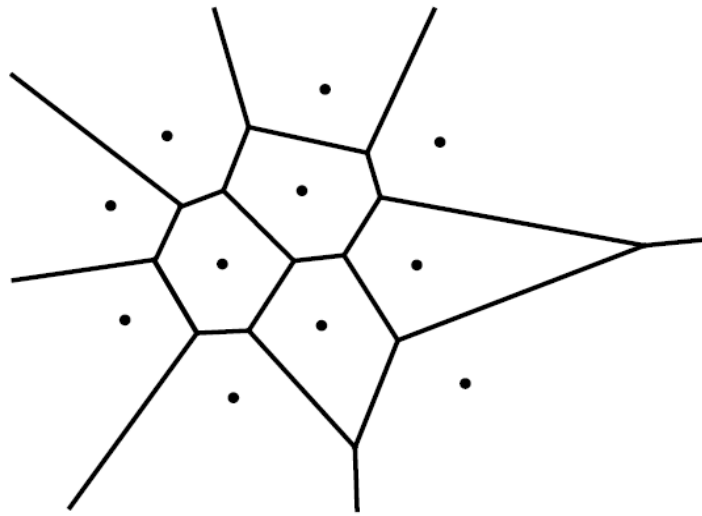
იგივე პრინციპით აიგება ვორონოის დიაგრამა 4 და მეტი წერტილისთვის (სურათი 20 (ა)). აღსანიშნავია, რომ წერტილები (გენერატორები) რომლებიც არ იღებენ მონაწილეობას ვორონოის რეგიონის ფორმირებაში არ არის აუცილებელი რომ გენერატორიდან უშორესი წერტილები იყოს, შესაბამისად მათი თანაჩართულობა დამოკიდებულია არა სიმორეზე არამედ განლაგებაზე (სურათი 20 (ბ)) [46].



სურათი 20. მრავალი წერტილით აგებული ვორონოის რეგიონი

ვორონოის სრული დიაგრამა ცალკეული რეგიონების გაერთიანებით მიიღება (სურათი 21) [46].

$$Vor(S) = \bigcup_{s \in S} Vor(s) \quad (22)$$



სურათი 21. ვორონოის დიაგრამა

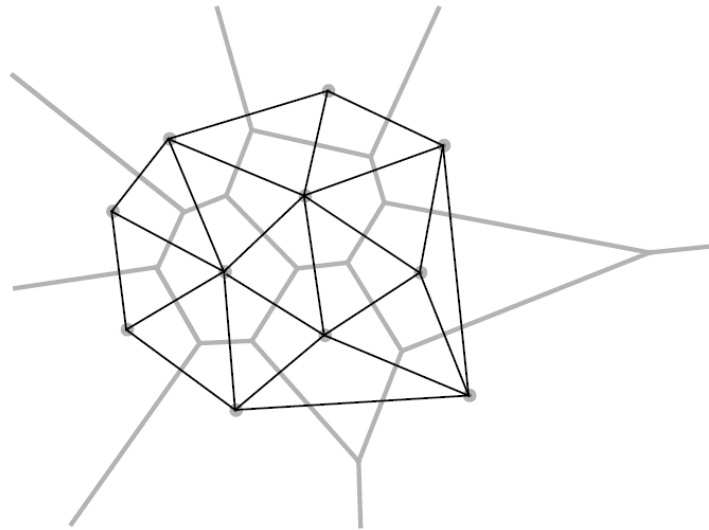
3.3.1.2 ვორონოის დიაგრამის მახასიათებლები

ვორონოის დიაგრამა ხასიათდება რამდენიმე თვისებით [46, 47]:

- თითოეული წერტილი (გენერატორი) ეკუთვნის მხოლოდ ერთ რეგიონს.
- ვორონოის წვეროების რაოდენობა არ აღემატება $2n - 5$.
- ვორონოის წიბოების რაოდენობა არ აღემატება $3n - 6$.
- ვორონოის წვერო არის ზუსტად 3 წიბოს თანაკვეთის წერტილი.
- თუ წერტილი $s_i \in S$ არის უახლოესი მეზობელი $s_j \in S$ წერტილის, მაშინ $V(s_i)$ და $V(s_j)$ რეგიონებს აქვთ საერთო წიბო.
- ვორონოის რეგიონები შეიძლება იყოს შემოსაზღვრული ან შემოუსაზღვრელი. $V(s)$ რეგიონი არის შემოუსაზღვრელი მაშინ და მხოლოდ მაშინ როცა s არის S სიმრავლის ექსტრემალური წერტილი (ყველაზე განაპირა წერტილი).

3.3.1.3 დელონეს ტრიანგულაცია

თუ წერტილებს N სიმრავლიდან შევაერთებთ ვორონოის რეგიონების გავლით, მივიღებთ ბრტყელ გრაფს რომელიც წარმოადგენს აღნიშნული წერტილების ტრიანგულაციას და მას ვორონოის დიაგრამის დუალიზმი ქვია, სხვანაირად კი როგორც მას ზემოთ ვუწოდეთ დელონეს ტრიანგულაცია (სურათი 22). აღნიშნული გრაფის წიბოები შესაძლებელია ორჯერაც კვეთდეს რეგიონის საზღვრებს (სურათი 22, ყველაზე მარჯვნივ მდებარე რეგიონი). ყველა Δt ტრიანგულაციისთვის, არცერთი ვორონოის გენერატორი არ მოხვდება მის შიგნით [46, 47].



სურათი 22. ვორონოის დიაგრამის დუალიზმი, დელონეს ტრიანგულაცია

3.3.2 კოჰონენის თვითორგანიზებადი რუკა (SOM)

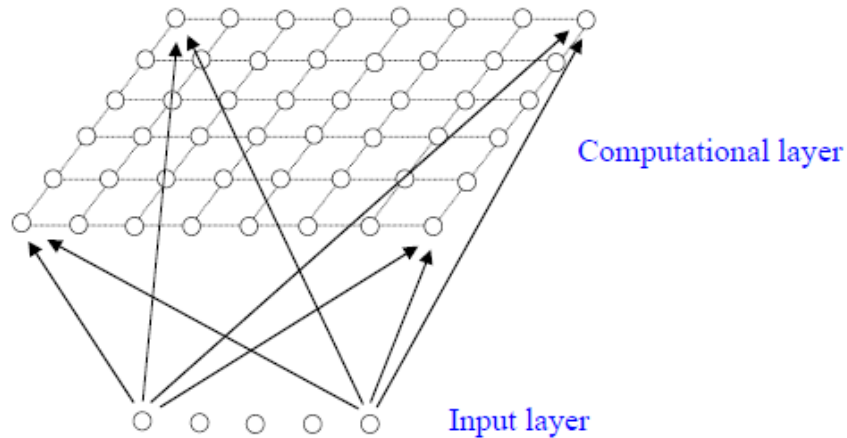
კოჰონენის თვითორგანიზებადი რუკები (Self-Organizing Maps, SOM) არაის ხელოვნური ნეირონული ქსელების ერთ-ერთი სახეობა, რომელიც იყენებს ქსელის არაკონტროლირებადი სწავლების მეთოდს. განსწავლის ესეთი მეთოდის დროს შესასწავლი ობიექტების სიმრავლე შედგენილია მხოლოდ შემავალი ცვლადების მნიშვნელობებისგან და სწავლების პროცესში ნეირონებიდან გამომავალი მნიშვნელობების შედარება არ ხდება ეტალონ მნიშვნელობებთან. შეიძლება ითქვას რომ ასეთი ქსელი სწავლობს მონაცემთა სტრუქტურის გაგებას.

თვითორგანიზებადი რუკები შეიძლება გამოყენებულ იქნას ისეთი ამოცანების გადასაჭრელად როგორცაა მოდელირება, პროგნოზირება, კანონზომიერებების და ურთიერთ დამოუკიდებელი ბლოკების აღმოჩენა მონაცემთა დიდ მასივებში, ინფორმაციის შეკუმშვა და სხვა. აღნიშნულ ამოცანებში კოჰონენის რუკების დანიშნულებაა მონაცემთა ანალიზი და კლასიფიკაცია და ახალი ფენომენების აღმოჩენა.

მონაცემთა ანალიზი და კლასიფიკაცია. კოჰონენის ქსელს შეუძლია ამოიცნოს როგორც კლასტერები მონაცემებში, ასევე კლასების მსგავსება. ასე რომ, კოჰონენის ქსელს შეუძლია კლასიფიკაციის ამოცანის გადაჭრა, რა თქმა უნდა თუ მონაცემებში ამოცნობადია კლასები. ამ მეთოდის გამოყენება შეიძლება ისეთ კლასიფიკაციის ამოცანებშიც, სადაც კლასები წინასწარ მოცემულია. ამ დროს, უპირატესობა ის არის, რომ ქსელს შეუძლია დაადგინოს მსგავსება სხვადასხვა კლასებს შორის.

ახალი ფენომენების აღმოჩენა. როგორც ზემოთ აღვნიშნეთ, კოჰონენის ქსელს შეუძლია შესასწავლი მონაცემებიდან კლასტერების გამოყოფა(აღმოჩენა) და შესაბამისი მონაცემების განაწილება ამა თუ იმ კლასტერზე. ამ გადანაწილების შემდეგ, თუ ქსელი აღმოაჩენს კლასტერს, რომლის მონაცემები არ ჰგავს არც ერთს ცნობილი ნიმუშებიდან, მაშინ იგი ვერ მოახერხებს ამ მონაცემების კლასიფიცირებას, რაც თავის მხრივ წარმოადგენს ახალ ფენომენს.

კოჰონენის ქსელის სწავლება. კოჰონენის ქსელი მრავალშრიანი ნეირონული ქსელისგან განსხვავებით ძალიან მარტივია. იგი შეიცავს მხოლოდ ორ შრეს: შემავალს და გამომავალს. რუკის ელემენტები განლაგებულნი არიან რაღაც სივრცეში, როგორ წესი ორ განზომილებიანში (სურათი 23).



სურათი 23. კოჰონენის ქსელი

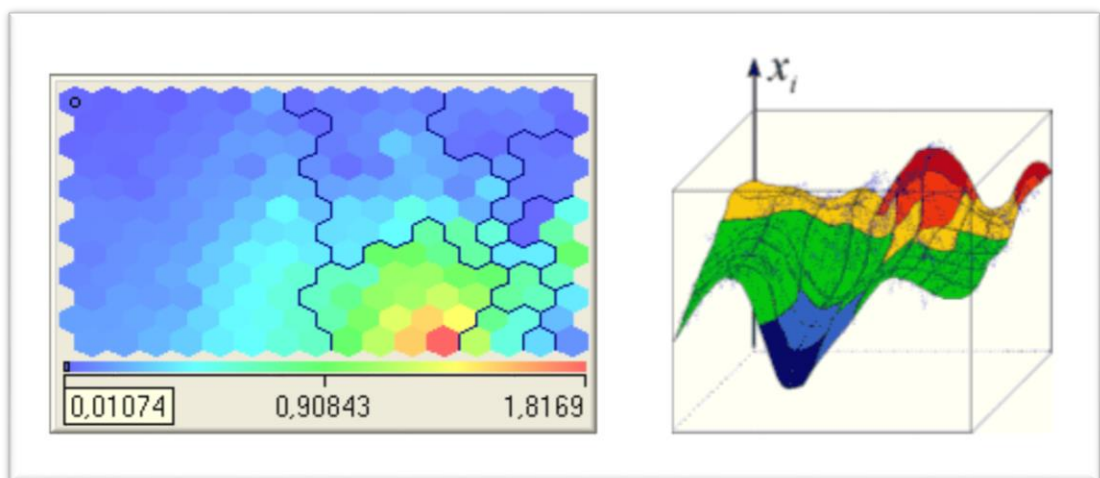
კოჰონენის ქსელის სწავლება ხდება მიმდევრობითი მიახლოების მეთოდით. სწავლების პროცესში ქსელს შესასვლელზე მიეწოდება რაღაც მონაცემები, რის შემდეგაც ქსელი იწყებს გადაწყობას. გადაწყობა ხდება შემავალი მონაცემების კანონზომიერების მიხედვით და არა გამომავალი მონაცემების ეტალონის მიხედვით. სწავლების დროს, მონაცემების თანმიმდევრული გადაცემით ქსელის შესასვლელზე განისაზღვრება გამარჯვებული ნეირონი(ნეირონი, რომლის წონის სკალარული ნამრავლი შესასვლელზე მიწოდებულ ვექტორთან მინიმალურია), რომელიც შემდეგში გამოიყენება მეზობელ ნეირონებში წონების განაწილების ცენტრად. ქსელის ასეთი სწავლება განიხილება როგორც „შეჯიბრებითი“ სწავლება, ნეირონებსა და გამარჯვებულ ნეირონს შორის მანძილის მხედველობაში მიღებით. ამ დროს სწავლების არსი შეცდომის მინიმიზაცია კი არ არის, არამედ შესაბამისობაში ყოფნა შემავალ მონაცემებთან.

კოჰონენის ალგორითმის იტერაციული მოდელი სრულდება რამდენიმე ეტაპად. თითოეულ ამ ეტაპზე მუშავდება მხოლოდ ერთი მაგალითი სასწავლო ნიმუშებიდან. თითოეულ შემავალ სიგნალზე ქსელში სასურველი გამომავალი სიგნალი არ განისაზღვრება. ქსელში შემავალი სინაპსური წონების ვექტორების საკმარისი რაოდენობის გაჩენის შემდეგ მას შეუძლია განსაზღვროს კლასტერი. წონები ორგანიზებულია ისე, რომ

ტოპოლოგიურად ახლო კვანძები მგრძნობიარენი არიან მსგავს შემავალ სიგნალებთან.

ალგორითმის მუშაობის შედეგად კლასტერის ცენტრი ექცევა იმ პოზიციაში რომელიც აკმაყოფილებს კლასტერიზაციის შესასწავლ მაგალითის, რომლისთვისაც მოცემული ნეირონი არის „გამარჯვებული“. საჭიროა განისაზღვროს ნეირონების მეზობლობის ზომა, ანუ გამარჯვებული ნეირონის შემოგარენი.

თვითორგანიზებადი რუკების უნიკალურობა მდგომარეობს n -განზომილებიანი სივრცის ორ განზომილებაში გადაყვანის შესაძლებლობაში. ორ განზომილებიანი ქსელის გამოყენება ნაკარნახებია იმით, რომ დიდ სირთულეებთან არის დაკავშირებული რთული სტრუქტურის წარმოდგენა მრავალგანზომილებიან ქსელში. მონაცემთა ასეთი სტრუქტურით წარმოდგენის დროს ადვილია განისაზღვროს დამოკიდებულების არსებობა/არარსებობა შემავალ მონაცემებს შორის. კოჰონენის ნეირონულ რუკებს განათავსებენ ორ განზომილებიანი მატრიცის სახით და აფერადებენ ნეირონების მიერ გაანალიზებული პარამეტრების წონის (მნიშვნელობის) მიხედვით (სურათი 24). ნეირონები რუკაზე გამოსახულია უჯრედების სახით და მიგნებადია კოორდინატებით.

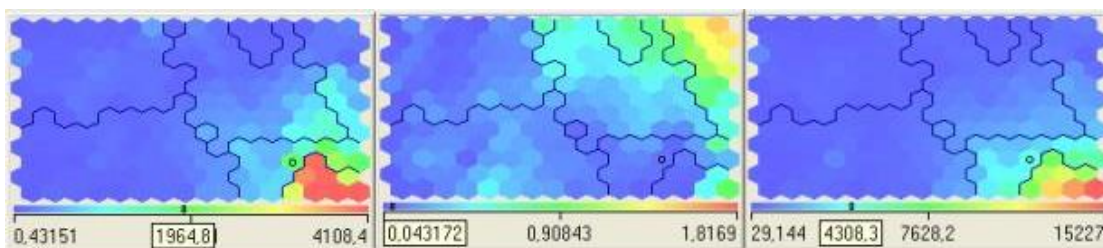


სურათი 24. კოჰონენის რუკა

ალგორითმის მუშაობის შედეგად ვლელულობთ სამი ტიპის რუკას:

- ნეირონებზე შემავალი ინფორმაციის რუკა
- ნეირონებიდან გამომავალი ინფორმაციის რუკა
- სპეციალური რუკები

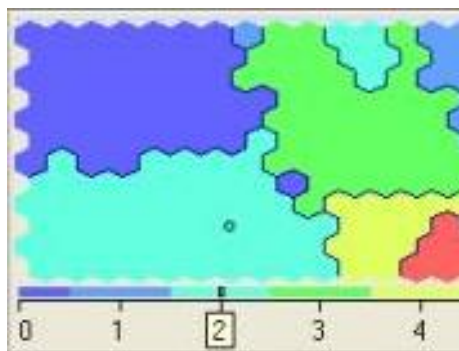
ნეირონებზე შემავალი ინფორმაციის რუკა. ნეირონების წონები რეგულირდება შემავალი მონაცემების მნიშვნელობებით ანუ მათი სტრუქტურის გამოხატულებათ. ყოველი შემავალი სიგნალისთვის იხატება დამოუკიდებელი რუკა, შეღებილი კონკრეტული ნეირონის წონის მიხედვით. მონაცემთა ანალიზისთვის ერთდროულად გამოიყენება რამდენიმე (როგორც წესი 3) ასეთი რუკა (სურათი 25).



სურათი 25. კოპონენის რუკა

ერთ-ერთ რუკაზე გამოყოფენ წინასწარ განსაზღვრული ფერის რეგიონს - ეს ნიშნავს, რომ შესაბამის შემავალ ნიმუშებს აქვთ დაახლოებით ერთნაირი მნიშვნელობები. ფერების გავრცელება ამ რეგიონში დარდება სხვა რუკებს მსგავსი თუ განსხვავებული თვისებების დასადგენად.

ნეირონებიდან გამომავალი ინფორმაციის რუკა. გამომავალი ინფორმაციის რუკაზე პროექტირდება შესწავლილი ინფორმაციის ერთობლივი განლაგება. ნეირონები, გამომავალი ინფორმაციის ერთნაირი მნიშვნელობებით აყალიბებენ კლასტერს - ჩაკეტილ რეგიონს რუკაზე (სურათი 26).



სურათი 26. კლასტერი

სპეციალური რუკები. ესენია, კლასტერის რუკა, დაშორებების მატრიცა, დაცემის სიმკვრივის მატრიცა და სხვა რუკები, რომლებიც აღწერენ(ახასიათებენ) კოჰონენის ქსელის სწავლების შედეგად მიღებულ კლასტერებს.

მნიშვნელოვანია გვახსოვდეს, რომ ზემოთ აღწერილ ყველა რუკას შორის არსებობს ურთიერთკავშირი, რადგან ყველა ეს რუკა არის სხვადასხვა შეფერილობა ერთი და იგივე ნეირონის. თითოეულ მაგალითს შესწავლის სიმრავლიდან გააჩნია ერთი და იგივე განლაგება ზემოთ აღწერილ თითოეულ რუკაზე [49, 50].

3.3.3 კლასტერიზაცია *K-Means* - ის მეთოდით

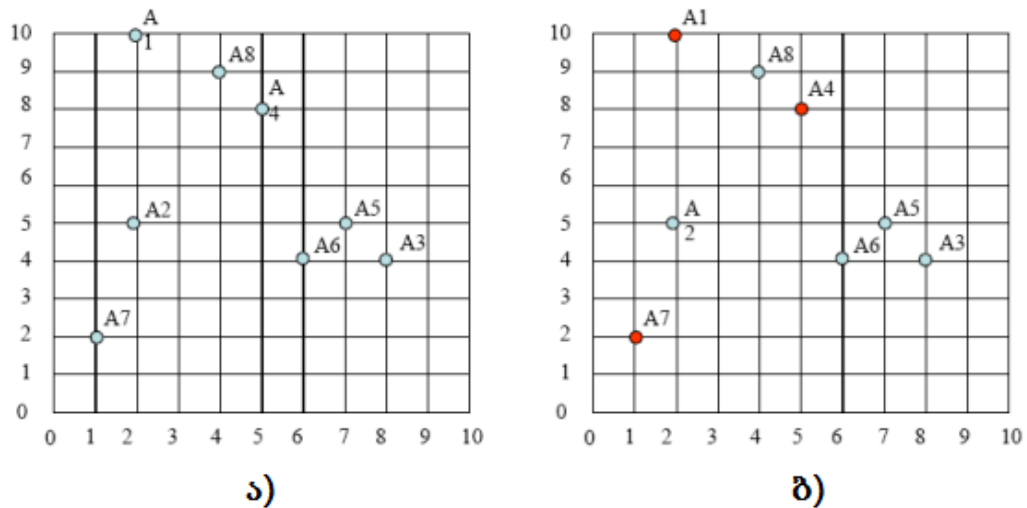
ტერმინი *K-Means* პირველად გამოიყენა ჯეიმს მაქქვინმა (James MacQueen) 1967 წელს, მაგრამ იდეა ეკუთვნის ჰუგო სტეინჰაუსს (Hugo Steinhaus), 1957 წელი. სტანდარტული ალგორითმი პირველად შეიმუშავა სტიუარდ ლოიდმა (Stuart Lloyd) 1957 წელს, როგორც იმპულსური კოდირების მოდულაციის მეთოდი. მოგვიანებით კი 1975-1979 წლებში ჰარტიგანმა (Hartigan) და ვონგმა (Wong) შემოგვთავაზეს უფრო ეფექტური და დახვეწილი მოდელი [51].

K-Means ალგორითმი მიეკუთვნება „პროტოტიპზე-დაფუძნებულ“ (prototype-based), „კარგად გაყოფილი“ (well-separated) კლასტერიზაციის ალგორითმების ჯგუფს. მიდგომა ძალზე მარტივია და მოიცავს მონაცემთა

დაყოფას/გრუპირებას კონკრეტული ატრიბუტებით/თვისებებით K ჯგუფად, სადაც K დადებითი მთელი რიცხვია. ალგორითმის პროტოტიპია ცენტროიდი, რომელიც კლასტერის გეომეტრიულ ცენტრს წარმოადგენს.

3.3.3.1 მარტივი მოდელი

K -Means - ის მარტივი მოდელი განვიხილოთ მაგალითის საფუძველზე. ვთქვათ მოცემულია $X = (A_1, A_2, A_3 \dots A_n)$ წერტილების რაოდენობა, რომლებიც უნდა დაიყოს K რაოდენობის კლასტერად (სურათი 27 (ა)). თავდაპირველად ხდება K რაოდენობის წერტილის არჩევა კლასტერების საწყის ცენტროიდებად (სურათი 27 (ბ)). ცენტროიდი როგორც წესი არის კლასტერის გეომეტრიული ცენტრი. საწყისი ცენტროიდების არჩევა ძირითად შემთხვევებში ხდება შემთხვევითად. ხანდახან, წერტილების ერთი და იგივე განლაგებისთვის, სხვადასხვა საწყისი ცენტროიდები სხვადასხვა საბოლოო შედეგს იძლევა.



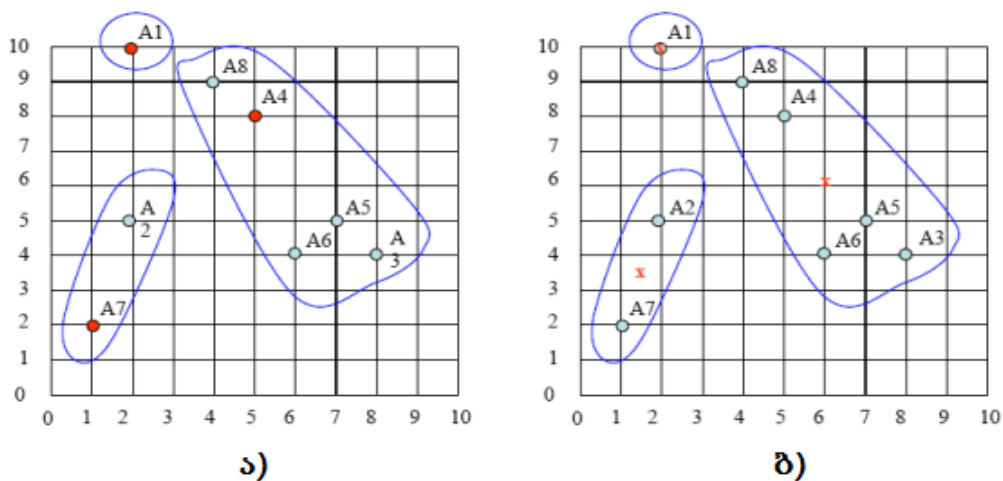
სურათი 27. კლასტერიზაციის ამოცანა, საწყისი მდგომარეობა

შემდეგ K - ური ცენტრისთვის ვპოულობთ, მეზობელ, უახლოეს წერტილებს, რომლებსაც გავაერთიანებთ აღნიშნულ კლასტერში (სურათი 28 (ა)). წერტილებს შორის სიახლოვის გასაზომად გამოიყენება ევკლიდური მანძილი.

$$a = (x_1, y_1), \quad b = (x_2, y_2) \quad (23)$$

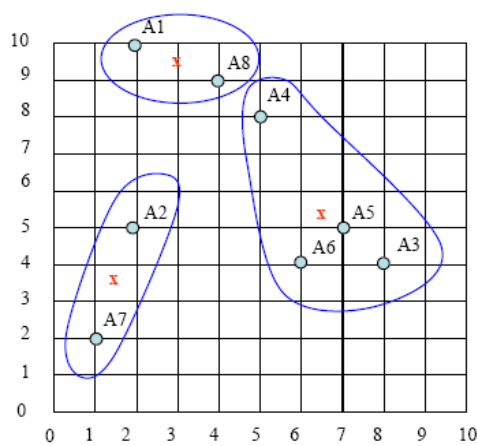
$$p(a, b) = |x_2 - x_1| + |y_2 - y_1|$$

და ბოლოს ყველა კლასტერისთვის გამოითვლება ცენტროიდი (სურათი 28 (ბ)).

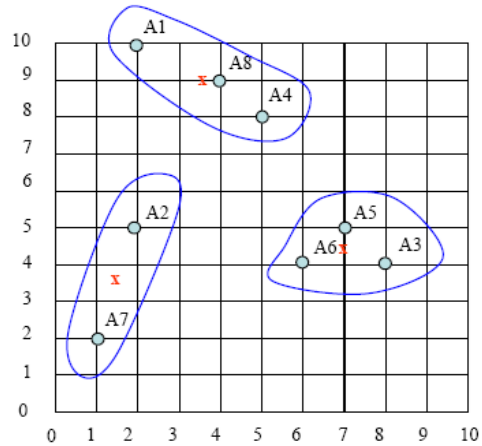


სურათი 28. კლასტერიზაციის ამოცანა, შუალედური მდგომარეობა

ზემოთ აღწერილი ბიჯები მეორდება უკვე ახალი ცენტროიდებისთვის. ცენტროიდების ცვლილება ძირითადად დამახასიათებელია პირველ რამდენიმე იტერაციაზე. იტერაციების მაქსიმალური რაოდენობა K^n . ცენტროიდის ცვლილებასთან ერთად, ყოველ ახალ იტერაციაზე, შესაძლებელია წერტილები ერთი კლასტერიდან გადავიდეს მეორე კლასტერში (სურათი 29). პროცესი გრძელდება მანამ სანამ ცენტროიდები იცვლება. სურათი 29 (ბ) - ზე მოცემულია კლასტერების საბოლოო მდგომარეობა.



ა)



ბ)

სურათი 29. კლასტერიზაციის ამოცანა, საბოლოო მდგომარეობა

K-Means - ალგორითმის ძირითადი ბიჯების ფსევდოკოდი გამოიყურება შემდეგნაირად:

1. ინიციალიზაცია:

- K - კლასტერების რაოდენობის განსაზღვრა.
- თითოეული კლასტერისთვის საწყისი ცენტროიდების არჩევა.

2. გაიმეორე:

- K - ური კლასტერისთვის ცენტროიდთან უახლოესი წერტილების მინიჭება.
- თითოეული კლასტერის ცენტროიდის ხელახლა გამოთვლა.

3. სანამ: ცენტროიდების პოზიციები იცვლება

არსებობს K-Means ალგორითმის მრავალი სახეცვლილება. თითოეული მათგანი მორგებული და ოპტიმიზირებულია კონკრეტული ტიპის ამოცანებისთვის. ქვემოთ განვიხილავთ ყველაზე მეტად ცნობილ და ფაქტიურად დასტანდარტებულ მიდგომებს.

3.3.3.2 K-Means – ის ვარიაციები

Fuzzy C-Means. მეთოდი დაფუძნებულია არამკაცრი კლასტერიზაციის ლოგიკაზე, სადაც ყოველ წერტილს, იმის მაგივრად რომ ეკუთვნოდეს

მხოლოდ ერთ კლასტერს, გააჩნიათ თითოეულ კლასტერთან მიკუთვნების ხარისხი(წონა). ამგვარად, წერტილებს კლასტერის საზღვრებთან შეიძლება ჰქონდეს ნაკლები მიკუთვნების ხარისხი ვიდრე წერტილებს ცენტრთან ახლოს. ალგორითმს K-Means - ისგან განსხვავებით გააჩნია ცენტროიდების დადგენის განსხვავებული ფორმულა, კერძოდ კლასტერის ცენტროიდი არის ყველა წერტილის საშუალო მნიშვნელობა, შეფასებული ამ წერტილის კლასტერთან მიკუთვნების ხარისხით. ანუ ნებისმიერ x წერტილს აქვს მინიჭებული k - ურ კლასტერთან მიკუთვნების კოეფიციენტი $w_k(x)$.

$$C_k = \frac{\sum_x w_k(x)x}{\sum_x w_k(x)} \quad (24)$$

კლასტერიზაციის ეს მეთოდი ყველაზე ხშირად გამოიყენება გრაფიკულ მონაცემებთან სამუშაოდ (სურათების კლასტერიზაცია და დამუშავება, გამოსახულების ამოცნობა) [52, 53].

გაუსის მიქსტურა (Mixture of Gaussians): ალგორითმი დაფუძნებულია *Expectation-maximization (EM)* - ალგორითმზე. ეს უკანასკნელი კი არის იტერაციული მეთოდი, რომელიც გამოიყენება სტატისტიკურ მოდელებში მაქსიმალური მსგავსების დასადგენად. გაუსის მიქსტურას K-Means - ისგან განსხვავებით დეტერმინისტულის ნაცვლად გააჩნია კლასტერებზე ალბათური გადანაწილების ბუნება, ხოლო „საშუალოს“ (means) ნაცვლად იყენებს გაუსის მრავალგანზომილებიან გაბნევას [52, 54].

K-Means ++: ალგორითმი შეიმუშავეს K-Means - თვის საწყისი მონაცემების შესარჩევად, რადგან ხანდახან ცუდი ინიციალიზაციის პირობებში იგი კლასტერიზაციის ცუდ შედეგს იძლევა. აღნიშნულ პრობლემის მოსაგვარებლად, დღესდღეობით, საუკეთესო ალგორითმია K-Means ++ [52, 55].

k-d tree: იგი წარმოადგენს სპეციალურ მონაცემთა სტრუქტურას, უფრო ზუსტად კი ბინარულ ხეს, რომელიც უზრუნველყოფს მრავალგანზომილებიანი სივრცის დაყოფას კლასტერებად და წერტილების

k განზომილებაზე გადანაწილებას. ანუ თითოეული კვანძი ამ ხეში წარმოადგენს k განზომილების წერტილს. იგი გამოიყენება მონაცემთა ფილტრაციისთვის K-Means ალგორითმის ბიჯების ასაჩქარებლად [52, 56].

მინკოვსკის წონითი K-Means (Minkowski Weighted K-Means):
 ალგორითმი წარმოადგენს K-Means – ს მინკოვსკის სივრცეში, სადაც ყველა კლასტერისთვის ავტომატურად გამოითვლება თვისობრივი წონები. სწორედ ამ თვისობრივი წონების საშუალებით იცვლება სტანდარტული ალგორითმის დისტანციის გამოთვლის წესები. თვისობრივი წონები w_{kv} დამოკიდებულია k - ური კლასტერის v - თვისებაზე, ამასთან დასაშვებია v - თვისებას სხვადასხვა წონა ჰქონდეს სხვადასხვა კლასტერში. მეთოდში დისტანცია თვისობრივი წონების გათვალისწინებით, ევკლიდური დისტანციის ანალოგიურად გამოითვლება:

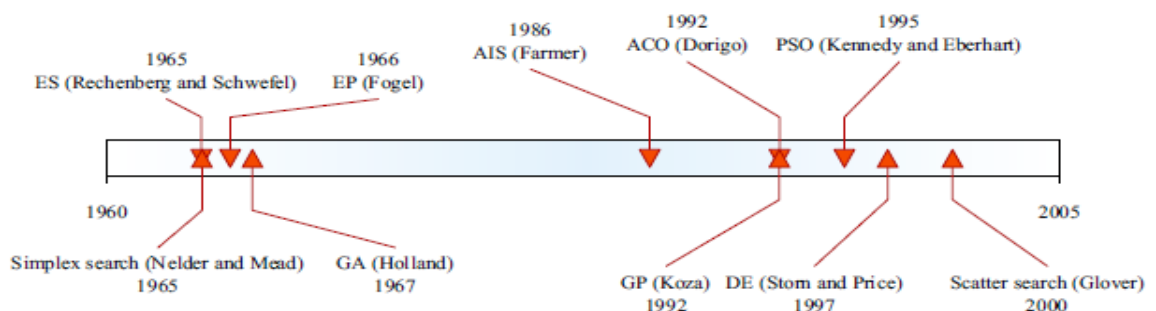
$$d_p(y_i, c_k) = \sum_{v=1}^V w_{kv}^p |y_{iv} - c_{kv}|^p \quad (25)$$

სადაც p არის მინკოვსკის ექსპონენტი, ცოლო c კლასტერის ცენტროიდი [57, 58].

თავი 4 ევოლუციური ალგორითმების მიმოხილვა და ანალიზი

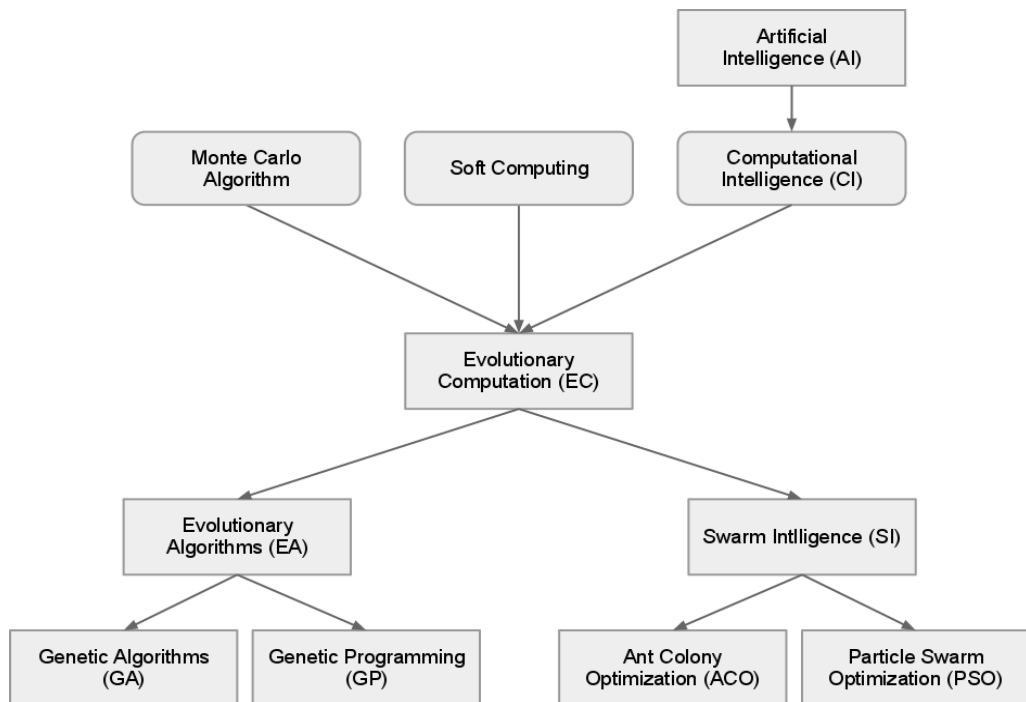
ცოცხალი ორგანიზმების ქცევების ანალიზის საფუძველზე შეიქმნა მეთოდების მთელი კლასი, რომელსაც ხელოვნური ინტელექტი (AI) უწოდეს. მასში გაერთიანებულია მრავალი ოპტიმიზაციის ალგორითმი, რომელთაც გააჩნიათ სწავლისა და გადაწყვეტილების მიღების უნარი. ბუნებრივ ევოლუციას და ახალ გარემოსთან ადაპტაციას ეს ალგორითმები განიხილავენ, როგორც სწავლის პროცესს და პრინციპი „სახეობაში ძლიერთა გადარჩენა“ (survival of the fittest), გამოიყენება ალგორითმის მიერ ოპტიმიზაციისა და სასურველი შედეგის მიღებისათვის. სწორედ ასეთ ალგორითმებს უწოდებენ ევოლუციურ ალგორითმებს (EA). ევოლუციური ალგორითმები არის პოპულაციაზე დაფუძნებული ალგორითმები და ისინი იყენებენ ბიოლოგიური ევოლუციის ელემენტებს, როგორებიცაა: გამრავლება, მუტაცია, რეკომბინაცია, ბუნებრივი სელექცია და „სახეობაში ძლიერთა გადარჩენა“. EA-პოპულაციის ინდივიდებს იყენებს საპრობლემო არის პარალელური შესწავლის მიზნით. თითოეული ინდივიდი წარმოადგენს საპრობლემო არის შესაძლო ამონახსნს, ამიტომ ოპტიმალური ამონახსნის მისაღებად ხდება ძლიერი და სუსტი ინდივიდების გადარჩევა [62].

სურათი 30-ზე მოცემულია ევოლუციური ალგორითმების ქრონოლოგია.



სურათი 30: ევოლუციური ალგორითმების ქრონოლოგია

სურათი 31-ზე მოცემულია ევოლუციური ალგორითმების კლასიფიკაცია.



სურათი 31. ევოლუციური ალგორითმების კლასიფიკაცია

ოპტიმიზაციის ალგორითმები იყოფა ორ ჯგუფად: „დეტერმინისტული“ (Deterministic)^I და „ალბათური“ (Probabilistic)^{II}. დეტერმინისტული მოდელი გამოიყენება როცა ნათელი კავშირი არსებობს შემავალ პარამეტრებსა და შესაძლო ამონახსნებს შორის. წინააღმდეგ შემთხვევაში გამოიყენება ალბათური მოდელი [61].

4.1 გენეტიკური ალგორითმი (GA)

გენეტიკური ალგორითმი (GA) დაფუძნებულია ბუნებრივი ევოლუციის პრინციპებზე, სადაც პოპულაციის ინდივიდები განუწყვეტლივ

^I[მათემატიკური ფუნქციები, რომლებიც ნებისმიერი არგუმენტისთვის იღებენ უნიკალურ მნიშვნელობას.](#)

^{II}[ალბათური ალგორითმები](#) ამონახსნის საძიებლად იყენებენ შემავალი პარამეტრების შემთხვევით მნიშვნელობებს. არის იმის შანსი რომ დააბრუნონ არასწორი შედეგი ან საერთოდ კრაჩით(შეცდომა ალგორითმის შესრულების დროს, სისტემას გამოაქვს შეტყობინება შეცდომის შესახებ და ასრულებს მუშაობას) დაამთავრონ გამოთვლები.

ცხოველქმედებენ, ვითარდებიან და იბრძვიან გადარჩენისთვის. ამ ბრძოლაში, უფრო ძლიერ და ადაპტირებულ ინდივიდებს გადარჩენის მეტი შანსი აქვთ ვიდრე სუსტებს, შესაბამისად მათი შთამომავლობაც გენეტიკურად უფრო ძლიერი იქნება.

პირველად ეს იდეა წარმოადგინა და დაამუშავა John Holland-მა 1975 წელს, ხოლო შემდეგში ფართო გამოყენება ჰპოვა სხვადასხვა მეცნიერების მხრიდან ოპტიმიზაციის სხვადასხვა ამოცანებში. მისი სიძლიერე იმაში მდგომარეობს რომ, იგი დიდი განზომილების მრავალექსტრემუმიანი და მრავალკრიტერიუმიანი ამოცანებისათვის ოპტიუმის სწრაფი ძებნის შესაძლებლობას იძლევა, რასაც მონაცემთა პარალელური დამუშავების გზით მიიღწევა [63].

4.1.1 გენეტიკური ალგორითმის კლასიკური მოდელი

გენეტიკური ალგორითმი (GA) არის საძიებო არეში მიახლოებითი საუკეთესო მნიშვნელობის ძიების ალგორითმი, რომელიც იყენებს გენეტიკურ მოდელს თავისი ქმედების წარსამართავად. თითოეული ამონახსნი ამ არეში წარმოდგენილია ქრომოსომის (გენების ერთობლიობის) სახით. ქრომოსომაში ამონახსნის კოდირების მეთოდის შერჩევა არის ალგორითმის აგების საწყისი ეტაპი. არსებობს კოდირების რამდენიმე მეთოდი [64]:

- ორობითი კოდირება - ქრომოსომას წარმოადგენენ ფიქსირებული ზომის ბიტურ სტრიქონებად, ანუ გადაყავთ თვლის ორობით სისტემაში(ნულებით და ერთებით) და მათზე ატარებენ მარტივ ორობით ოპერაციებს როგორებიცაა NOT, OR, AND, XOR.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

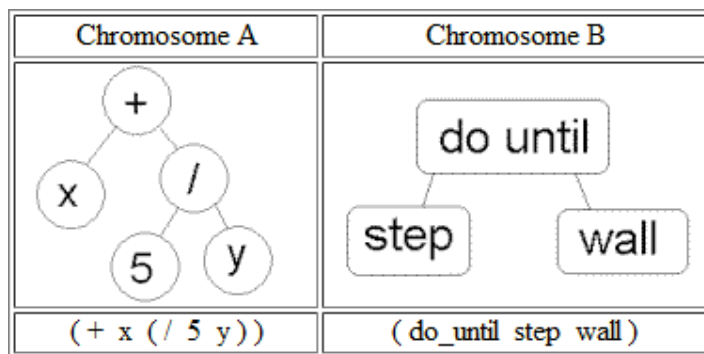
სურათი 32. GA-ში ორობითი კოდირება

- ალფაბეტური და რიცხვითი კოდირება - ქრომოსომა წარმოდგენილია სიტყვების ან ნამდვილი რიცხვების სახით და მიღებულ სტრიქონებზე ხდება შესაბამისი ოპერაციები.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

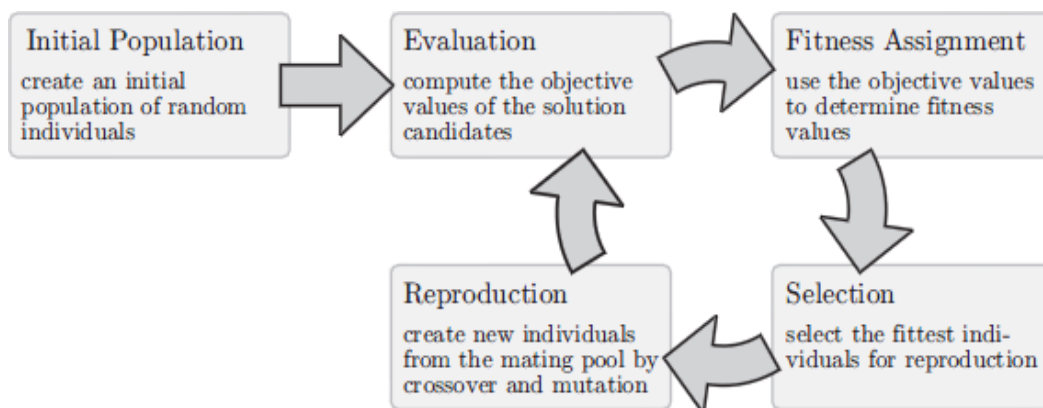
სურათი 33. GA-ში ალფაბეტური და რიცხვითი კოდირება

- ხისებრი სტრუქტურით კოდირება - თითოეული ქრომოსომა წარმოდგენილია ხის სტრუქტურის სახით. ეს კოდირება ძირითადად გენეტიკურ პროგრამირებაში (GP) გამოიყენება.



სურათი 34. GA-ში ხისებრი სტრუქტურის კოდირება

ალგორითმი შედგება რამდენიმე ძირითადი ბიჯისგან (სურათი 35).
განვიხილოთ თითოეული მათგანი [65].



სურათი 35. გენეტიკური ალგორითმის მოქმედების ძირითადი ციკლი

ინიციალიზაცია (Initialization)

გენეტიკური ალგორითმისათვის საწყისი პოპულაცია შემთხვევითად გენერირდება საძიებო არის შესაძლო ამონახსნებისაგან. პოპულაციის ზომა დამოკიდებულია კონკრეტულ ამოცანაზე, მაგრამ როგორც წესი შეიცავს რამდენიმე ასეულ ან ათას შესაძლო ამონახსნს (ქრომოსომას).

ევოლუცია (Evolution) და „ფიტნეს ფუნქციის“ განსაზღვრა

თითოეული ქრომოსომა ატარებს თავისი „ფიტნეს ფუნქციის“ (Fitness function) მნიშვნელობას, რომელიც განსაზღვრავს თუ რამდენად კარგ ამონახსნად ითვლება იგი შესაძლო ამონახსნთა სიმრავლეში, ანუ განსაზღვრავს ინდივიდის გენის სიმძლიერეს. რაც უფრო ძლიერია გენი მით უფრო დიდი შანსი აქვს გადარჩენის, ანუ სელექციის დროს ასეთი გენებისგან შემდგარი ინდივიდი დიდი ალბათობით მოხვდება არჩეულ ინდივიდებს შორის.

სელექცია (Selection)

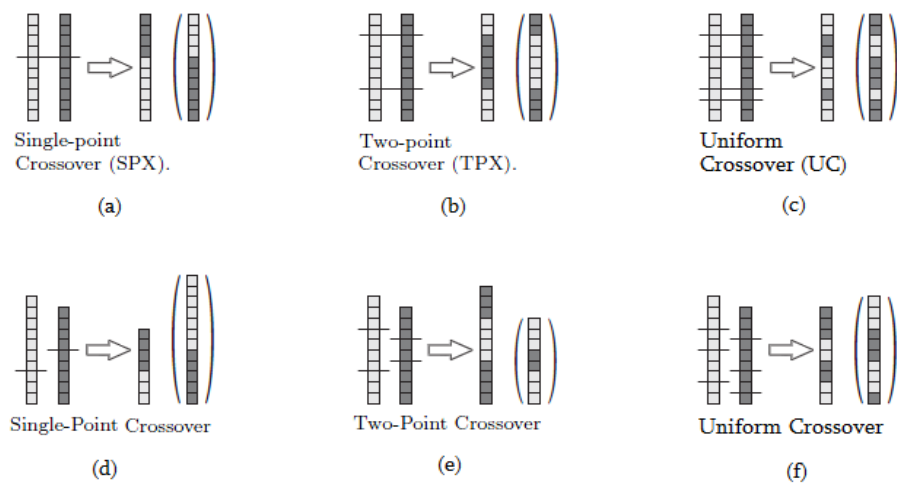
სელექცია (Selection) არის პროცესი როცა პოპულაციის ინდივიდები ირჩევა (ფიტნეს ფუნქციის გათვალისწინებით) რეპროდუქციისათვის ანუ შემდეგი თაობის პოპულაციის მისაღებად. ახალი თაობის წარმომადგენელი ინდივიდები უფრო ძლიერი გენების მატარებლები იქნებიან ვიდრე წინა თაობის ინდივიდები (ქრომოსომები). სელექციის დროს გასათვალისწინებელია რამდენიმე მომენტი. პირველ რიგში, უნდა განისაზღვროს თუ რამდენი შთამომავალი უნდა შეიქმნას შემდეგი თაობისათვის, ასევე როგორ უნდა მოხდეს სელექციის დაბალანსება, რადგან ძალიან „მკაცრი“ სელექციის პროცესი გამოიწვევს მრავალფეროვნების შემცირებას გენებს შორის (ანუ მოხდება ყველა ძლიერი გენის არჩევა), რაც საჭიროა შემდეგი ცვლილებებისა და პროგრესირებისათვის, მეორეს მხრივ

კი ძალიან „სუსტი“ სელექცია ევოლუციის პროცესს შეანელებს. არსებობს სელექციის რამდენიმე მეთოდი [66]:

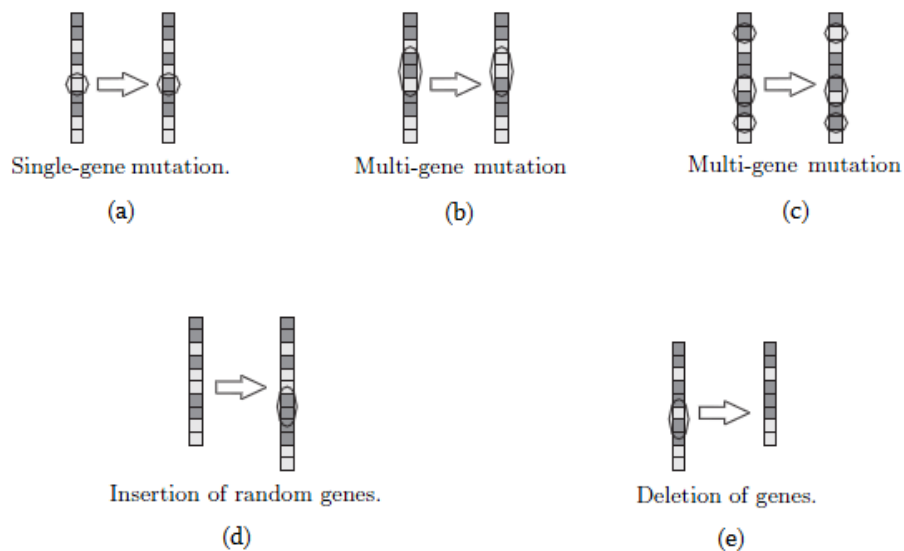
- „ფიტნეს-პროპორციული“ ან რულეტკის წესით სელექცია (Fitness proportionate, Roulette-wheel selection);
- „ბოლცმანის სელექცია“ (Boltzmann Selection);
- „სელექცია რანჟირებით“ (Rank Selection);
- „სელექცია ტურნირის წესით“ (Tournament Selection);
- „სელექცია დამრგვალების (ჩამოჭრის) წესით“ (Truncation selection);
- „ამორჩევა მყარი წესით“ (Steady-State Selection);
- „ამორჩევა სტოქასტიკურ-უნივერსალური წესით“ (Stochastic universal sampling);

რეპროდუქცია და პოპულაციის შეცვლა (Reproduction)

რეპროდუქცია (Reproduction) ნიშნავს შემდეგი თაობის პოპულაციის გენერაციას წინა თაობის ინდივიდებისაგან. შემდეგი თაობის გენერაცია ხდება გენეტიკური ოპერატორების დახმარებით. ეს იმას ნიშნავს, რომ შემდეგი თაობის ერთი წარმომადგენელი მიიღება წინა თაობის ერთ ან რამდენიმე წარმომადგენელს შორის კონკრეტული გენეტიკური ოპერაციის ჩატარებით. არსებობს რამდენიმე გენეტიკური ოპერატორი(მაგ: ინვერსია (inversion), გენის დუბლირება (gene doubling), გადაჯგუფება (regrouping), კოლონიზაცია (colonization), მიგრაცია (migration) და სხვა), მაგრამ მათ შორის ორი არის ძირითადი და გამოიყენება გენეტიკური ალგორითმის კლასიკურ მოდელში. ესენია „შეჯვარება“(Crossover) (სურათი 36) და „მუტაცია“(Mutation) (სურათი 37). რეპროდუქციის შემდეგ წინა თაობის ჩანაცვლება ხდება ახალი თაობის პოპულაციით (Replacement) [67, 68, 69].



სურათი 36. Ga-ში ქრომოსომების შეჯვარება



სურათი 37. Ga-ში ქრომოსომების მუტაცია

ალგორითმის დასრულება (Termination)

ალგორითმი გრძელდება მანამ, სანამ „დასრულების პირობა“ არ შესრულდება, ანუ არ შესრულდება ჩამოთვლილთაგან ერთ-ერთი [70, 71]:

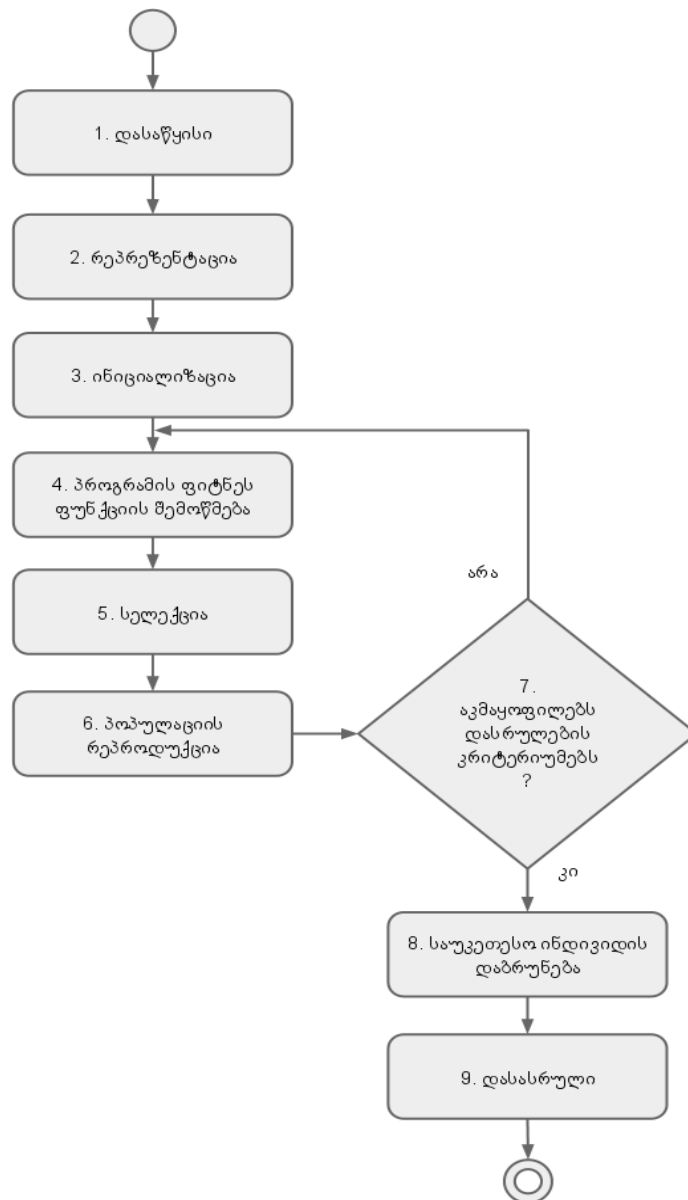
- ნაპოვნია პასუხი, რომელიც აკმაყოფილებს მინიმალურ კრიტერიუმებს;

- ალგორითმმა წინასწარ განსაზღვრული რეპროდუქციების რიცხვს გადააჭარბა;
- ალგორითმმა წინასწარ განსაზღვრულ „ბიუჯეტს“ (დრო/სხვა რესურსები) გადააჭარბა;
- შესაძლო ამონახსნის „ფიტნეს ფუნქციის“ მნიშვნელობამ მაქსიმუმს მიაღწია ან შემდეგი იტერაციების შედეგად აღარ უმჯობესდება;

4.2 გენეტიკური პროგრამირება (GP)

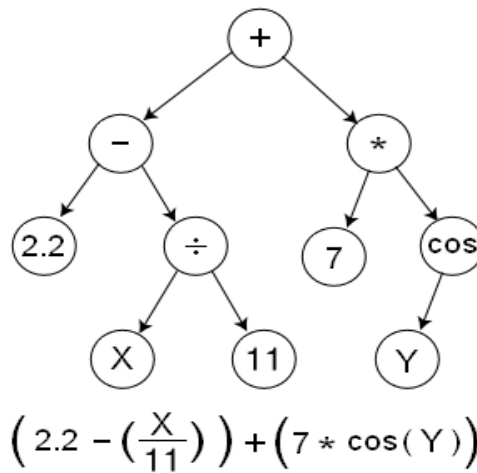
გენეტიკური პროგრამირება არის გენეტიკური ალგორითმის გაფართოებული ვარიანტი და ორიენტირებულია კომპიუტერული პროგრამების სივრცეში ოპტიმიზაციის ამოცანების ამოხსნისკენ. იგი, ამოცანის გადასაწყვეტად, უცნობი ალგორითმისთვის პროგრამის ავტომატურ სინთეზს იყენებს. მისი დამფუძნებელია ჯ. კოზა (J. Koza 1992.წ.) გენეტიკურ პროგრამირებაში ევოლუციას განიცდის კომპიუტერული პროგრამების პოპულაცია, რის შედეგადაც მიიღება ახალი პოპულაცია „უკეთესი“ პროგრამებით.

GP მეთოდი უპირატესად გამოიყენება შემდეგ შემთხვევებში: როდესაც კლასიკური მათემატიკური ანალიზი არ იძლევა ანალიტიკურ შედეგს; როდესაც კავშირი დამოუკიდებელ და დამოკიდებულ ცვლადებს შორის უცნობია, ან ერთობ საეჭვოა; როდესაც დაშვებულია მიახლოებითი ამონახსნი. როდესაც საქმე გვაქვს მონაცემთა დიდ მასივებთან და საჭიროა მათი დამუშავება, კლასიფიკაცია, ინტეგრირება და სხვ. გენეტიკური პროგრამირების ალგორითმის მოქმედების ბლოკ-სქემა მოცემულია სურათი 38-ზე [72].



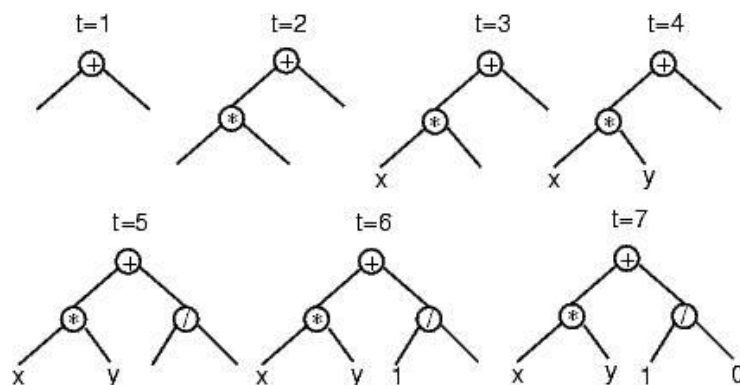
სურათი 38. გენეტიკური პროგრამირების ალგორითმის ბლოკ-სქემა

რეპრეზენტაცია: გენეტიკურ პროგრამირებაში გამოიყენება მონაცემთა წარმოდგენის ხისებრი სტრუქტურა. სურათი 39-ზე მოცემულია $2.2 - \frac{x}{11} + 7 * \cos(y)$ გამოსახულების ხისებრი სტრუქტურით წარმოდგენის მაგალითი, სადაც ცვლადები, კონსტანტები და არითმეტიკული ოპერაციები წარმოდგენილია ხის ფოთლების სახით. ცვლადებს და კონსტანტებს ტერმინალებს უწოდებენ, ხოლო არითმეტიკული ოპერაციებს ფუნქციებს [73].

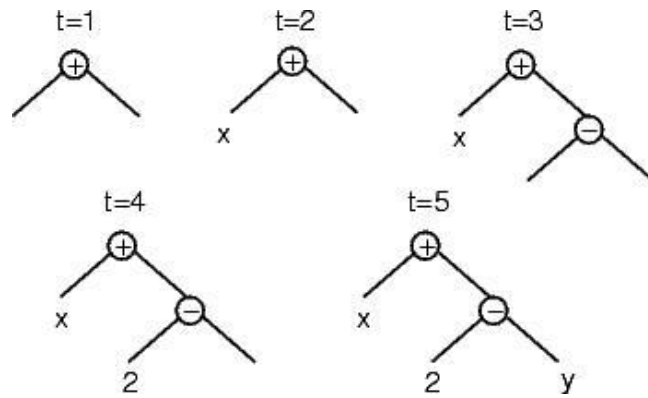


სურათი 39. გამოსახულების წარმოდგენა ხისებრი სტრუქტურით

ინიციალიზაცია: სხვა ევოლუციური ალგორითმების მსგავსად გენეტიკურ პროგრამირებაშიც საწყისი პოპულაცია შემთხვევითად გენერირდება. არსებობს ინიციალიზაციის ორი მეთოდი „სრული“ (full) და „გაზრდის“ (grow). „სრული“ მეთოდის დროს ტერმინალები წარმოდგენილია ხის ერთ დონეზე (სურათი 40), რაც გარკვეულ შემთხვევაში შეზღუდვად კი შეიძლება ჩაითვალოს. „გაზრდის“ მეთოდის დროს კი ნებისმიერი დონის ხის ტოტის ასაგებად გამოიყენება როგორც ტერმინალი ასევე ფუნქცია (სურათი 41), რაც მეტ თავისუფლებას იძლევა მონაცემების წარმოსადგენად [74].



სურათი 40. ინიციალიზაცია „სრული“ (Full) მეთოდით

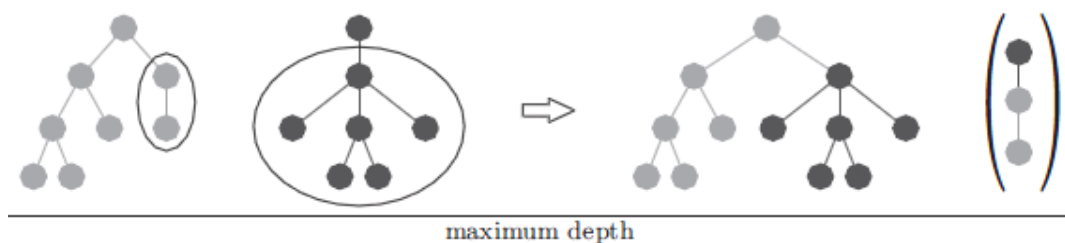


სურათი 41. ინიციალიზაცია „გაზრდის“ (Grow) მეთოდით

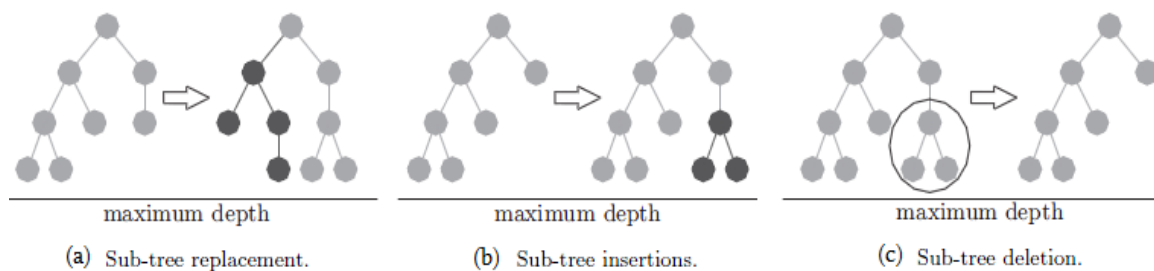
პროგრამის ფიტნეს ფუნქციის შემოწმება: იმისათვის, რომ GP-მ განსაზღვროს თუ როგორ მუშაობს პროგრამა, ის უშვებს ამ პროგრამას და მიღებულ შედეგებს ადარებს ეტალონს. შემდეგ ეს შედარება ფასდება რიცხობრივი მნიშვნელობით რომელსაც „ფიტნეს“ მნიშვნელობა ჰქვია.

სელექცია: „ფიტნეს“ მნიშვნელობაზე დაყრდნობით აირჩევა პროგრამების ჯგუფი რომლებიც რეპროდუქციისათვის გამოიყენება.

პოპულაციის რეპროდუქცია: გენეტიკური ოპერატორების „შეჯვარების“ და „მუტაციის“ გამოყენებით წინა ბიჯზე ალგორითმის მიერ არჩეული პოპულაციის ინდივიდებისაგან ყალიბდება ახალი თაობის პოპულაცია. სურათი 42-ზე მოცემულია „შეჯვარების“ მაგალითი, ხოლო სურათი 43-ზე მოცემულია „მუტაციის“ სახესხვაობები [75].



სურათი 42. GP-ში შეჯვარება



სურათი 43. GP-ში მუტაცია

დასრულების პირობის შემოწმება: მოწმდება ალგორითმის დასრულების წინასწარ განსაზღვრული პირობები (მისაღები ამონახსნი ნაპოვნია თუ არა, ახალი თაობის გენერაციის მაქსიმალურ რაოდენობას გადააჭარბა თუ არა და სხვა).

დასასრული, საუკეთესო ინდივიდების დაბრუნება: ამ ბიჯით ალგორითმი ასრულებს მუშაობას და აბრუნებს მის მიერ ნაპოვნ საუკეთესო ინდივიდს(პროგრამას), მაგრამ როგორც ზემოთ აღვნიშნე არ არსებობს გარანტია, რომ ალგორითმმა იპოვნა გლობალურად საუკეთესო ამონახსნი, არამედ უნდა ჩაითვალოს რომ ეს ამონახსნი არის ოპტიმალური(საუკეთესოსთან მიახლოებული).

4.3 ხელოვნური ნეირონული ქსელები (ANN)

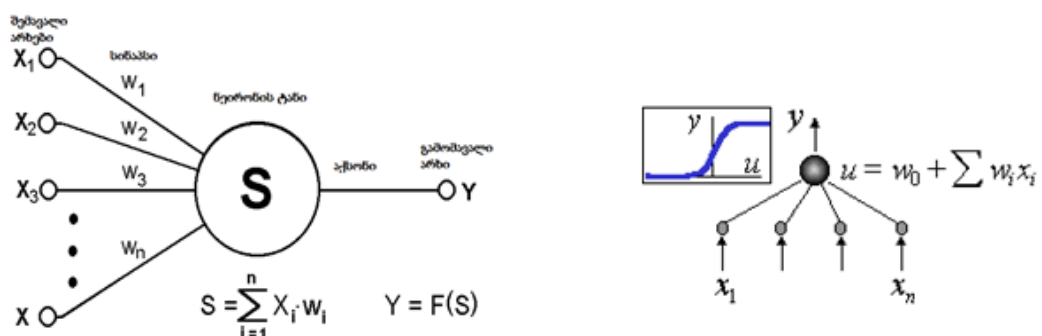
ხელოვნური ნეირონული ქსელი ადამიანის თავის ტვინის ფუნქციონირებაზე დაფუძნებული არაწრფივი სტრუქტურის იმიტაციური მოდელია. მისი ძირითადი შემადგენელი ნაწილები - ხელოვნური ნეირონები ბიოლოგიური ნეირონების ანალოგია. ნეირონი იღებს იმპულსებს სხვა ნეირონების დენდრიტების მეშვეობით და გადასცემს სიგნალებს აქსონის მეშვეობით. დენდრიტების და აქსონების გადამზის წერტილებში მდებარეობს სპეციალური წარმონაქმნი სინაპსი, რომელიც ზემოქმედებს იმპულსების სიდიდეზე. ხელოვნური ნეირონი, ისევე როგორ ბუნებრივი, შედგება სინაპსებისაგან, რომლებიც ნეირონის შესასვლელებს აკავშირებენ ბირთვთან, ნეირონის ბირთვი კი თავის მხრივ ახორციელებს

შემომსვლელი სიგნალების დამუშავებას და გადაცემას სხვა ფენის ნეირონებთან [76, 77].

ხელოვნური ნეირონული ქსელი წარმოადგენს მოდელირების მძლავრ ინსტრუმენტს, რადგან მას შეუძლია შეისწავლოს და დაამუშავოს არაზუსტი და დაზიანებული მონაცემები. ANN-ს წარმატებით იყენებენ არაორდინალური პრობლემების გადასაწყვეტად ისეთ განსხვავებულ სფეროებში როგორებიცაა მედიცინა, ფინანსები, გეოლოგია, ინჟინერია, ფიზიკა და ბიოლოგია. სტატისტიკის კუთხით მას იყენებენ პროგნოზირების და კლასიფიკაციის ამოცანების მოდელირებისთვის. ANN-ის გამოყენებით შექმნეს ისეთი ალგორითმები როგორიცაა ტექსტის გამშიფრავი, ხმის ამომცნობი, სურათიდან გამოსახულების (ტექსტი, ადამიანის სახე, ან სხვა ობიექტი) გამშიფრავი და სხვა.

4.3.1 ნეირონის მათემატიკური მოდელი

ნეირონი არის მარტივი ავტომატი, რომელიც გარდაქმნის შემავალ სიგნალებს გამომავალ სიგნალებში (სურათი 44). სიგნალები ძალებით x_1, x_2, \dots, x_n , ხვდებიან რა სინაპსზე, გარდაიქმნებიან წრფივად, ანუ ნეირონის სხეულზე მიიღება სიგნალები ძალებით $w_1 * x_1, w_2 * x_2, \dots, w_n * x_n$, სადაც w_i არის შესაბამისი სინაპსის წონა.



სურათი 44. ნეირონის მათემატიკური მოდელი

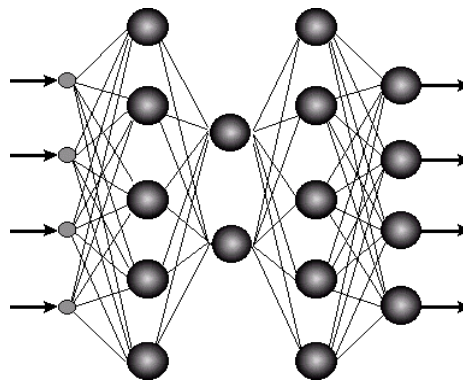
მოსახერხებლობისთვის, ნეირონს უმატებენ კიდევ ერთ შემავალ არხს, წონით w_0 . ითვლება რომ ამ შესასვლელზე ყოველთვის მიეწოდება 1-ის

ტოლი ძალის სიგნალი. ნეირონის ბირთვში ხდება სიგნალების გარდაქმნა შეკრების გზით:

$$S = \sum_{i=0}^n w_i * x_i \quad (26)$$

შემდეგ, ამ ჯამს მიუყენდება რაიმე ფიქსირებულ ფუნქცია f და გამოსავალზე გამოდის სიგნალი ძალით $Y = f(S)$.

ნეირონული ქსელების არქიტექტურა მარტივი და ამასთანავე უნივერსალურია. კავშირების სპეციალიზაცია წარმოიშვება მათი სწავლების ეტაპზე კონკრეტული მონაცემების ზეგავლენით (სურათი 45).



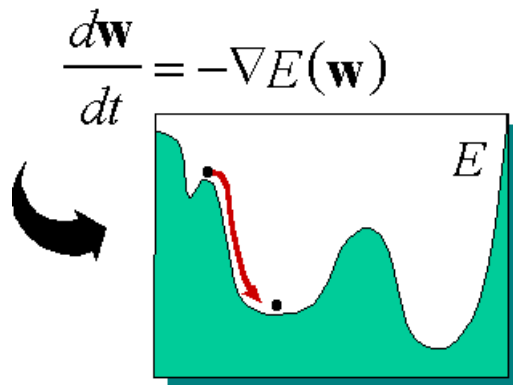
სურათი 45. კავშირების გლობალურობა ხელოვნურ ნეირონულ ქსელებში

ტიპიური ნეირონი ახორციელებს უმარტივეს ოპერაციას - ამრავლებს შემოსულ მონაცემებს თავის, ლოკალურ წონებზე და ახორციელებს არაწრფივ გარდაქმნას მათ ჯამზე. არაწრფივობას მივყავართ იქითკენ, რომ მთლიანი ქსელის შესაძლებლობები ბევრად აღემატება მისი შემადგენელი თითოეული ნეირონის შესაძლებლობებს.

4.3.2 ხელოვნური ნეირონული ქსელის სწავლება

თავდაპირველად, შეცდომის ფუნქცია, რომელიც აფასებს ქსელის მიმდინარე კონფიგურაციას, ხელოვნურ ნეირონულ ქსელს მიეწოდება გარედან იმის მიხედვით თუ რა მიზანს ემსახურება სწავლება. ამის შემდეგ, ქსელი თანდათან იწყებს თავისი კონფიგურაციის მოდიფიკაციას - ყველა თავისი სინაპსის წონები ცვლილება, ისე რომ მინიმუმამდე დავიდეს

შეცდომის ფუნქციის მნიშვნელობა. სწავლების პროცესში ქსელი უფრო და უფრო კარგად ართმევს თავს მასზე დაკისრებულ დავალებას. ეს პროცესი შეიძლება წარმოვიდგინოთ როგორც შეცდომის ფუნქციის $E(w)$ მინიმუმის ძიების პროცესი, რომელიც დამოკიდებულია მის ყველა სინაპსურ w წონეზე (სურათი 46).



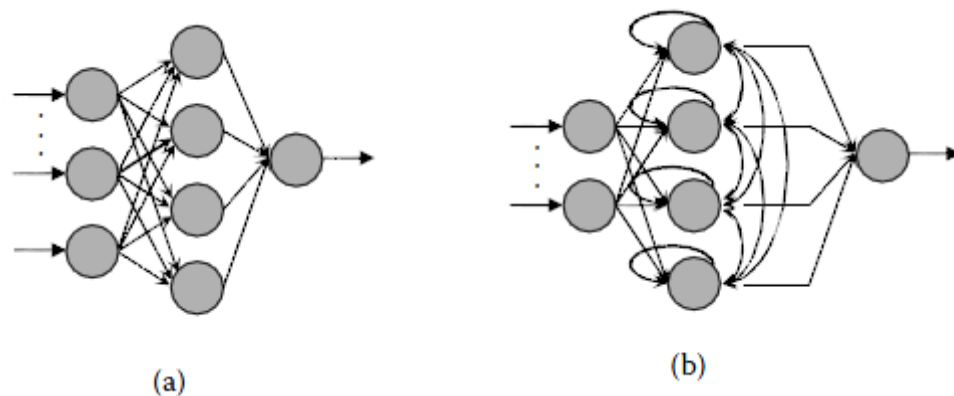
სურათი 46. ქსელის სწავლება როგორც ოპტიმიზაციის ამოცანა

დღესდღეობით ცნობილ სწავლების ალგორითმების უმეტესობას საფუძვლად უდევს გრადიენტული ოპტიმიზაციის მეთოდი - სინაპსური წონების იტერაციული ცვლილება, რომელიც თანდათანობით (ნაბიჯ-ნაბიჯ) ამცირებს ნეირონული ქსელის სწავლების საგნის დამუშავების შეცდომებს. ამასთან, სინაპსის წონების შეცვლა ხდება შეცდომის ფუნქციის ლოკალური გრადიენტის გათვალისწინებით. ამ გრადიენტის ძიების ეფექტურ მეთოდად ითვლება ე.წ. შეცდომის უკუგავრცელების ალგორითმი (error back-propagation). განსხვავებები ქსელის სასურველ და ფაქტიურ შედეგებს შორის ნეირონის გამომავალ შრეზე, ვრცელდება ქსელში სიგნალების შემომავალი ნაკადების მიმართ. საბოლოოდ თითოეულ ნეირონს შესაძლებლობა ეძლევა მარტივად განსაზღვროს თავისი თითოეული წონის წილი ქსელის ჯამური შეცდომის ფუნქციის მნიშვნელობაში. იგი გამოითვლება აღნიშნული სხვაობისა და შესაბამისი შემომავალი სიგნალის მნიშვნელობის ნამრავლით. უმარტივესი სწავლების წესი მდგომარეობს გრადიენტულ ვარდნაში (Gradient descent). სინაპსური წონების ცვლილება პროპორციულია მისი წილის საერთო შეცდომის მნიშვნელობაში. ამდენად, ნეირონული ქსელის

კავშირების ერთი და იგივე სტრუქტურა ეფექტურად გამოიყენება როგორც ქსელის ფუნქციონირებაში ასევე სწავლებაში. ასეთი სტრუქტურა შესაძლებლობას იძლევა გამოითვალოს მიზნობრივი ფუნქციის გრადიენტი თითქმის იმავე სისწრაფით როგორც თვით ეს ფუნქცია. ამასთან ერთად, ორივე სიტუაციაში გამოთვლები არის განაწილებული - თითოეული ნეირონი დამოუკიდებლად ახორციელებს გამოთვლებს, როგორც კი მიიღებს სიგნალს შემავალი ან გამომავალი არხიდან. ამრიგად, სწავლების ალგორითმების უმეტესი ნაწილისთვის ძირითადი იდეა არის კონფიგურაციის არეში ლოკალური გრადიენტის გამოთვლა შეცდომის ფუნქციის უსწრაფესი ვარდნის ტრაექტორიის არჩევისთვის. მაგრამ, შეცდომის ფუნქციას შეიძლება ქონდეს მრავალი ლოკალური მინიმუმი, რომლებიც თავის მხრივ წარმოადგენენ სუბ-ოპტიმალურ ამონახსნს. ასეთ შემთხვევაში, ქსელის კონფიგურაციის ცვლილების შეფერხების თავიდან ასაცილებლად, გრადიენტული მეთოდები მდიდრდება სტოქასტიკური ოპტიმიზაციის ელემენტებით. სწავლების იდეალურმა მეთოდმა სწრაფად უნდა იპოვოს ქსელის კონფიგურაციის გლობალური ოპტიმუმი.

4.3.3 ხელოვნური ნეირონული ქსელის მოდელები

არსებობს ხელოვნური ნეირონული ქსელის(ANN) მოდელის რამდენიმე ტიპი, მაგრამ ორი მათგანი არის ყველაზე მეტად გამოყენებადი (სურათი 47) [76].



სურათი 47. ANN-ის მოდელის ტიპები

1. „პირდაპირი კავშირის“ (Feedforward) ქსელი (სურათი 47 (a)) - ინფორმაციული ნაკადი არის ცალმხრივი და მიმართულია შემავალი დონიდან (input layer) გამომავალი დონისკენ (output layer). აქ არ არსებობს უკუკავშირები ან კავშირები თავის თავთან, ანუ ნებისმიერი დონიდან გამომავალი ინფორმაცია აღარ მოქმედებს იმავე ან მის წინა დონეზე და მიმართულია მომდევნო დონისაკენ.
2. „უკუკავშირის“ (Recurrent) ქსელი (სურათი 47 (b)) - ამ ტიპის ქსელი პირდაპირი კავშირის ქსელისგან განსხვავდება იმით, რომ ამ შემთხვევაში არსებობს ერთი უკუკავშირი მაინც. ასეთ ქსელებში შეიძლება შეგვხვდეს კავშირი თავისთავთან, ანუ ნეირონიდან გამომავალი ინფორმაცია გადაეცემა ისევ ამ ნეირონს როგორც შემავალი ინფორმაცია.

4.4 კოლექტიური ინტელექტი

ბუნებაში მრავლად გვხვდება მარტივი ბიოლოგიური არსებების სიმრავლე (მაგალითად, ჭიანჭველების კოლონია, თაფლის ფუტკრების გროვა, ფრინველების გუნდი, ცხოველების ჯოგი, ბაქტერიული წარმონაქმნები, თევზების გუნდი და ა.შ.), რომლებიც ხასიათდებიან კოლექტიური ქვეცნობიერებით ანუ ე.წ. „გროვის ინტელექტით“, რაც წარმოადგენს ზოგადად დეცენტრალიზებული, თვითორგანიზებადი ბუნებრივი ან ხელოვნური სისტემის ერთობლივი ქმედების საფუძველს. მთავარი იდეა მდგომარეობს იმაში, რომ ინდივიდუალთა ჯგუფი (მაგ. ადამიანები, მწერები, რობოტები ან პროგრამული უზრუნველყოფის აგენტები) შეიძლება იყოს უფრო ჭკვიანი, ვიდრე რომელიმე მისი წევრი. ამასთან, გროვის რთული და ინტელექტუალური ქცევა შეიძლება წარმოიშვას ცალკეულ ინდივიდთა შორის მარტივ წესებზე დაფუძნებული ურთიერთქმედებებით შექმნილი სინერგიისაგან.

კონცეფცია გამოიყენება ხელოვნური ინტელექტის (AI) სისტემებში. SI-სისტემები ჩვეულებრივ შედგება გროვაში შემავალი მარტივი

აგენტებისაგან¹, რომლებიც თავიანთ სპეციფიკურ გარემოში მარტივი ინსტრუქციებით ურთიერთქმედებენ ერთმანეთთან. ამ სისტემაში არ არსებობს ცენტრალიზებული კონტროლის მექანიზმი, რაც იმას ნიშნავს, რომ სისტემაში შემავალი თითოეული აგენტის მოქმედება არის აბსოლუტურად თვითნებური და გარკვეული ხარისხით შემთხვევითი, რადგან მათ არავენ კარნახობს შემდეგ ქმედებებს. ასეთი დამოუკიდებელი აგენტების ქმედება ჯამში იწვევს მთლიანი სისტემის ინტელექტუალურ ქმედებას.

SI არის მულტი-აგენტური სისტემის (MAS) ნაწილი. MAS-ი შეიძლება აღვწეროთ როგორც სისტემა, შედგენილი, ფიზიკური ან ვირტუალური ინდივიდებისაგან, რომლებიც ერთმანეთთან ურთიერთობენ, ცვლიან ინფორმაციასა და ცოდნას და თანამშრომლობენ გარკვეული ამოცანების შესასრულებლად. MAS-ში შემავალი აგენტები შეიძლება იყოს „ინტელექტუალური“ ან „რეაგირებადი“. „ინტელექტუალურ“ აგენტებს გააჩნიათ ინტელექტის გარკვეული დონე, გარკვეული უნარები საკომუნიკაციოდ და თანამშრომლობისათვის, აქვთ მიზანი და „იბრძვიან“ ამ მიზნის მისაღწევად. „რეაგირებად“ აგენტებს არ აქვთ ინტელექტი, ისინი სხვა აგენტებთან ურთიერთქმედებენ გარკვეული წესებით, პირადად ან ირიბად, რეაგირებენ რა სხვადასხვა მოვლენასა თუ პარტნიორის საკომუნიკაციო ნიშანზე.

ზოგადად მულტი-აგენტური სისტემისთვის დამახასიათებელი ნიშნებია:

1. ავტონომიურობა: სისტემაში შემავალი აგენტები სრული თუ არა ნაწილობრივი ავტონომიურობით ხასიათდება.
2. ადგილობრივი (შეზღუდული) ხედვა: არცერთმა აგენტმა არ იცის მთელი სისტემა, ან სისტემა მთლიანობაში იმდენად რთულია, რომ

¹ ხელოვნურ ინტელექტში აგენტი წარმოადგენს ავტონომურ ერთეულს, რომელიც გარკვეულ გარემოში ქმედებისას, დასახული მიზნის მისაღწევად იყენებს საწყის ცოდნას ან სწავლის უნარს.

აგენტი ვერ ჰპოვებს ამ სისტემის მთელი ცოდნის პრაქტიკულ გამოყენება.

3. დეცენტრალიზაცია: სისტემაში არ არსებობს სპეციალურად გამოყოფილი მმართველი აგენტი.
4. თვითორგანიზება და თვითმართვა: მულტი-აგენტური სისტემა წარმოადგენს თვითორგანიზებულ და ასევე თვითმართველ სისტემას, რომელიც ასრულებს რთულ დავალებებს მიუხედავად იმისა, რომ იგი შედგება მარტივი ერთეულებისაგან (აგენტებისაგან).

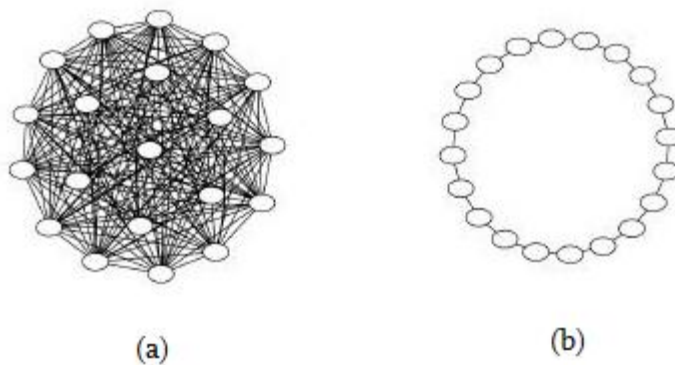
4.5 ნაწილაკების გროვის ოპტიმიზაცია (PSO)

კენედიმ და მენდესმა შეისწავლეს პოპულაციის განსხვავებული ტოპოლოგიები PSO-ს ეფექტურობის დასადგენად. ნაწილაკებს შორის მეზობლობის სხვადასხვა კონცეფციები შეიძლება დადგინდეს. სხვადასხვა კონცეფცია წარმოადგენს ნაწილაკების სხვადასხვა ტოპოლოგიას, სხვადასხვა ტოპოლოგია კი პირდაპირ მოქმედებს ნაწილაკების საკომუნიკაციო შესაძლებლობებზე და საერთო ჯამში, რა თქმა უნდა, მოქმედებს მთელი ჯგუფის (გროვის) ეფექტურობაზე [78].

როგორიც არ უნდა იყოს გროვის ტოპოლოგია, მასში შემავალი ნაწილაკების მეზობლობა ორი ფაქტორით განისაზღვრება:

1. k -განსაზღვრავს ნაწილაკისთვის მეზობელი ნაწილაკების რაოდენობას. დადგინდა, რომ k -ს დიდი მნიშვნელობა უზრუნველყოფს ძიების სიჩქარეს, ხოლო მისი მცირე მნიშვნელობა ზრდის საბოლოო მიზნის მიღწევის შანსს (სურათი 48-ზე მოცემულია ორივე სიტუაცია).
2. C -ნაწილაკის ისეთი მეზობელი ნაწილაკების რაოდენობა, რომლებიც თვითონაც არიან მეზობლები.

ქვემოთ მოცემულია გროვაში ნაწილაკების სხვადასხვა ტოპოლოგია.

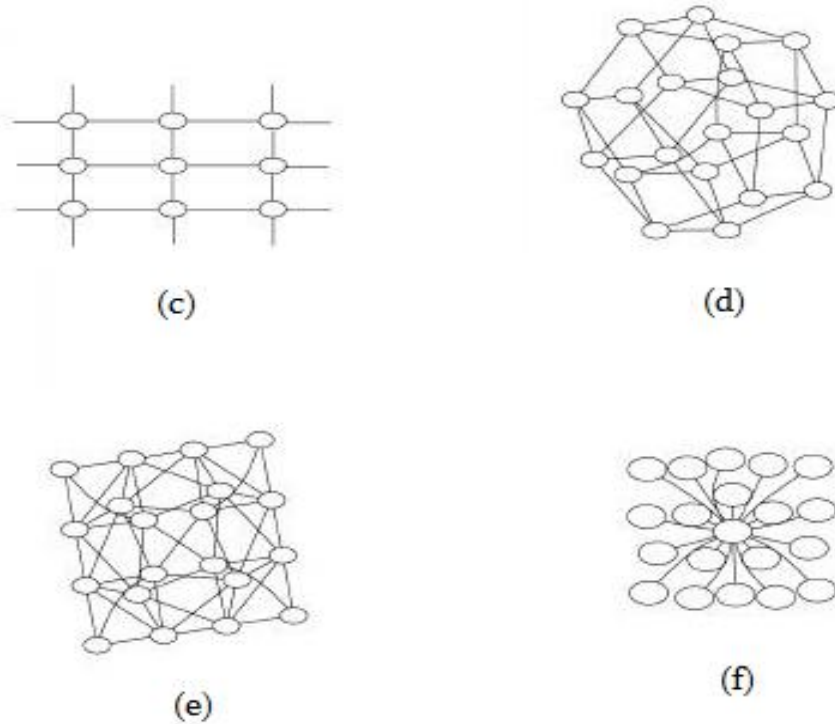


სურათი 48. PSO-ს ტოპოლოგია

სურათი 48 (a)-ზე წარმოდგენილია გლობალური მეზობლობა (neighborhood), სადაც ყოველ ნაწილაკს აქვს წვდომა მეზობელი ნაწილაკის საუკეთესო პოზიციასთან. სწორედ ეს პოზიცია გამოიყენება გროვის სიჩქარის განახლების გამოსათვლელად. ამ წყობას უწოდებენ “gbest”-ს. ამ ტოპოლოგიის გროვა საძიებო არეში სწრაფად პოულობს ლოკალურ ოპტიმუმებს, მაგრამ თუ გლობალური ოპტიმუმი არ მოხვდა საუკეთესო პოზიციასთან ახლოს, მაშინ გროვა შეიძლება დასჯერდეს ლოკალურ ოპტიმუმებს და გლობალური ოპტიმუმის აღმოსაჩენად აღარ შეისწავლოს ახალი არეები საძიებო გარემოში. რაც შეეხება სურათი 48 (b)-ს, აქ გროვის ნაწილაკები წრიულად არიან დაკავშირებული ერთმანეთთან და ყოველმა ნაწილაკმა იცის მხოლოდ მისი წინა და უკანა ნაწილაკის საუკეთესო პოზიცია (lbest). ამგვარად, გროვის სიჩქარეზე ზემოქმედება მხოლოდ რამდენიმე ნაწილაკს თუ შეუძლია. ამ უკანასკნელს უფრო დიდი დრო სჭირდება ერთმანეთში ინფორმაციის გაცვლისა და ოპტიმუმების ძიებისთვის, მაგრამ წინა ტოპოლოგიისგან განსხვავებით, გლობალური ოპტიმუმის მიგნების ბევრად მეტი შანსი აქვს [79].

თუ შევადარებთ გროვის ამ ორი ტოპოლოგიის ნაწილაკების მოქმედებას, მივიღებთ, რომ თუ „gbest“ მოდელი ინახავს მხოლოდ ერთ საუკეთესო ამონახსნს (პოზიციას) და ყველა ნაწილაკი მოძრაობს თავისი წინა საუკეთესო პოზიციისკენ და მთელი გროვის საუკეთესო პოზიციისკენ, „lbest“ მოდელის ყველა ნაწილაკი მოძრაობს თავისი წინა საუკეთესო

პოზიციისკენ და თავისი შეზღუდული რაოდენობის მეზობლების საუკეთესო პოზიციისკენ. მნიშვნელოვანია იმის აღნიშვნაც, რომ „gbest“-ი არის „lbest“-ის კერძო შემთხვევა [78].



სურათი 49. PSO-ს ტოპოლოგია

- ფონ ნეიმან (Von Neumann)-ის ტოპოლოგია: სურათი 49 (c) წარმოადგენს ერთ სიბრტყეზე ნაწილაკების კავშირს (მეზობლობას), ხოლო სურათი 49 (d) მის სამგანზომილებიან ანალოგს.
- პირამიდული ტიპის ტოპოლოგია (სურათი 49 (e)), სადაც თითოეულ ნაწილაკთან მისი მეზობელი ნაწილაკები ქმნიან პირამიდულ განლაგებას.
- ვარსკვლავური ტოპოლოგია (სურათი 49 (f)), სადაც ყველა ნაწილაკი დაკავშირებულია ერთ, ცენტრალურ ნაწილაკთან ე.წ. ლიდერთან (leader) და ნებისმიერი კომუნიკაცია ნაწილაკებს შორის შესაბამისად ხდება ლიდერის გავლით.

დასკვნები იმაზე, თუ რომელი ტოპოლოგია ჯობია, დამოკიდებულია ალგორითმის პარამეტრების შერჩევაზე, მაგრამ ზოგადად დადგინდა, რომ

„Von Neumann“-ის და პირამიდული ტოპოლოგიის მქონე გროვა არის საუკეთესო, ხოლო „gbest“-ი არის ყველაზე ცუდი. ზემოთ განხილული ტოპოლოგიები იყო სტატიკური (static). ნაწილაკების გროვის მოდელში შემოიტანეს ასევე დინამიური (dynamic) ტოპოლოგიის კონცეფცია, რაც იმას ნიშნავს, რომ გროვაში ნაწილაკების განლაგება მუდმივად იცვლება დროში შემდეგი სახით:

- თავდაპირველად ნაწილაკებს აქვს „lbest“-ის განლაგება (სურათი 48 (b)), დროთა განმავლობაში ნაწილაკს ემატება მეზობლები და საბოლოოდ მისი გროვის ტოპოლოგია გარდაიქმნება „gbest“-ად (სურათი 48 (a)).
- თითოეული ნაწილაკისთვის მეზობლად ნაწილაკის არჩევა ხდება თითოეულ იტერაციაზე ნაწილაკსა და მის პოტენციურ მეზობელ ნაწილაკს შორის „ჰემინგის მანძილის“ (Hamming distance)¹ გათვალისწინებით.

ამ ტოპოლოგიის ძირითადი მინუსი ის არის, რომ საჭიროა დამატებითი გამოთვლები, ანუ თითოეული ნაწილაკისათვის უნდა გამოითვალოს მეზობელი ნაწილაკები ყოველ იტერაციაზე [79].

4.5.1 PSO-ს პარამეტრები

PSO-ს ძირითად პარამეტრებს წარმოადგენს ინერცია w , c_1 , c_2 , v_{max} მაქსიმალური სიჩქარე და გროვის ზომა N . ამ პარამეტრების შერჩევით განისაზღვრება ალგორითმის ოპტიმიზაცია საძიებო არეში. პარამეტრის ერთი და იგივე მნიშვნელობამ შეიძლება სხვადასხვა ეფექტი მოგვცეს სხვადასხვა ტიპის ამოცანებში. ამიტომ საჭიროა ვიცოდეთ თითოეული პარამეტრის თვისებები და თუ რა გავლენას ახდენს იგი ალგორითმის

¹ „ჰემინგის მანძილი“-ორ ერთი და იგივე სიგრძის სტრიქონში პოზიციურად ტოლი განსხვავებული სიმბოლოების რაოდენობა.

მოქმედებაზე, რომ სწორად შევარჩიოთ იგი კონკრეტული შემთხვევისათვის [80].

ინერციის მაჩვენებელი

ინერცია გამოიყენება ნაწილაკის წინა ბიჯზე არსებული სიჩქარის მიმდინარე სიჩქარესთან ზემოქმედების შესაფასებლად და საკონტროლოდ. ეს პარამეტრი არეგულირებს გლობალური და ადგილობრივი საძიებო გარემოს შესწავლის უნარებს შორის ბალანსს. ინერციის დიდი მაჩვენებელი ამარტივებს გლობალურ ძიებას (საძიებო არეში ახალი არეების აღმოჩენა), ხოლო პატარა მაჩვენებელი უზრუნველყოფს ლოკალურ ძიებას, ანუ აზუსტებს საძიებო წერტილს ლოკალურ არეალში. ინერციის მაჩვენებლის სწორი შერჩევა, როგორც წესი ბალანსს ამყარებს გლობალურ და ლოკალურ ძიებას შორის და ამცირებს იტერაციების რაოდენობას, რომელიც საჭიროა ფუნქციის ოპტიმუმის დასადგენად. ცდებით დადგინდა, რომ უკეთესია დასაწყისში ინერციის მაჩვენებელს მივანიჭოთ დიდი მნიშვნელობა, რომ მივიღოთ ძიების ფართო არეალი და თანდათან შევამციროთ იგი, რათა მივიღოთ უფრო ზუსტი ამონახსნი. მაგალითად, კარგ გადაწყვეტილებად ითვლება თუ თავდაპირველად w -ს მივანიჭებთ 1.2-ს და თანდათან შევამცირებთ მის მნიშვნელობას 0-მდე. საუკეთესო მიდგომაა, თუ გამოვიყენებთ ადაპტურ მეთოდებს ინერციის შესამცირებლად, რადგან ასეთი მეთოდები ამოცანის ტიპისა და ალგორითმის მიმდინარე მდგომარეობის გათვალისწინებით დააკორექტირებენ ინერციის მაჩვენებლის მნიშვნელობას. ერთ-ერთი ასეთი მეთოდია „fuzzy controller“, რომელიც PSO-ს შესძენს თვით ადაპტირების უნარს და ალგორითმი თვითონ დააკორექტირებს ინერციის მაჩვენებელს [80, 81].

c_1 და c_2 პარამეტრები

c_1 და c_2 პარამეტრები არ არის კრიტიკული SPO-სათვის, მაგრამ ამ პარამეტრების სწორად შერჩევამ შეიძლება მიგვიყვანოს შედეგის სწრაფად

მოძებნამდე ლოკალური მინიმუმების ძიებისას. როგორც წესი, $c_1 = c_2 = 2$, მაგრამ გარკვეულმა ტესტებმა აჩვენეს რომ გარკვეულ შემთხვევებში $c_1 = c_2 = 1.49$ უფრო კარგ შედეგებს იძლევა. ბოლო შედეგების მიხედვით ყველაზე კარგ რეზულტატს იძლევა თუ c_1 -ს ავირჩევთ უფრო დიდს ვიდრე c_2 -ს შემდეგი პირობით $c_1 + c_2 = 4$ [80, 81].

მაქსიმალური სიჩქარე

ნაწილაკის მაქსიმალური სიჩქარე v_{max} -ი განსაზღვრავს, თუ რა მაქსიმალური მანძილის გავლას შეძლებს ნაწილაკი წინა პოზიციიდან ერთი იტერაციის განმავლობაში. როგორც წესი, v_{max} -ის ზომად იღებენ მთელი საძიებო არის ზომას. მაგალითად, თუ ნაწილაკის პოზიცია განისაზღვრება შემდეგი ვექტორით $\vec{x} = (x_1, x_2, x_3)$, სადაც $-10 \leq x_i \leq 10$, $i = 1, 2, 3$, მაშინ მაქსიმალური სიჩქარე უნდა იყოს $v_{max} = 20$ [80].

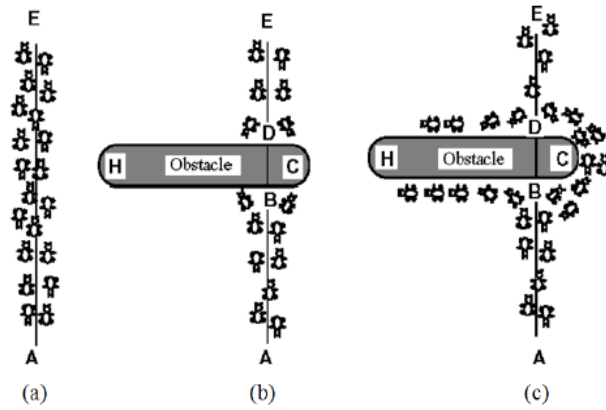
გროვის ზომა

გროვის ზომა დროის მოცემულ მომენტში ნაწილაკებისათვის ფიქსირებული სიდიდეა. ერთი შეხედვით შეიძლება ვიფიქროთ, რომ რაც უფრო მეტია გროვაში ნაწილაკი, მით უფრო ცოტა იტერაცია დაჭირდება საძიებელი შედეგის მიღებას. მაგრამ იტერაციების რაოდენობა ამ შემთხვევაში არ არის ეფექტურობის შეფასების კარგი კრიტერიუმი. უფრო მნიშვნელოვანია, თუ რამდენჯერ მოხდება იმ ფუნქციის შეფასება, რომლის მინიმუმსაც ვეძებთ, რადგან რეალური ამოცანების უმრავლესობაში ამ შეფასებას საკმაოდ დიდი დრო მიაქვს. ნათელია, რომ ერთი იტერაციისათვის ასეთი შეფასებების რაოდენობა უდრის გროვაში ნაწილაკების რაოდენობას, ამიტომ თუ გვსურს, რომ შევამციროთ ფუნქციის შეფასების საერთო რაოდენობა სანამ მივაღწევთ შედეგს, იძულებულები ვართ შევამციროთ ნაწილაკების რაოდენობა გროვაში; მეორეს მხრივ, ძალიან მცირე გროვა ამონახსნის ძიებისას წაიღებს დიდ დროს ან საერთოდ ვერ

შეძლებს მის პოვნას. ერთი სიტყვით, უნდა მივალწიოთ კომპრომისს. გამოცდილებამ აჩვენა, რომ თითქმის ყველა კლასიკური ამოცანის ამოსახსნელად საკმარისია გროვა, რომელიც შედგება 20-დან 30-მდე ნაწილაკისაგან [82].

4.6 ჭიანჭველების კოლონიის ოპტიმიზაცია (ACO)

ჭიანჭველების კოლონიის ოპტიმიზაციის (Ant Colony Optimization - ACO) ალგორითმი წარმოადგენს გამოთვლითი პრობლემის ამოხსნის ალბათურ მეთოდს, რომელიც გამოიყენება გრაფზე საუკეთესო გზის განსაზღვრისათვის. ამ მეთოდის შექმნას საფუძვლად დაედო ჭიანჭველების ქცევა, როდესაც ისინი პოულობენ გზას საკვებისაკენ. სურათი 50 ასახავს ნამდვილი ჭიანჭველების ქცევას. სურათი 50 (ა) გვიჩვენებს გზას, რომელიც გაკვალულია ჭიანჭველების მიერ A კვების წყაროსა და E ბუდეს შორის. როდესაც გზა მოჭრილია დაბრკოლების მიერ, როგორც ნაჩვენებია სურათი 50 (ბ), ჭიანჭველები, რომლებიც იმყოფებიან B წერტილში A-დან E-სკენ მიმავალი და ისინი, რომლებიც იმყოფებიან D წერტილში E -დან A-ში მიმავალი, იძულებული არიან აირჩიონ ერთ-ერთი გზა, რომელიც გადის ან C ან H წერტილზე. სანამ არ გაჩნდება ფერომონის კვალი ერთ ან ორ ალტერნატიულ გზაზე, პირველი ჭიანჭველების მიერ ერთ-ერთი გზის შერჩევა იქნება თანაბარალბათური. ვინაიდან BCD გზა უფრო მოკლეა BHD გზასთან შედარებით, ჭიანჭველა, რომელიც ირჩევს გზას C წერტილის გავლით, D წერტილში მივა უფრო ადრე, ვიდრე ჭიანჭველა, რომელმაც აირჩია გზა H წერტილის გავლით. მაშასადამე, ჭიანჭველა, რომელიც ბრუნდება E-დან A-ში და იმყოფება D წერტილში, იპოვის უფრო ძლიერ კვალს DCB გზაზე.



სურათი 50. ჭიანჭველების კოლონის გზა საკვების მოსაპოვებლად

ჭიანჭველების ქცევა შეესაბამება მათ არჩევენს BCD გზის გავლით. შესაბამისად, DCB გზის არჩევის ალბათობა იქნება უფრო მეტი, ვიდრე DHB გზისა. შედეგად, ფერომონის რაოდენობა BCD გზაზე გაიზრდება სწრაფად, ვიდრე BHD გზაზე, რადგან მეტი რაოდენობის ჭიანჭველები მიყვებიან BCD გზას დროის ერთეულში. გარკვეული დროის შემდეგ უკვე ყველა ჭიანჭველა აირჩევს უმოკლეს გზას.

ჭიანჭველების კოლონის ოპტიმიზაციის მეთოდი, რომელიც ასახავს ჭიანჭველების ბუნებრივ ქცევას, გამოიყენება კომბინატორული ოპტიმიზაციის ამოცანების გადასაწყვეტად ე.წ. ხელოვნური ჭიანჭველების მეშვეობით.

რეალურ სამყაროში ჭიანჭველები რანდომიზებულად მოძრაობენ და როცა პოულობენ საკვებს, თავიანთ კოლონაში დაბრუნებისას ტოვებენ ფერომონის ნაკვალევს. თუ სხვა ჭიანჭველებიც პოულობენ ასეთ გზას, ისინი უკვე აღარ აგრძელებენ რანდომიზებულ მოძრაობას, არამედ, ნაცვლად გაკვალვის გაგრძელებისა, ბრუნდებიან და განამტკიცებენ მას თუ საბოლოოდ იპოვეს საკვები.

ჭიანჭველა მოძრაობს i კვანძიდან j კვანძისაკენ შემდეგი ალბათობით

$$p_{i,j} = \frac{(\tau_{ij}^\alpha)(\eta_{ij}^\beta)}{\sum (\tau_{ij}^\alpha)(\eta_{ij}^\beta)} \quad (27)$$

სადაც:

τ_{ij} არის ფერომონის რაოდენობა i,j რკალზე;

α არის τ_{ij} ზეგავლენის მართვის პარამეტრი;

η_{ij} არის i,j რკალის მიზანშეწონილობა;

β არის η_{ij} ზეგავლენის მართვის პარამეტრი.

ფერომონის განახლება:

$$\tau_{i,j} = \rho \tau_{i,j} + \Delta \tau_{i,j} \quad (28)$$

სადაც:

$\tau_{i,j}$ არის ფერომონის რაოდენობა მოცემულ i,j რკალზე;

ρ არის ფერომონის აორთქლების შეფასება;

$\Delta \tau_{i,j}$ არის დაგროვილი ფერომონის რაოდენობა, რომელიც ჩვეულებრივ მოცემულია შემდეგი პირობით

$$\Delta \tau_{i,j}^k = \begin{cases} 1/L_k & \text{if ant } k \text{ travels on arc } i,j \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

სადაც: L_k არის k -ური ჭიანჭველის მოგზაურობის ფასი (ანუ გზის სიგრძე) [83, 84].

4.7 ამოწვის მეთოდი (Annealing)

ამოწვის იმიტაციის ალგორითმი (Simulated annealing) არის ზოგადი ალგორითმული მეთოდი, გლობალური ოპტიმიზაციის და განსაკუთრებით დისკრეტული და კომბინატორული ოპტიმიზაციის ამოცანების ამოსახსნელად. მისი ერთ-ერთი მაგალითია მონტე - კარლოს მეთოდი.

ალგორითმი დაფუძნებულია ფიზიკური პროცესის იმიტაციაზე, რომელსაც ადგილი აქვს ნივთიერების კრისტალიზაციისას ანუ მისი თხევადი მდგომარეობიდან მყარ მდგომარეობაში გადასვლის დროს, ასევე მეტალების გამოწვის დროს. გამოწვის დროს ატომები განლაგდებიან კრისტალურ მესერში, თუმცა ჯერ კიდევ დასაშვებია ცალკეული ატომების გადასვლა ერთი უჯრედიდან მეორეში. მიღებულია რომ ეს პროცესი მიმდინარეობს მუდმივად კლებადი ტემპერატურის ვითარებაში. ატომის

გადასვლა ერთი უჯრედიდან მეორეში ხდება რაღაც ალბათობით და ეს ალბათობაც მცირდება ტემპერატურის შემცირებასთან ერთად. მყარი კრისტალური მესერი შეესაბამება ატომის მინიმალურ ენერგიას, ამდენად ატომი ან გადადის უფროს დაბალი დონის ენერგიის მდგომარეობაში ან რჩება ადგილზე.

ასეთი პროცესის მოდელირების დასახმარებლად იძებნება ისეთი წერტილი ან წერტილთა სიმრავლე, რომელზეც მიიღწევა მინიმუმი შემდეგი რიცხვითი ფუნქციის $F(\bar{x})$ სადაც $\bar{x} = (x_1, \dots, x_m) \in X$. შემოგვყავს წერტილების მიმდევრობა $\bar{x}_0, \bar{x}_1, \dots, \bar{x}_n$ სივრცეში X . ალგორითმი პოულობს მომდევნო წერტილს დაწყებულს \bar{x}_0 , რომელიც არის მიახლოების დასაწყისი. ალგორითმი ჩერდება \bar{x}_n წერტილის მიახლოებისთანავე.

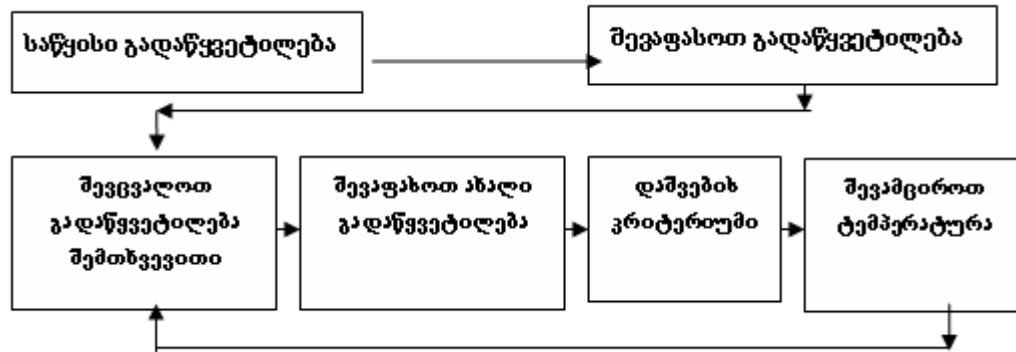
წერტილი \bar{x}_{i+1} ალგორითმის მიხედვით მიიღება წერტილის \bar{x}_i -ის საფუძველზე შემდეგი გზით. წერტილისთვის \bar{x}_i გამოიყენება ოპერატორი A , რომელიც შემთხვევითი არჩევით შეარჩევს შესაბამის წერტილს, რის შედეგადაც მიიღება ახალი წერტილი \bar{x}^* . წერტილი \bar{x}^* გადაიქცევა წერტილად \bar{x}_{i+1} ალბათობით $P(\bar{x}^*, \bar{x}_{i+1})$, რომელიც გამოითვლება გიბსის განაწილების შესაბამისად:

$$P(\bar{x}^* \rightarrow \bar{x}_{i+1} | \bar{x}_i) = \begin{cases} 1, & F(\bar{x}^*) - F(\bar{x}_i) < 0 \\ \exp\left(-\frac{F(\bar{x}^*) - F(\bar{x}_i)}{Q_i}\right), & F(\bar{x}^*) - F(\bar{x}_i) \geq 0 \end{cases} \quad (30)$$

სადაც $Q_i > 0$ არის კლებადი წარმოებულის ელემენტი, რომელიც მიისწრაფვის ნულისაკენ. კლებადობის სიჩქარე და კლებდობის წესს არეგულირებს სურვილის მიხედვით ალგორითმის შემქმნელი.

ამოწვის იმიტაციის ალგორითმი მსგავსია გრადიენტული დადაბლების მეთოდის. თუმცა შუალედური წერტილის შემთხვევითი არჩევის ხარჯზე მისი მოხვედრა ლოკალურ მინიმუმში ხდება უფრო იშვიათად, ვიდრე გრადიენტულ დადაბლების მეთოდში. ამოწვის იმიტაციის ალგორითმი არ იძლევა ფუნქციის მინიმუმის მოძებნის გარანტიას, მაგრამ X სივრცეში შემთხვევითი წერტილის გენერაციის სწორი

პოლიტიკის არჩევის შედეგად, როგორც წესი, ხდება საწყისი მიახლოების გაუმჯობესება. ამოწვის იმიტაციის ალგორითმის ბლოკ-სქემას აქვს შემდეგი სახე (სურათი 51):



სურათი 51. ამოწვის იმიტაციის ალგორითმი

4.8 იმპერიალისტური მეთოდი (ICA)

იმპერიალისტური შეჯიბრებითი ალგორითმი (ICA) ეფუძნება იმპერიალისტურ მეტოქეობას, სადაც ძლიერი სახელმწიფოები ქმნიან კოლონიებს სხვადასხვა ტერიტორიებზე და ამასთან ებრძვიან სხვა ძლიერ სახელმწიფოებს ძალაუფლებისთვის და ახალი მიწებისთვის. ესეთი მეტოქეობის დროს ისინი უფრო ძლიერდებიან და ან კარგავენ ძალაუფლებას. აღნიშნულ მეთოდს 2007 წელს საფუძველი ჩაუყარა ორმა ირანელმა მეცნიერმა Esmaeil და Caro-მ. იგი არის ახალი სოციოპოლიტიკურად მოტივირებული გლობალური ძიების სტრატეგია, რომელიც თავიდანვე ერთ-ერთი ძლიერ ევოლუციურ ალგორითმად იქნა აღიარებული. იგი გამოიყენება სხვადასხვა ოპტიმიზაციის ამოცანების გადასაწყვეტად როგორც ცალკე ასევე სხვა ევოლუციურ ალგორითმებთან კომპოზიციაში. მაგალითად ფართოდ გამოიყენება სინაპსის წონების ოპტიმიზაციის მიზნით ხელოვნურ ნეირონულ ქსელებში (ANN), კლასტერიზაციის ამოცანებში k-Means ალგორითმთან ერთად (K-MICA), ასევე შემდეგი თაობის ვებ ტექნოლოგიებში ინფორმაციის სემანტიკური ძებნისთვის (Semantic Web Searching) და ა.შ.

განვიხილოთ ტერმინოლოგია, რომელიც პირობითად ასახავს ალგორითმის არსს:

- ქვეყანა - სახელმწიფო, რომელიც მონაწილეობს იმპერიალისტურ პაექრობაში როგორც კოლონია ან როგორც იმპერია.
- იმპერია - სახელმწიფო, რომელიც ებრძვის სხვა სახელმწიფოებს კოლონიების მართვის უფლების მოსაპოვებლად.
- კოლონია - სახელმწიფო, რომელიც მოხვედრილია იმპერიის მართველობის ქვეშ.
- ქვეყნის ძალა(სიძლიერე) - ქვეყნის ხელსაყრელი მდგომარეობა სხვა ქვეყნებთან შედარებით. მათემატიკური თვალსაზრისით ეს ნიშნავს ფუნქციის ოპტიმალური მნიშვნელობასთან სიახლოვეს.

4.8.1 მეთოდის აღწერა

თავდაპირველად ხდება ქვეყნების N_{pop} - ზომის პოპულაციის გენერირება. შემდეგ ამ პოპულაციიდან ირჩევა N_{imp} - რაოდენობის ყველაზე ძლიერი ქვეყნები იმპერიების შესაქმნელად. დარჩენილი N_{col} ქვეყნები კი გახდებიან იმპერიის კოლონიები. კოლონიების იმპერიებზე გადანაწილება ხდება იმპერიების სიძლიერის(ძალის) პროპორციულად. იმპერიის სიძლიერის გამოსათვლელად უნდა შემოვიტანოთ იმპერიის ნორმალიზებული წონა:

$$C_n = c_n - \max_i \{c_i\} \quad (31)$$

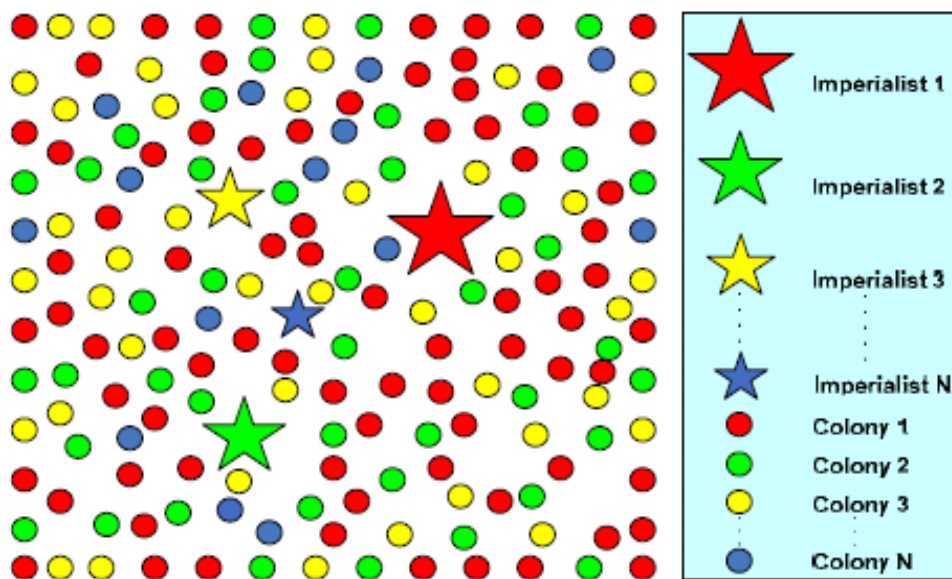
სადაც C_n არის n -ური იმპერიის წონა. მას შემდეგ რაც ცნობილია იმპერიის ნორმალიზებული წონა შეგვიძლია გამოვთვალოთ მისი ნორმალიზებული ძალა:

$$p_n = \frac{C_n}{\sum_{i=1}^{N_{imp}} C_i} \quad (32)$$

როგორც უკვე ვთქვით, საწყის ეტაპზე იმპერიის ძალა განსაზღვრავს მისი კოლონიების პორციას მთელი პოპულაციიდან. შესაბამისად, იმპერიის კოლონიების რაოდენობა ინიციალიზაციის დროს იქნება:

$$N.C_n = \text{round}\{p_n \cdot N_{col}\} \quad (33)$$

სურათი 52-ზე მოცემულია იმპერიების და კოლონიების გადანაწილების მაგალითი. სურათზე ჩანს, რომ რაც უფრო ძლიერია იმპერია მით მეტი კოლონია ერგო ინიციალიზაციის შემდეგ.



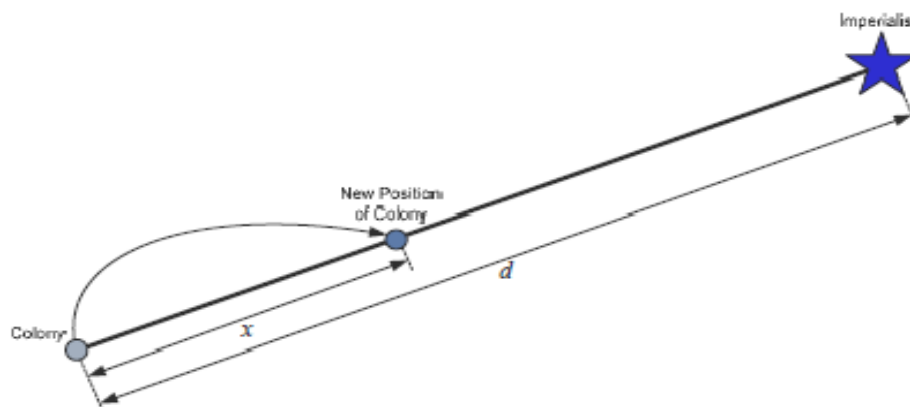
სურათი 52. იმპერიებისა და კოლონიების განაწილება

კოლონიების გადანაწილების შემდეგ, რადგან მათაც გააჩნიათ გარკვეული ძალა, რა თქმა უნდა შეიცვლება იმპერიების სიძლიერე. შედეგად იმპერიის მთლიანი ძალა იქნება იმპერიის ძალას დამატებული თავისი კოლონიების პროცენტული ძალა:

$$power_n = \text{cost}(\text{imperialists}_n) + \varepsilon \text{mean}\{(\text{cost}(\text{colonies of empires}_n))\} \quad (34)$$

მოდელის ინიციალიზაციის შემდეგ იწყება ალგორითმის შემდეგი ეტაპი. იმპერიები იწყებენ ბრძოლას ერთმანეთში რათა დაიპყრონ მეტი კოლონია და გააფართოვონ თავიანთი ტერიტორია. ეს ფაქტი მოდელირებულია კოლონიების მოძრაობაში თავიანთი იმპერიალისტის მიმართ. სურათი 53 - ზე ნაჩვენებია კოლონიის გადაადგილება x -

ერთეულით. გადაადგილება არის ვექტორი რომელიც მიმართულია იმპერიალისტისკენ.



სურათი 53. კოლონიის მოძრაობა იმპერიალისტის მიმართ

x არის შემთხვევითი რიცხვი რომელიც მნიშვნელობას: $x \sim U(0, \beta \times d)$, სადაც d არის მანძილი კოლონიასა და იმპერიალისტს შორის, β კი არის რიცხვი ყოველთვის მეტი ერთზე - $\beta > 1$.

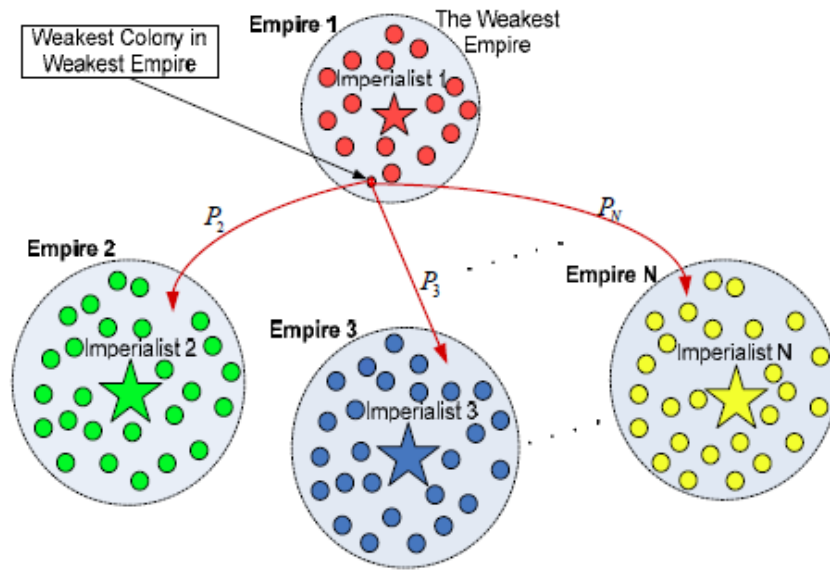
კოლონიის იმპერიალისტისკენ მოძრაობის დროს შეიძლება კოლონია მოხვდეს ისეთ პოზიციაში როცა მისი წონა აღემატება იმპერიალისტის წონას. ასეთ შემთხვევაში კოლონია და იმპერიალისტი ცვლიან როლებს და კოლონია ხდება იმპერიალისტი და პირიქით (სურათი 54).



სურათი 54. კოლონიის და იმპერიალისტის როლების შეცვლა

როგორც ზემოთ აღვნიშნეთ ალგორითმი დაფუძნებულია იმპერიების პაექრობაზე. იმპერიები ცდილობენ დაეუფლონ სხვა იმპერიების კოლონიებს. ამ პაექრობის შედეგად სუსტი იმპერიები კიდევ უფრო

სუსტდება, ხოლო ძლიერები კიდევ უფრო ძლიერდება. პაექრობა მოდელირებულია შემდეგნაირად: ირჩევა ყველაზე სუსტი იმპერიის ყველაზე სუსტ კოლონი და იწყება ბრძოლა ამ კოლონიის დასაპყრობად. ალგორითმის მიხედვით არ არის აუცილებელი რომ ყველაზე ძლიერმა იმპერიამ დაიპყროს აღნიშნული კოლონია, მაგრამ რა თქმა უნდა ამისი შანსი დიდია. სურათი 55-ზე მოცემულია იმპერიების პაექრობა კოლონიის დაპყრობის მიზნით.



სურათი 55. იმპერიების პაექრობა კოლონიების დაპყრობის მიზნით

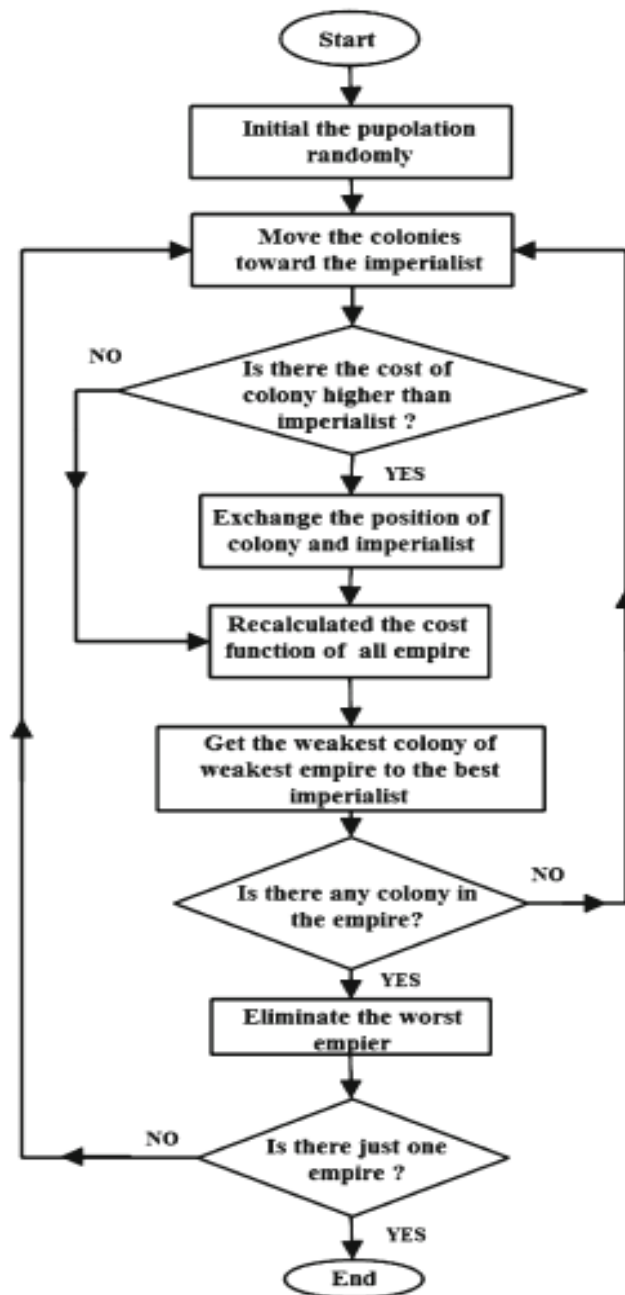
იმის დასადგენად თუ რომელი იმპერია მიითვისებს აღნიშნულ კოლონიას, თითოეული იმპერიისთვის უნდა გამოითვალოს ალბათობა ნორმალიზებულ საერთო ძალაზე დაყრდნობით.

$$N.T.C._n = T.C._n - \max\{T.C._i\} \quad (35)$$

სადაც $T.C._n$ და $N.T.C._n$ არის n -ური იმპერიის საერთო და საერთო ნორმალიზებული წონები. ალბათობა იმისა რომ კონკრეტული იმპერია დაიპყრობს კოლონიას გამოითვლება ფორმულით:

$$P_{p_n} = \left| \frac{N.T.C._n}{\sum_{i=1}^{N_{\text{emp}}} N.T.C._i} \right| \quad (36)$$

სუსტი იმპერიებისგან კოლონიების მითვისების პროცესი გრძელდება იქამდე სანამ იმპერიაში არ დარჩება მარტო იმპერიალისტი. ამის შემდეგ კი იმპერია წყვეტს არსებობას. ამგვარად ალგორითმის მუშაობის პროცესში მუდმივად ხდება სუსტი იმპერიების ჩაყლაპვა და პროცესი იქამდე გრძელდება სანამ ერთი ყველაზე ძლიერი იმპერია არ დარჩება. ალგორითმის მუშაობის მთლიანი პროცესის ძირითადი ბიჯები მოცემულია ბლოკ-სქემის სახით სურათი 56 - ზე [85, 86].

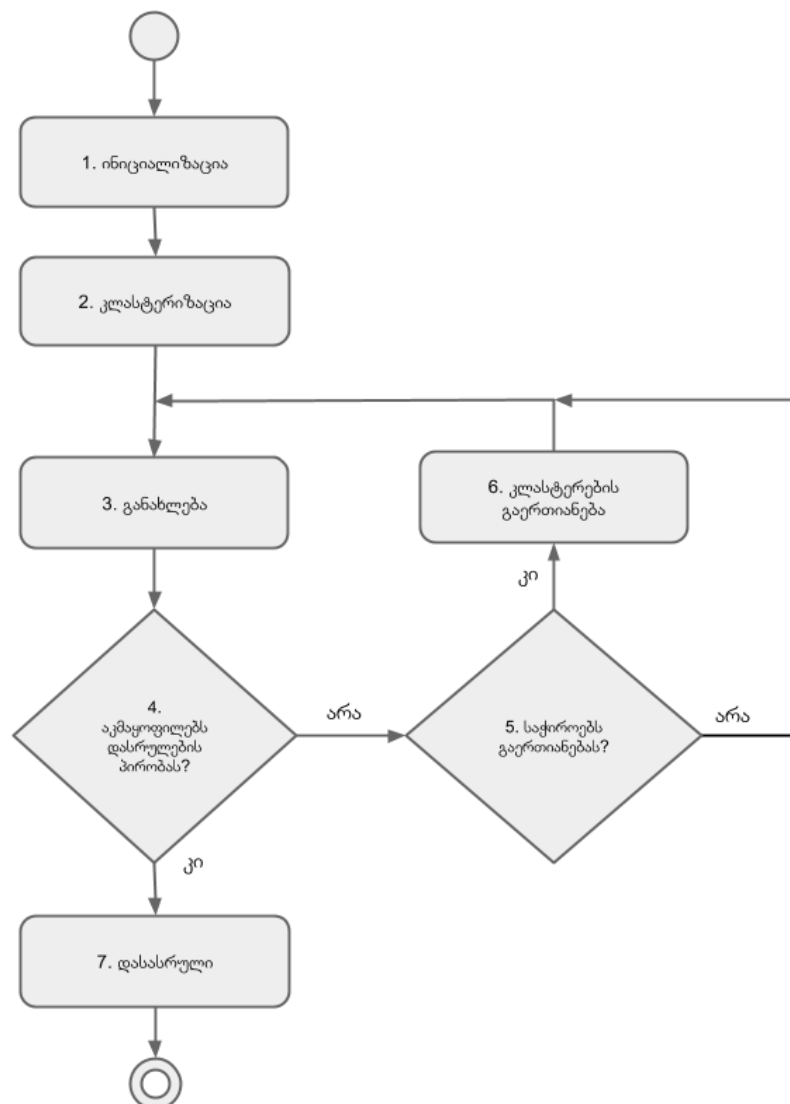


სურათი 56. ICA – ს ბლოკ-სქემა

თავი 5 მულტი-რობოტული სისტემების მართვის ახლებური მიდგომა

5.1 ალგორითმის აღწერა

PSO-ს პარადიგმების მიხედვით, სამიეზო არეში მოძრავი გროვის ნებისმიერი ნაწილაკი წარმოადგენს ოპტიმიზაციის ამოცანის პოტენციურ ამონახსნს. ალგორითმის მიზანია ნაწილაკების შეგროვება მიზნობრივი ფუნქციის გლობალურ (მთელი სამიეზო არე) და ლოკალურ (კონკრეტული კლასტერი) ოპტიმუმებში. ალგორითმის მუშაობის სქემა მოიცავს რამდენიმე ბიჯს (სურათი 57):



სურათი 57. სენსორული რობოტების ჯგუფის მართვის ბლოკ-სქემა

5.1.1 ინიციალიზაცია

თითოეულ ნაწილაკს სამი თვისება გააჩნია:

პოზიცია p_k^i - i -ური ნაწილაკის პოზიცია k -ურ დროში ან ბიჯზე განისაზღვრება კოორდინატებით:

$$p_k^i = [x_k^i, y_k^i], \quad i = 1, 2, \dots, N. \quad (37)$$

სიჩქარე v_k^i - ნაწილაკები ოპტიმუმის ძიების პროცესში განუწყვეტლივ მოძრაობენ საძიებო არეში. მოძრაობა განისაზღვრება მიმდინარე კოორდინატებიდან სწორად განსაზღვრული მანძილით პოზიციის ცვლილებით.

ფიტნეს ფუნქცია $f(p_k^i)$ - განსაზღვრავს თუ რომელ ნაწილაკს აქვს საუკეთესო მნიშვნელობა გროვაში და ასევე, ნაწილაკის საუკეთესო პოზიციას დროის ან იტერაციების მანძილზე.

ნაწილაკების გროვა შედგება ზემოთ აღწერილი ნაწილაკების სიმრავლისგან:

$$P_k = \{p_k^i\}, \quad i = 1, 2, \dots, N. \quad (38)$$

ინიციალიზაციის დროს უნდა განისაზღვროს საწყისი პარამეტრები: გროვის ზომა N , წონის პარამეტრები c_1 , c_2 , რომელთაც შემეცნებით და სოციალურ პარამეტრებსაც უწოდებენ, საძიებო არის საზღვრები x_{min} , x_{max} , y_{min} , y_{max} და ალგორითმის შეწყვეტის კრიტერიუმები G , μ . რა თქმა უნდა, თავდაპირველად $k \leftarrow 0$. ამის შემდეგ, წინასწარ განსაზღვრულ არეში, უნდა დავსვათ N რაოდენობის ნაწილაკი შემთხვევითად დაგენერირებულ პოზიციებზე (სურათი 58).

$$p_0^i = p_{min} + rand(p_{max} - p_{min}). \quad (39)$$

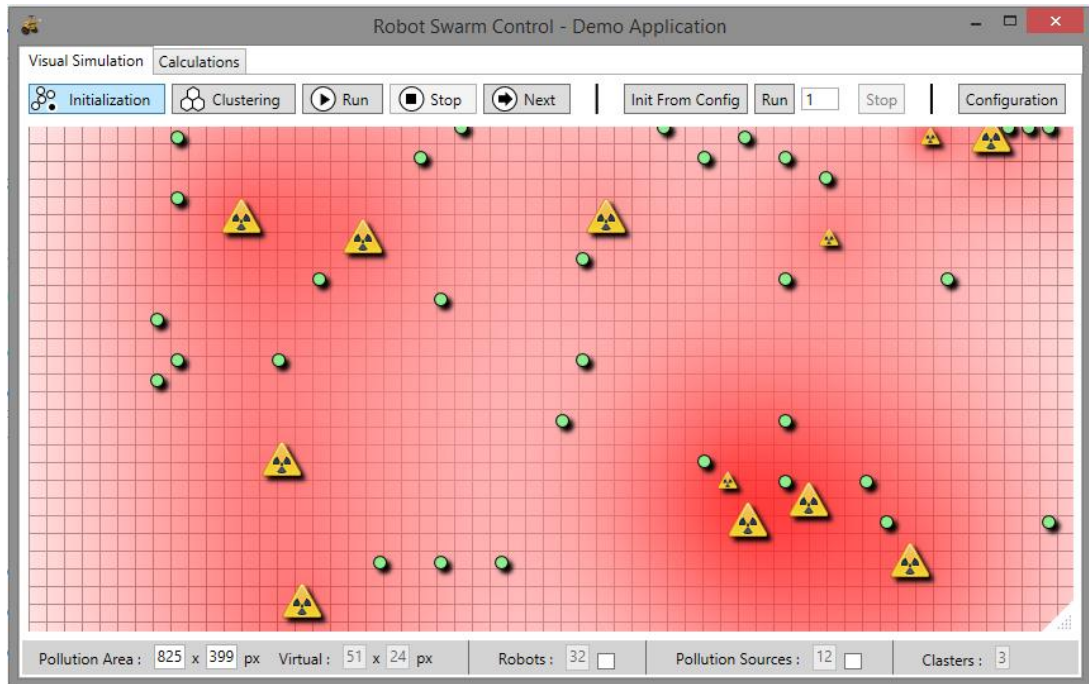
სადაც, p_{min} და p_{max} წარმოადგენს საძიებო არის შესაბამის მინიმალურ და მაქსიმალურ ლიმიტებს.

შემდეგ გამოითვლება ფიტნეს ფუნქციის მნიშვნელობა თითოეული ნაწილაკისთვის და ინახება ნაწილაკის მუხსიერებაში. ნაწილაკის საუკეთესო პოზიცია ინიციალიზაციის ეტაპზე იქნება მიმდინარე პოზიცია p_0^i . და


ბოლოს, განისაზღვრება ყოველი ნაწილაკის საწყისი სიჩქარე, რომლითაც უნდა დაიწყოს მოძრაობა:

$$v_0^i = \frac{p_{min} + rand(p_{max} - p_{min})}{\Delta t} \quad (40)$$


ქვემოთ თანდართულ სურათი 58 - ზე წარმოდგენილია ნაწილაკების გროვის ინიციალიზაციის პროცესის გრაფიკული გამოსახულება.



სურათი 58. შემთხვევითად განაწილებული ნაწილაკები რადიაციის წყაროებთან

 - რობოტი;

 - რადიაციის წყარო;

 - წითელი ფონი სურათზე აღნიშნავს რადიაციის გავრცელების არეალს.

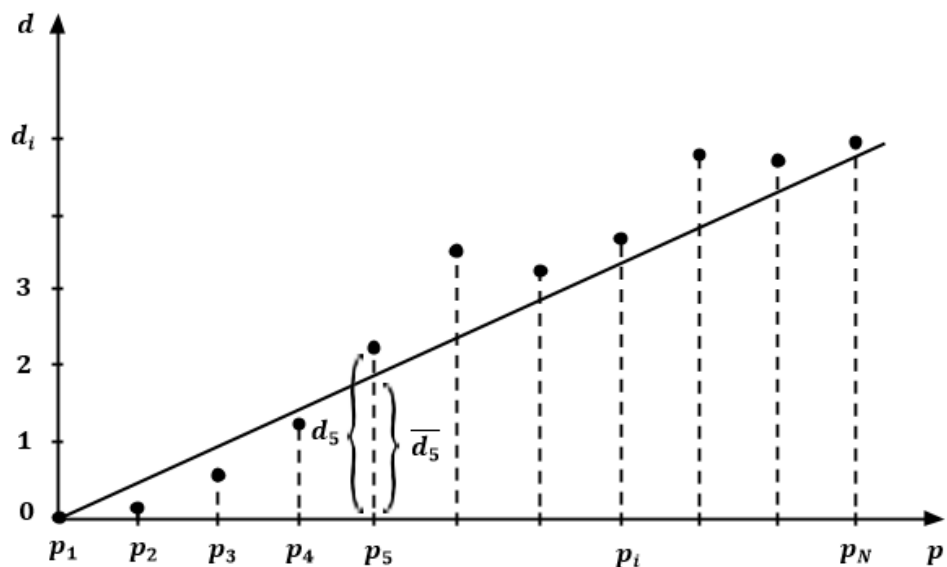
5.1.2 კლასტერიზაცია

როგორც ზემოთ აღვნიშნეთ, მიზნობრივი ფუნქციის ლანდშაფტის ზონებად დაყოფისა და ლოკალური ექსტრემუმების პოვნის ამოცანის გადაწყვეტა ნიშნავს კლასტერული ანალიზის ჩატარებას, რაც ასევე შეიძლება

იქნას გამოყენებული პროგნოზირებისთვის და სწორი გადაწყვეტილების მიღებისათვის. ეს კი საჭიროებს მრავალი სხვადასხვა სახის მონაცემების სწრაფი და ავტომატიზებული კლასტერიზაციის მექანიზმს. კლასიკური კლასტერიზაციის მეთოდებით კი ამ დონის ოპტიმიზაცია შეუძლებელია. ბიო-ინსპირირებული ოჯახის ალგორითმებმა, კერძოდ კოლექტიურ ინტელექტზე დაფუძნებულმა მეთოდებმა, უკეთესი შედეგი აჩვენეს ბევრ კლასიკურ მეთოდებთან შედარებით.

კლასტერიზაციისთვის აუცილებელია თითოეული ნაწილაკისთვის საწყის კოორდინატებში ფიტნეს ფუნქციის გამოთვლა. ფიტნეს ფუნქციის საფუძველზე, ანუ საუკეთესო პოზიციებზე, ირჩევა M რაოდენობის ლიდერი ნაწილაკები, დანარჩენი ნაწილაკები კი ავტომატურად ხდებიან აუთსაიდერები (სურათი 59).

$$l_j = \{p_k^i\}, j = 1, 2, \dots, M. \quad (41)$$



სურათი 59. ლიდერების არჩევა

ლიდერების არჩევა შემდეგნაირად ხდება: იგება გრაფიკი, სადაც აბსცისთა ღერძზე განლაგდება ნაწილაკები (p) დალაგებული რადიაციის დონის კლებადობის მიხედვით, ხოლო ორდინატთა ღერძზე გადაიზომება რადიაციის დონეებს შორის დისტანციები (d). ვიმახსოვრებთ თითოეულ კოორდინატს (p, d), რომელიც შეესაბამება p ნაწილაკს d რადიაციის დონის

სხვაობით. რა თქმა უნდა, კოორდინატთა სათავეში მოხვდება p_1 ნაწილაკი რადიაციის მაქსიმალური დონით.

$$r_1 = r_{max}; \quad r_i = f(p_i), \quad i = 1, 2, \dots, N. \quad (42)$$

$$d_i = r_{max} - r_i \quad (43)$$

$$\bar{d} = \frac{\sum_{i=1}^N d_i}{N}; \quad \bar{d}_i = \frac{\sum d_i}{i} \quad (44)$$

ამის შემდეგ, კოორდინატთა სათავედან აიგება წრფე, შემდეგი ფორმულით:

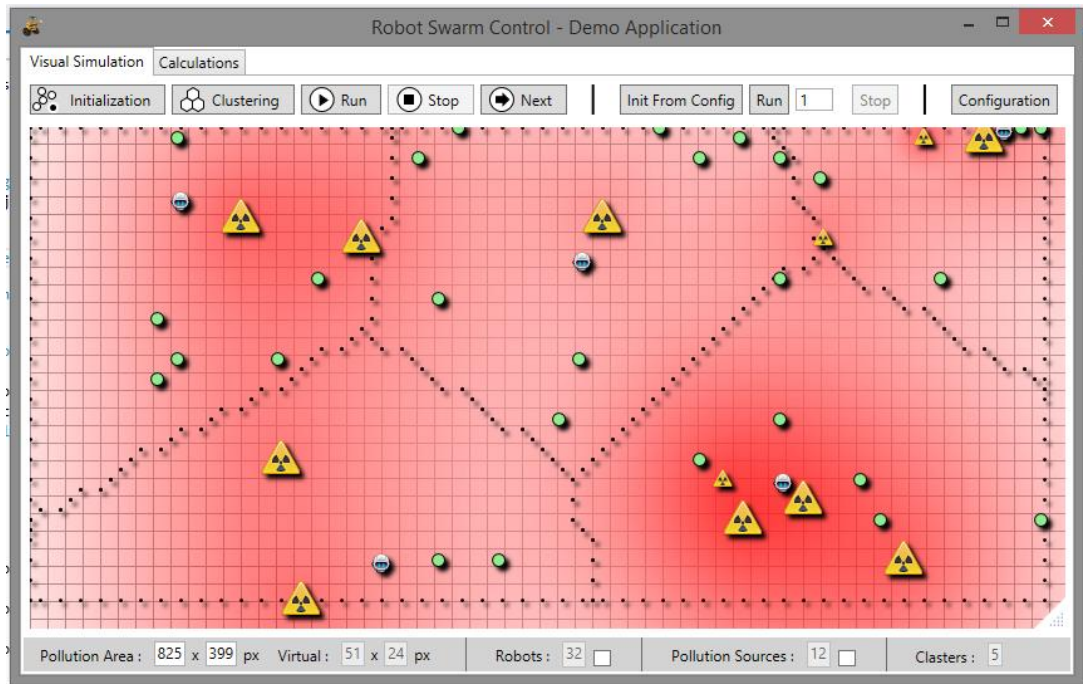
$$\tan \alpha = \frac{\sum_{i=1}^N r_i}{N(r_{max} - r_N)} \quad (45)$$

წერტილები, რომლებიც მოხვდებიან აღნიშნული წრფის ქვემოთ, გახდებიან ლიდერები, ხოლო ზემოთ - აუტსაიდერები:





$$p_i \equiv \begin{cases} d_i \leq \bar{d}_i & - \text{leader} \\ \text{otherwise} & - \text{outsider} \end{cases} \quad (46)$$

მას შემდეგ, რაც მოხდება ლიდერების არჩევა, გადავდივართ კლასტერების ფორმირებაზე. კლასტერიზაცია ხდება K-Means ალგორითმის გამოყენებით. კერძოდ, კლასტერის ცენტროიდებად ვირჩევთ ლიდერ ნაწილაკებს და ამ ცენტროიდებიდან აუტსაიდერ ნაწილაკებამდე მანძილების ანალიზით ხდება კლასტერებში აუტსაიდერების განაწილება (სურათი 60). K-Means ალგორითმისგან განსხვავებით, ჩვენს მიდგომაში კლასტერის ცენტრების დინამიური ცვლილება არ ხდება. თითოეულ კლასტერში ნაწილაკებს შორის ურთიერთობის ფორმა „star“ - ტოპოლოგიით გვაქვს განსაზღვრული. ანუ, თითოეულ აუტსაიდერ ნაწილაკს კავშირი აქვს მხოლოდ ლიდერ ნაწილაკთან.

სურათი 60 - ზე გრაფიკულად გამოსახულია ლიდერი რობოტების არჩევისა და კლასტერების ფორმირების მაგალითი. სურათზე პუნქტირით შემოსაზღვრული რეგიონები წარმოადგენენ კლასტერებს.



სურათი 60. კლასტერიზაცია

-  - ლიდერი რობოტი;
-  - აუტსაიდერი რობოტი;
-  - რადიაციის წყარო;
-  - წითელი ფონი სურათზე აღნიშნავს რადიაციის გავრცელების არეალს.

ალგორითმი - K-Means საშუალებას იძლევა გადავანაწილოთ N აუტსაიდერი M ლიდერების სიმრავლეზე $L = \{l_r\}$, $r = 1, 2, \dots, M$. იგი ცდილობს მინიმუმამდე დაიყვანოს კლასტერის წერტილების საერთო კვადრატული გადახრა კლასტერის ცენტრიდან, რომელიც ჩვენს შემთხვევაში, როგორც ზემოთ იყო აღნიშნული, ლიდერ ნაწილავს შეესაბამება:

$$\operatorname{argmin}_L = \sum_{l_c=1}^M \sum_{p_k^i \in L} \|p_k^i - p_k^j\|^2 \quad (47)$$

5.1.3 განახლება

განახლება იტერაციული პროცესია და მოიცავს ნაწილაკის სიჩქარისა და პოზიციის, ასევე, ნაწილაკისა და გროვის მეხსიერების განახლებას. მნიშვნელოვანია ის ფაქტი, რომ თითოეული იტერაცია ხდება კლასტერის ფარგლებში და დამოკიდებულია მხოლოდ ამ კლასტერის თვისებებსა და მის პოპულაციაზე. სიჩქარის განახლება, თავის მხრივ დამოკიდებულია ნაწილაკის წინა იტერაციის მოძრაობის თვისებებზე (სიჩქარე, აჩქარება, პოზიცია), ნაწილაკის საუკეთესო პოზიციასა და ამ ნაწილაკზე გროვის გავლენაზე. იგი აღიწერება ფორმულით:

$$v_{k+1}^i = wv_k^i + c_1 \text{rand} \frac{(p_k^l - p_k^i)}{\Delta t} + c_2 \text{rand} \frac{(p_k^g - p_k^i)}{\Delta t} \quad (48)$$

სადაც:

w - არის ინერცია;

p_k^l - ლოკალური (ნაწილაკის) საუკეთესო პოზიცია;

p_k^g - გლობალური (გროვის) საუკეთესო პოზიცია;

wv_k^i - მოძრაობა მიმდინარე იტერაციის დროს;

$\frac{(p_k^l - p_k^i)}{\Delta t}$ - ნაწილაკის გავლენის ფაქტორი;

$\frac{(p_k^g - p_k^i)}{\Delta t}$ - გროვის გავლენის ფაქტორი;

c_1, c_2 - შემეცნებითი და სოციალური პარამეტრები;

როგორც წესი, c_1, c_2 - არის კონსტანტები, რომლებიც ირჩევა ინიციალიზაციის ეტაპზე, მაგრამ ჩვენს მიდგომაში ეს პარამეტრები ყოველ იტერაციაზე ზუსტდება, რაც უფრო ეფექტურ შედეგს იძლევა სტანდარტულ მეთოდთან შედარებით:

$$c_1 = p_k^l / p_k^l, \quad c_2 = 1 - c_1. \quad (49)$$

თუ, ნაწილაკის წინა იტერაციის პოზიცია არ იყო ვარგისი, მაშინ მიმდინარე იტერაციაზე კანულებთ და მხედველობაში აღარ ვიღებთ წინა იტერაციის სიჩქარეს. შესაბამისად, ასეთ შემთხვევაში wv_k^i - პარამეტრი ფორმულიდან ამოვარდება.

ნაწილაკის პოზიციის განახლება კი ხდება უკვე გამოთვლილი სიჩქარის ვექტორის დახმარებით:

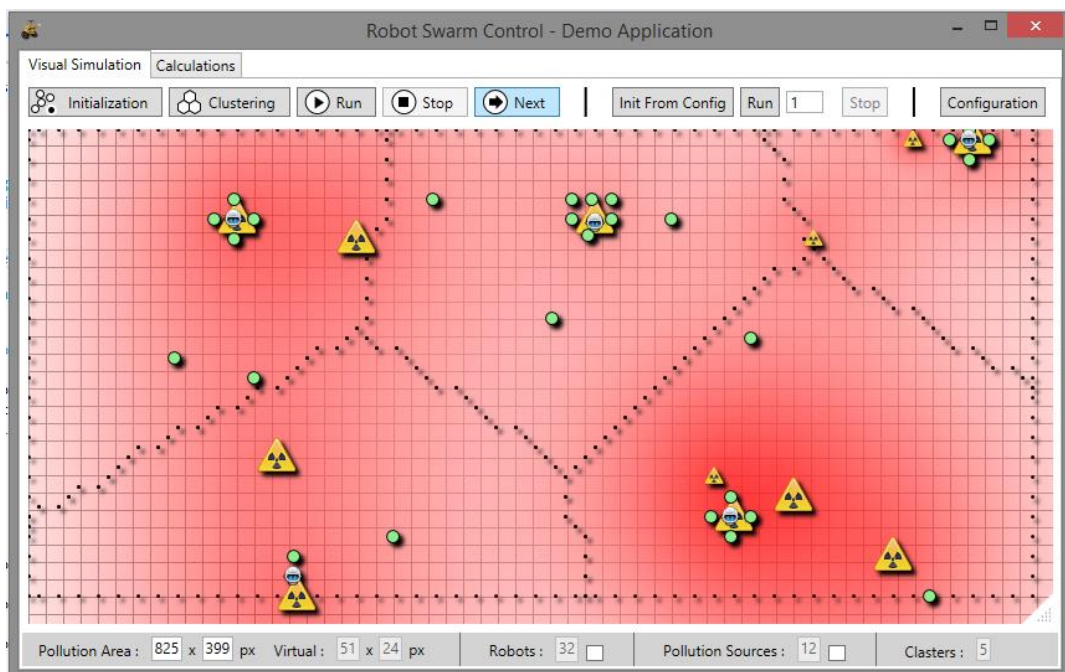
$$p_{k+1}^i = p_k^i + v_{k+1}^i \Delta t \quad (50)$$

თითოეულ იტერაციაზე, ნაწილაკების სიჩქარისა და პოზიციის განახლების შემდეგ, ხდება ნაწილაკების და გროვის საუკეთესო პოზიციების განახლება. გროვის საუკეთესო პოზიცია განისაზღვრება ლიდერების საუკეთესო პოზიციებით და დგინდება შემდეგნაირად:

$$l_{k+1}^r = p_{k+1}^g = [x_{k+1}^g, y_{k+1}^g], \quad r = 1, 2, \dots, M. \quad (51)$$

$$p_{k+1}^g \equiv \begin{cases} p_{k+1}^i & \text{if } f(p_{k+1}^i) \leq f(p_k^g), \\ p_k^g & \text{otherwise} \end{cases} \quad (52)$$

სურათი 61 - ზე გრაფიკულად გამოსახულია რობოტების განლაგება და გროვის მდგომარეობა k -ურ ბიჯზე.



სურათი 61. გროვის მდგომარეობა k -ურ ბიჯზე

5.1.4 კლასტერების გაერთიანება

კლასტერების გაერთიანება არ არის აუცილებელი პროცესი და ხდება მხოლოდ იმ შემთხვევაში, როცა ლიდერი ნაწილაკი იმყოფება კლასტერის

საზღვარზე და გადაადგილების შემდეგი ბიჯის ლოკაცია ეკუთვნის მეორე კლასტერს. კლასტერების გაერთიანების შემდეგ ახალ კლასტერში ხდება ახალი ლიდერის არჩევა და ყველა ნაწილაკისათვის - „განახლება“ ბიჯის გამეორება (იხ. 5.1.3).

5.1.5 ალგორითმის შეწყვეტა

ალგორითმის შემოწმება შეწყვეტის კრიტერიუმებზე ხორციელდება რამდენიმე ეტაპით:

I ეტაპზე დგინდება მოძრაობისას რამდენად აუმჯობესებს ლიდერი საუკეთესო პოზიციას:

$$\partial_{k+1} = |f(p_{k+1}^g) - f(p_k^g)| \leq \mu \quad (53)$$

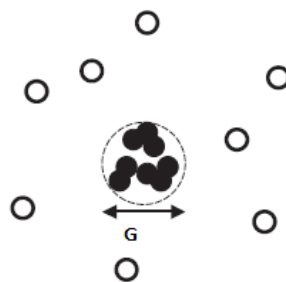
სადაც, μ - არის წინასწარ განსაზღვრული ბიჯი.

II ეტაპზე დგინდება ლიდერი რობოტის მიმართ კლასტერში ნაწილაკების შეჯგუფების ხარისხი:

$$D = \frac{1}{N} \sum_{l=1}^N \sqrt{|p_k^i - p_k^j|^2} \leq G \quad (54)$$

სადაც, p_{k+1}^c - არის შეჯგუფების ცენტრალური წერტილი (სურათი 62).

$$p_{k+1}^c = \frac{1}{Q} \sum_{l=1}^Q \sqrt{|p_{k+1}^c - p_{k+1}^i|^2}, \quad i \neq j. \quad (55)$$

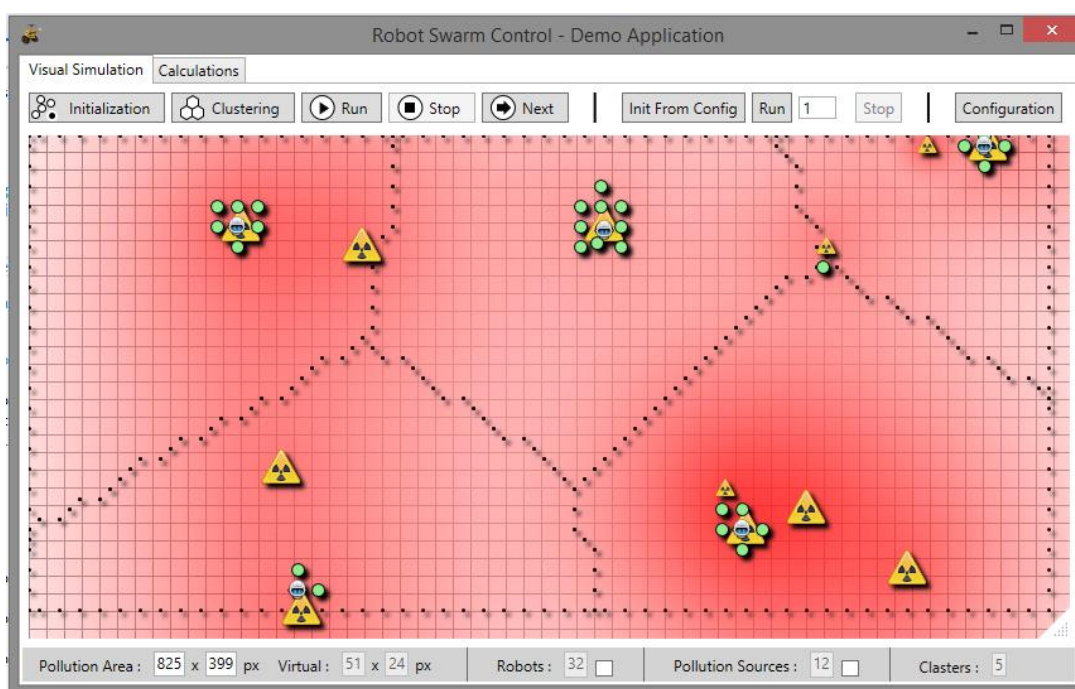


სურათი 62. შეჯგუფების ცენტრალური წერტილი

III ბოლო ეტაპზე დგინდება აღნიშნული კომპონენტების ფუნქციის მნიშვნელობა:

$$S = \partial_{k+1} + D \Rightarrow \min. \quad (56)$$

იმ შემთხვევაში, თუ, კმაყოფილდება ზემოთ აღწერილი კრიტერიუმი, მაშინ ალგორითმი წყვეტს მუშაობას და გამოგვაქვს შედეგი (სურათი 63), წინააღმდეგ შემთხვევაში ალგორითმი იწყებს ახალ იტერაციას „განახლება“ ბიჯზე გადასვლით (იხ. 5.1.3).



სურათი 63. საბოლოო მდგომარეობა

5.2 ახლებური მიდგომის უპირატესობები

კლასიკური PSO-თან შედარებით ჩვენს ალგორითმს აქვს რამდენიმე უპირატესობა:

დინამიური პარამეტრები - კლასიკურ მიდგომაში გროვის პარამეტრები ინიციალიზაციის შემდეგ არ იცვლება, ამიტომ ძალზე მნიშვნელოვანია ყველა პარამეტრის საფუძვლიანად შერჩევა. ჩვენს მოდელში კი, პარამეტრების დიდი ნაწილი დინამიურად გამოითვლება იტერაციების და გროვის მდგომარეობის მიხედვით.

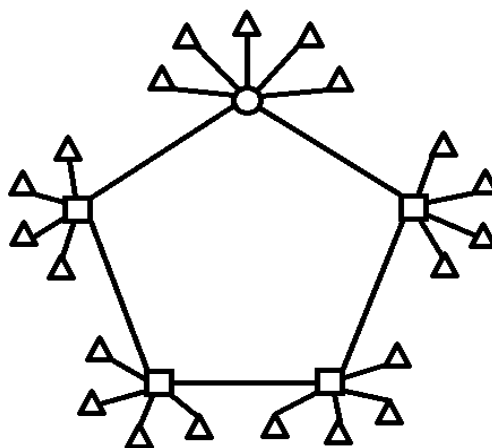
გროვის ზომა - რობოტების ოპტიმალური რაოდენობის დასადგენად ჩვენ მოვახდინეთ კონფლიქტური კრიტერიუმების (რაოდენობრივი

შეზღუდვა, ენერგო შეზღუდვა და დრო) ოპტიმიზაცია და აღნიშნული შეზღუდვების გათვალისწინებით შევიმუშავეთ რაოდენობის დინამიურად დადგენის ფორმულა.

კლასტერიზაცია/გაერთიანება - ალგორითმში შემოთავაზებულია კლასტერიზაციის ჩვენებური მიდგომა, კერძოდ - კლასტერის ცენტროიდების არჩევა. ამისათვის შემოტანილია ლიდერი და აუთსაიდერი რობოტების ცნებები და აღწერილია ლიდერების შერჩევის კრიტერიუმები.

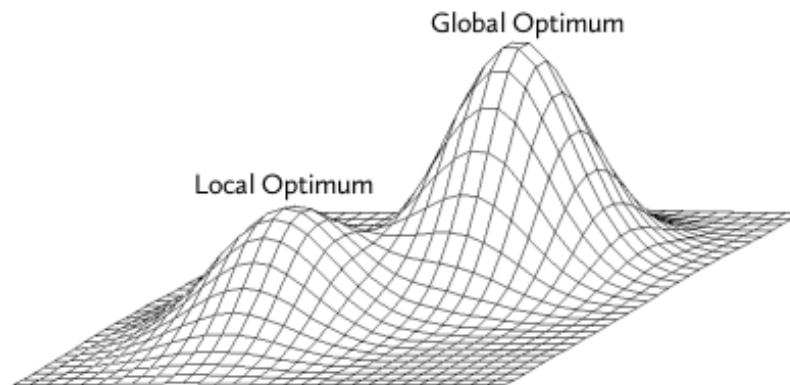
ალგორითმში კლასტერიზაცია, ისევე როგორც კლასტერების გაერთიანება, ხდება საძიებო არის ნაწილაკების სტატუსის მიხედვით. შესაბამისად, ძიება ხდება არა მთელ არეალზე, არამედ თითოეული კლასტერის შიგნით, რაც გვაძლევს ოპტიუმების სწრაფად პოვნის საშუალებას, ალგორითმის შესრულების პროცესს ხდის მარტივად დაპარალელებადს და მოიხმარს ნაკლებ მანქანურ რესურს განაწილებულ სისტემებში.

ტოპოლოგია - ალგორითმში გამოყენებულია კლასიკური PSO-ს ტოპოლოგიებიდან ორის ნაზავი. კერძოდ, კლასტერებში რობოტებს შორის კავშირი განისაზღვრება „Star“ ტოპოლოგიით, ხოლო თავის მხრივ, კლასტერები ქმნიან „lBest“ ტოპოლოგიას, რაც აადვილებს კომუნიკაციას და ასევე, ზოგავს მეხსიერების და მონაცემთა გაცვლის ქსელურ რესურსებს (სურათი 64).



სურათი 64. ალგორითმში გამოყენებული ტოპოლოგიის სქემა

მრავალმიზნობრივი ოპტიმიზაცია - ალგორითმში ხდება რამდენიმე კონფლიქტური კრიტერიუმების ოპტიმიზაცია, ასევე კლასიკური ნაწილაკების გროვის ოპტიმიზაციისგან (PSO) განსხვავებით, გლობალური ოპტიმუმის გარდა ალგორითმი პოულობს ლოკალურ ოპტიმუმებსაც (იხ. სურათი 65).



სურათი 65. სამიუბო არის ლოკალური და გლობალური ოპტიმუმები

5.3 ექსპერიმენტის დაგეგმვა და რეალიზაცია

PSO-ს პარადიგმების მიხედვით, სამიუბო არეში მოძრავი გროვის ნებისმიერი ნაწილაკი წარმოადგენს ამოცანის პოტენციურ ამონახსნს. ალგორითმის მიზანია, ნაწილაკების შეგროვება მიზნობრივი ფუნქციის გლობალურ (მთელი სამიუბო არე) ან ლოკალურ (კონკრეტული კლასტერი) ოპტიმუმებში.

ექსპერიმენტის დროს უნდა განისაზღვროს საწყისი პარამეტრები, რომლებიც დაყვანილი სახით არის წარმოდგენილი: გროვის ზომა N , წონის პარამეტრები c_1 , c_2 , სამიუბო არის საზღვრები x_{min} , x_{max} , y_{min} , y_{max} და ალგორითმის შეწყვეტის კრიტერიუმები G , μ .

ექსპერიმენტის სიმულაციისთვის ჩვენ შევქმენით პროგრამა, რომელიც ახდენს რობოტების მოძრაობის სიმულაციას ზემოთ აღწერილი ალგორითმის საფუძველზე. პროგრამა ვიზუალური ეფექტების გარდა აგროვებს სტატისტიკურ მონაცემებს, რისი ანალიზის საფუძველზეც

დადგინდება ჩვენი მიდგომის ეფექტურობა/არაეფექტურობა. თვალსაჩინოებისთვის, პროგრამაში შემოვიღეთ პირობითი ზომები და დავადგინეთ მათი მაქსიმალური და მინიმალური მნიშვნელობები. მაგალითად, რადიაციის გამოსხივების საზღვრებად ავიღეთ $[0, 100]$ შუალედი, სამუშაო არეალის აღსაწერად იგი დავყეთ პირობით წერტილებად (პიქსელებად), ხოლო ფიტნეს ფუნქციის შესაფასებლად გამოვიყენეთ პროცენტი (%).

ექსპერიმენტის დასაწყებად საჭიროა პროგრამაში საწყისი მონაცემების შეტანა, ანუ ინიციალიზაცია. პარამეტრების ნაწილი განისაზღვრება სტატიკურად და პროგრამაში შეიტანება ხელით, დანარჩენი კი, პროგრამის მიერ დინამიურად გამოითვლება შესაბამისი ალგორითმებით.

სტატიკურად შევიყვანეთ არეალის ზომა - 70×45 პიქსელი და რადიაციის 13 წყარო გამოსხივების სხვადასხვა სიმძლავრით, რომელთა კოორდინატებიც მოცემულია

ცხრილი 1 - ში.

ცხრილი 1. რადიაციის წყაროები

რადიაციის წყარო №	X კოორდინატი	Y კოორდინატი	რადიაციის ზომა
1	2	7	95
2	5	22	33
3	4	39	83
4	44	1	14
5	24	6	72
6	61	16	78
7	6	39	27
8	52	16	61
9	56	15	12
10	5	29	22
11	39	30	56
12	6	3	32

13	61	7	18
----	----	---	----

აღნიშნული მონაცემების განსაზღვრის შემდეგ პროგრამა ახდენს რადიაციის გავრცელების სიმულაციას. რადიაციის გავრცელება დამოკიდებულია მრავალ ფაქტორზე, მაგალითად, წყაროდან გამოსხივებული ნაწილაკების ბუნებაზე, ხელისშემშლელ და ხელშემწყობ პირობებზე (მაგ: წინაღობა ან ხელშემწყობი ქარი) და ა.შ. ამიტომ, ჩვენ გამოვიყენეთ ენერგიის გავრცელების ზოგადი წესები იდეალური გარემოსთვის:

$$P_i = \sum_{k=1}^n \frac{P_k^t}{R^2} \quad (57)$$

სადაც, P_i - არის ენერგიის (Power Density) ზომა i -ურ წერტილში, $k = 1, 2, \dots, n$ - ენერგიის წყაროების რაოდენობა, P_k^t - k - ური ენერგიის წყაროს ინტენსივობა (სიმძლავრე, Total Power Density), R - მანძილი P_k გამოსხივების წყაროდან P_i - წერტილამდე.

შემდეგ ხდება რობოტების ოპტიმალური რაოდენობის დადგენა, ფორმულით:

$$\text{Robot Count} = \frac{a*b}{\sqrt{\left(\frac{a}{2}\right)^2 + \left(\frac{b}{2}\right)^2}} \quad (58)$$

სადაც a და b საძიებო არის ზომები.

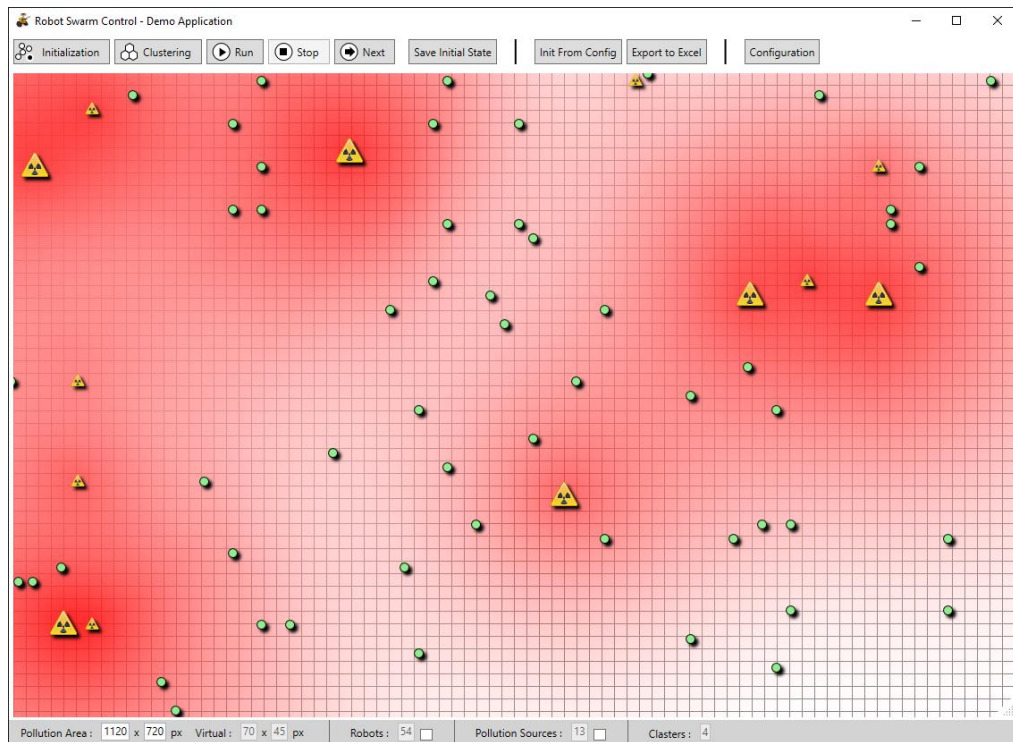
ოპტიმალურობის განსაზღვრა ამ შემთხვევაში მოიცავს ქვემოთ ჩამოთვლილ შეზღუდვებს შორის გარკვეული ბალანსის დამყარებას:

- რაოდენობრივი შეზღუდვა - ასეთი ტექნიკით აღჭურვილი რობოტების წარმოება ძვირი ჯდება. ამიტომ საჭიროა რობოტების ოპტიმალური რაოდენობის დადგენა საძიებო არის გამოსაკვლევად.
- ენერგო შეზღუდვა - არსებული ენერგიით რობოტმა უნდა შეძლოს მისი შესრულება და უკან დაბრუნება. შესაბამისად, რობოტების

რაოდენობა ისე უნდა შეირჩეს, რომ მოხდეს მთელი სამიეზო არის დაფარვა მოცემული შეზღუდვის გათვალისწინებით.

- დროის ფაქტორი - არსებული ამოცანა უნდა შესრულდეს მისაღებ დროში. სამიეზო არის კვლევის დრო რობოტების რაოდენობის უკუპროპორციულია.

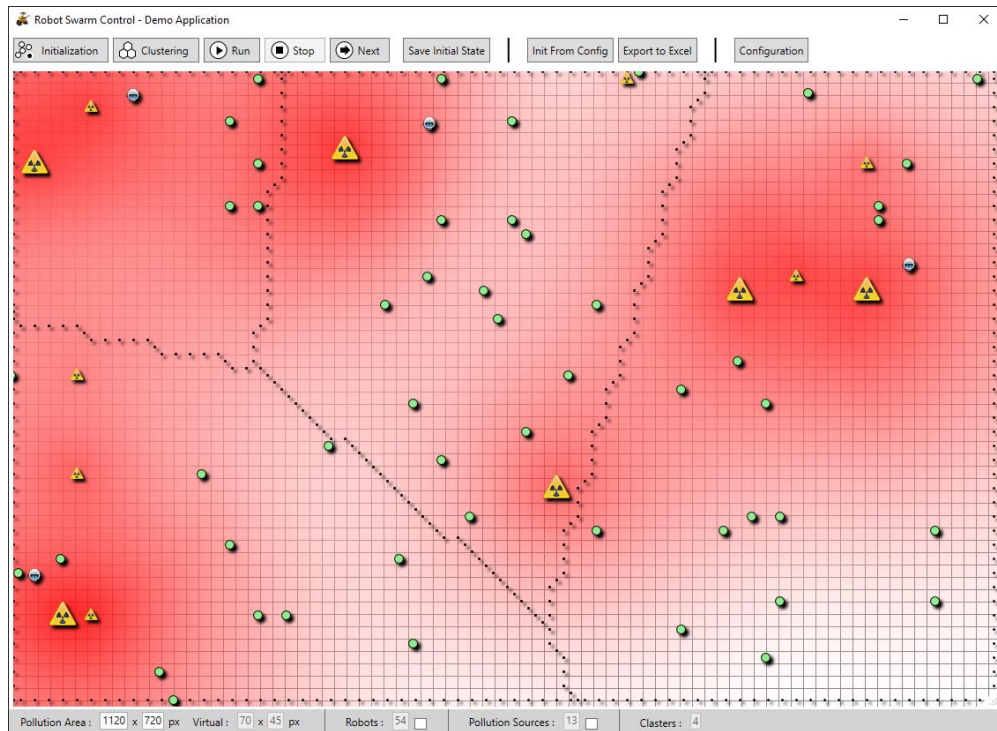
ჩვენს სადემონსტრაციო მაგალითში ალგორითმის მიერ რობოტების ოპტიმალურ რაოდენობად განისაზღვრა 54 ერთეული (ცალი). და ბოლოს, რობოტები სამიეზო არეში უნდა დავსვათ შემთხვევითად დაგენერირებულ პოზიციებზე. ამით ინიციალიზაციის პროცესი მთავრდება (სურათი 66).



სურათი 66. ინიციალიზაცია

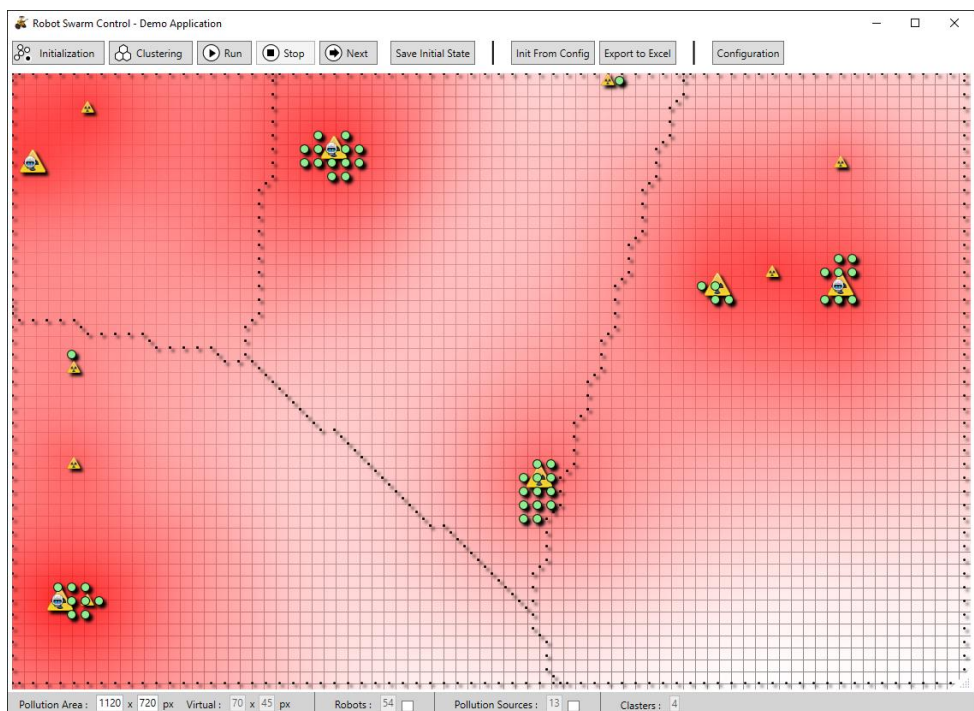
ინიციალიზაციას მოსდევს კლასტერიზაცია. კლასტერიზაციისთვის აუცილებელია თითოეული ნაწილაკისთვის საწყის კოორდინატებში ფიტნეს ფუნქციის გამოთვლა. ფიტნეს ფუნქციის საფუძველზე, ანუ საუკეთესო პოზიციებზე, ირჩევა M რაოდენობის ლიდერი ნაწილაკი, დანარჩენი ნაწილაკები კი, ავტომატურად ხდებიან აუტსაიდერები. კლასტერში

შემავალი ნაწილაკები იზოლირებულნი არიან კლასტერის დონეზე და არ აქვთ კავშირი სხვა კლასტერის ნაწილაკებთან (სურათი 67).



სურათი 67. კლასტერიზაცია

კლასტერიზაციის შემდეგ იწყება ალგორითმის იტერაციული ნაწილი, ე.ი. რობოტები იწყებენ მოძრაობას რადიაციის წყაროებისკენ. მოძრაობა განისაზღვრება პოზიციის ცვლილებით მიმდინარე კოორდინატებიდან სწორად განსაზღვრული მანძილებით. ძიების პროცესში, თუ აღმოჩნდა ისე, რომ რომელიმე აუტსაიდერი ნაწილაკის პოზიცია უკეთესია ვიდრე ლიდერი ნაწილაკის, მაშინ აუტსაიდერი ნაწილაკი ავტომატურად გახდება ლიდერი და პირიქით. დასაშვებია კლასტერების გაერთიანებაც ალგორითმის შესრულების რომელიმე იტერაციაზე. რობოტების მოძრაობა გრძელდება მანამ, სანამ არ დაკმაყოფილდება ზემოთ აღწერილი შეჩერების კრიტერიუმებიდან ერთ-ერთი მაინც. სურათი 68 - ზე გამოსახულია ალგორითმის დასრულების შემდეგ, კლასტერების შიგნით რობოტების შეჯგუფება მათ მიერ აღმოჩენილი რადიაციის წყაროებთან.



სურათი 68. ალგორითმის დასრულება

ჩვენი ალგორითმის მიხედვით ჩავატარეთ 20 ცდა. ყველა ცდა ჩავატარეთ ერთი და იგივე საწყის მონაცემებზე, გარდა რობოტების საწყისი პოზიციებისა, რადგან როგორც ზემოთ აღვნიშნეთ, ისინი შემთხვევითად გენერირდება ყოველი ცდის ინიციალიზაციის დროს. თანდართულ ცხრილებში (

ცხრილი 2) წარმოდგენილია თითოეულ ჩატარებულ ცდაზე შეგროვებული მონაცემები რომელთა ანალიზიც წარმოდგენილია მომდევნო ქვეთავში.

ცხრილი 2. 1-20 ცდის დეტალური შედეგები

ცდა №	საწყ. კლასტ. რაოდ.	კლასტ. №	ლიდ. №	ლიდ. ბიჯების რაოდ.	გროვის ბიჯების რაოდ.	ლიდ. ფიტ. ფუნქციის სიზუსტე (%)	გროვის ფიტ. ფუნქციის სიზუსტე (%)	ლოკ. საუკეთესო პოზიცია (x, y)	გლობ. საუკეთესო პოზიცია (x, y)
1	5	1	14	5	26	99.4	98.3	(4, 39)	(2, 7)
		2	49	4		99.44		(61, 16)	
		3	10	2		98.97		(24, 6)	
		4	18	4		99.6		(39, 30)	

		5	42	10		99.32		(2, 7)	
2	4	1	54	6	24	99.32	98.34	(2, 7)	(2, 7)
		2	27	5		99.4		(4, 39)	
		3	1	4		99.44		(61, 16)	
		4	11	4		99.48		(24, 6)	
3	3	1	10	6	20	99.32	98.58	(2, 7)	(2, 7)
		2	7	4		99.44		(61, 16)	
4	4	1	23	3	25	98.64	98.29	(2, 7)	(2, 7)
		2	5	4		99.4		(4, 39)	
		3	24	5		99.44		(61, 16)	
		4	52	4		99.48		(24, 6)	
5	4	1	50	2	19	98.64	98.26	(2, 7)	(2, 7)
		2	43	3		98.81		(4, 39)	
		3	13	4		99.44		(61, 16)	
		4	2	2		98.97		(24, 6)	
6	3	1	47	4	28	99.4	98.6	(4, 39)	(2, 7)
		2	6	6		99.32		(2, 7)	
		3	4	5		99.44		(61, 16)	
7	2	1	22	5	25	99.32	98.1	(2, 7)	(2, 7)
		2	44	3		99.44		(61, 16)	
8	4	1	6	2	26	99.32	98.73	(2, 7)	(2, 7)
		2	20	5		99.44		(61, 16)	
		3	45	5		98.81		(4, 39)	
		4	52	4		99.48		(24, 6)	
9	3	1	28	5	29	99.32	98.43	(2, 7)	(2, 7)
		2	16	3		99.4		(4, 39)	
		3	51	9		99.44		(61, 16)	
10	3	1	14	2	21	98.81	98.61	(4, 39)	(2, 7)
		2	30	2		98.88		(61, 16)	
		3	50	7		99.32		(2, 7)	
11	4	1	5	3	23	99.32	98.55	(2, 7)	(2, 7)
		2	43	2		98.81		(4, 39)	
		3	27	4		99.44		(61, 16)	
		4	45	5		99.48		(24, 6)	
12	3	1	16	3	21	98.81	98.35	(4, 39)	(2, 7)
		2	18	6		99.32		(2, 7)	
		3	19	5		99.44		(61, 16)	
13	5	1	13	3	19	99.32	98.59	(2, 7)	(2, 7)
		2	20	7		99.44		(61, 16)	
		3	35	6		99.48		(24, 6)	

		4	26	6		98.81		(4, 39)	
14	3	1	37	5	24	99.32	98	(2, 7)	(2, 7)
		2	14	5		99.4		(4, 39)	
		3	4	7		99.44		(61, 16)	
15	4	1	51	3	27	99.4	98.55	(4, 39)	(2, 7)
		2	9	2		99.44		(61, 16)	
		3	36	7		99.32		(2, 7)	
		4	10	4		98.97		(24, 6)	
16	3	1	17	3	23	98.64	98.6	(2, 7)	(2, 7)
		2	32	4		99.44		(61, 16)	
		3	20	7		99.4		(4, 39)	
17	5	1	7	2	26	99.32	97.93	(2, 7)	(2, 7)
		2	50	4		98.88		(61, 16)	
		3	9	6		99.4		(4, 39)	
		4	48	4		99.48		(24, 6)	
		5	21	3		99.2		(39, 30)	
18	4	1	41	3	27	98.64	97.39	(2, 7)	(2, 7)
		2	18	5		99.44		(61, 16)	
		3	19	4		99.48		(24, 6)	
		4	54	7		99.4		(4, 39)	
19	3	1	34	4	28	99.32	98.61	(2, 7)	(2, 7)
		2	15	5		99.44		(61, 16)	
		3	19	6		99.4		(4, 39)	
20	4	1	23	3	29	99.32	98.64	(2, 7)	(2, 7)
		2	51	3		98.81		(4, 39)	
		3	27	3		99.44		(61, 16)	
		4	5	5		99.48		(24, 6)	

ჩატარებული ცდების საბოლოო შედეგების მისაღებად გამოვთვალეთ ლიდერი ნაწილაკების რამდენიმე ძირითადი პარამეტრის - ლიდერებისა და გროვის ფიტნეს ფუნქციის, ასევე, მათი ბიჯების საშუალო მნიშვნელობები. შედეგები წარმოდგენილია ცხრილი 3 - ში.

ცხრილი 3. ცდების ჯამური შედეგები

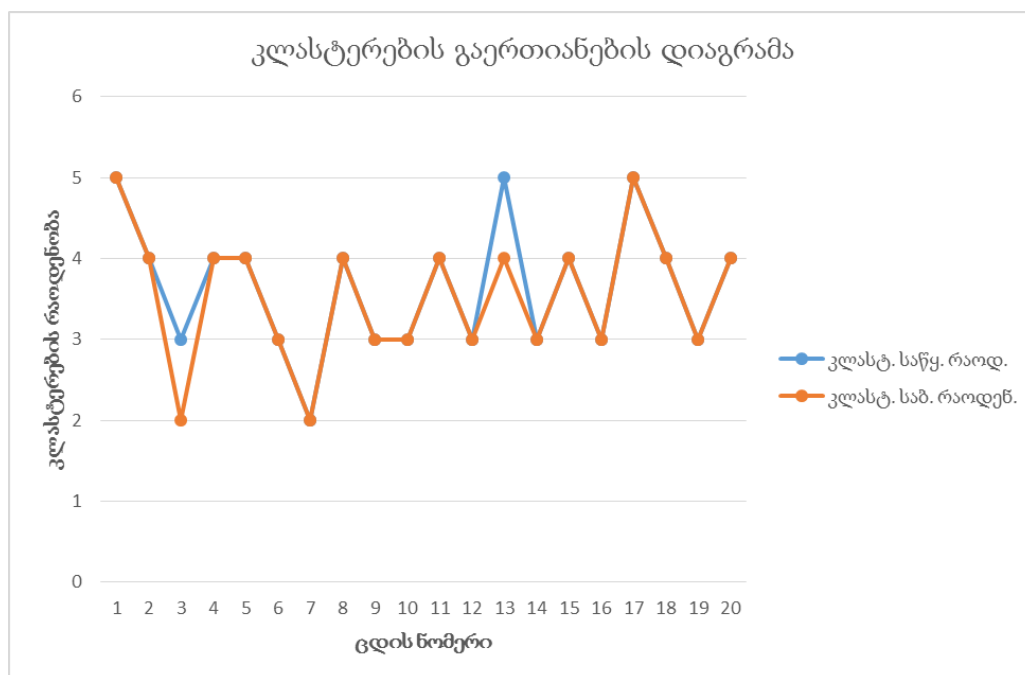
ნაწილაკების რაოდენობა	კლასტერების საშუალო რაოდენობა	ლიდერების ბიჯების საშუალო რაოდენობა	გროვის ბიჯების საშუალო რაოდენობა	ლიდერის ფიტნეს ფუნქციის საშუალო სიზუსტე	გროვის ფიტნეს ფუნქციის საშუალო სიზუსტე
--------------------------	-------------------------------------	--	---	---	--

54	3.65	4.39	24.5	99.27	98.37
----	------	------	------	-------	-------

თავი 6 შედეგები და მათი განსჯა

ანალიზის შედეგებს წარმოვადგენთ რამდენიმე ნაწილად, რომელთაც თვალსაჩინოებისთვის დავურთეთ შესაბამისი დიაგრამები.

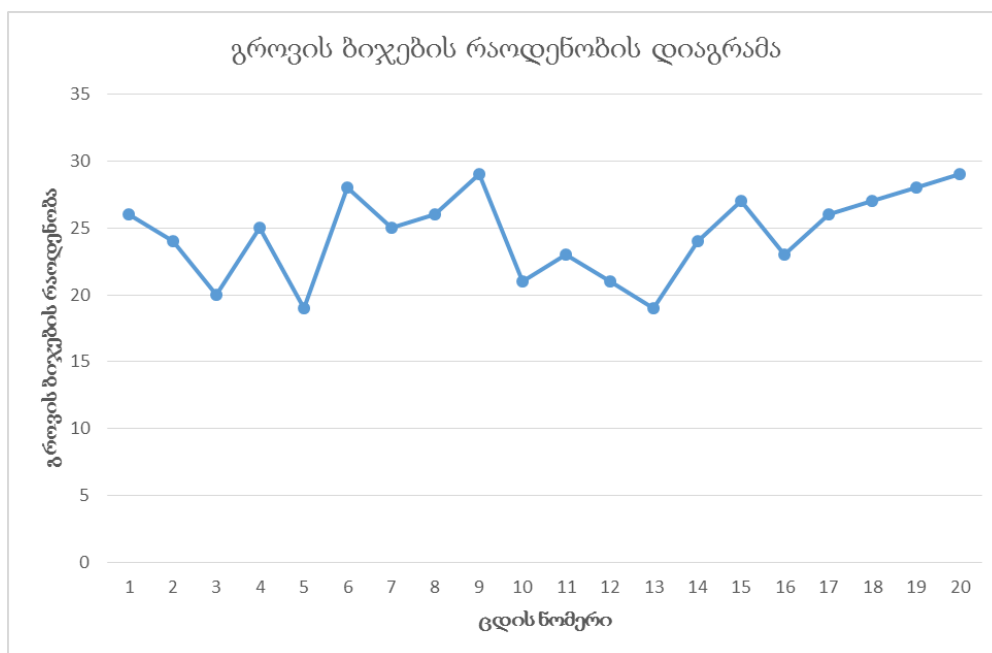
ანალიზის შედეგად დადგინდა, რომ 20 ცდიდან 2 ცდაში საჭირო გახდა კლასტერების გაერთიანება (merging), კერძოდ, მესამე ცდაზე 3 კლასტერის ნაცვლად დარჩა 2, ხოლო მეცამეტე ცდაზე - 5 - ის ნაცვლად 4 კლასტერი. როგორც ცხრილი 3 - დან ჩანს აღნიშნული არეალისთვის კლასტერების საშუალო რაოდენობა მერყეობს 3-დან 4 კლასტერამდე. სურათი 69-ზე წარმოდგენილია კლასტერების რაოდენობის და გაერთიანების დინამიკა.



სურათი 69. კლასტერების გაერთიანების დიაგრამა

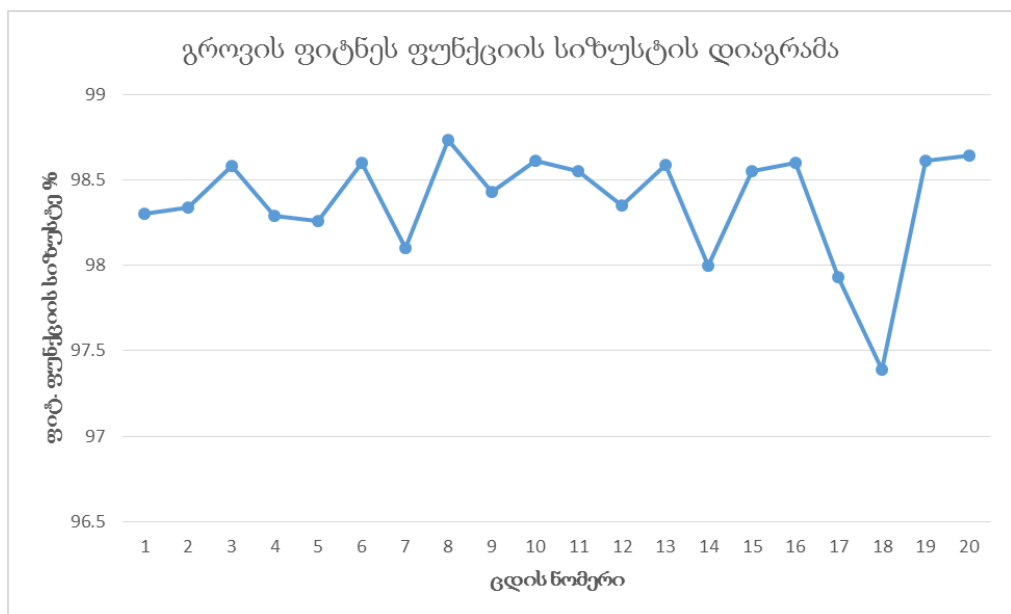
ასევე, დადგინდა, რომ როგორც ლიდერების ბიჯების რაოდენობა, ასევე გროვის ბიჯების საერთო რაოდენობაც, დამოკიდებულია რობოტების საწყის პოზიციებზე და კლასტერების რაოდენობაზე (სურათი 70). თუმცა, წინასწარ შეუძლებელია იმის დადგენა თუ რა საწყის პოზიციებზე უნდა განვათავსოთ რობოტები, რომ მინიმუმამდე დავიყვანოთ ბიჯების რაოდენობა. ჩვენს

შემთხვევაში ლიდერი რობოტების ბიჯების საშუალო რაოდენობა არის 4.39 ბიჯი, ხოლო გროვის ბიჯების საშუალო რაოდენობა კი 24.5 ბიჯი.



სურათი 70. გროვის ბიჯების რაოდენობის დიაგრამა

რაც შეეხება ფიტნეს ფუნქციის მნიშვნელობებს ლიდერი რობოტებისა და მთელი გროვისთვის, შედეგები შემდეგნაირად გამოიყურება: ლიდერი რობოტების - 99.27, გროვის - 98.37. ამდენად, ალგორითმმა საკმაოდ დიდი სიზუსტე აჩვენა ამ ასპექტშიც, როგორც ლიდერი რობოტებისათვის ასევე მთლიანად გროვისთვის (სურათი 71).



სურათი 71. გროვის ფიტნეს-ფუნქციის სიზუსტის დიაგრამა

თავი 7 დასკვნა

სადისერტაციო ნაშრომში „კოლექტიური ინტელექტის მეთოდების გამოყენება მულტი-რობოტული სისტემების მართვაში“, ჩვენ დავსვით საკმაოდ აქტუალური ამოცანა, მიმოვიხილეთ მისი გადაჭრის არსებული მეთოდები, ვაჩვენეთ პრობლემური მხარეები და წარმოვადგინეთ ჩვენი მიდგომა, რომელიც ვთვლით, რომ უფრო ოპტიმალურია და უმეტეს შემთხვევებში უკეთეს შედეგს იძლევა.

ალგორითმის პრაქტიკული გამოყენება შესაძლებლობას იძლევა, განხორციელდეს გარემოს დაბინძურების, განსაკუთრებით ატომური ენერგეტიკით, მონიტორინგი, რაც მიიღწევა მობილური, უკაბელო სენსორული ქსელიდან ინფორმაციის უწყვეტად მიღებით და ამ ინფორმაციის შეგროვება-დამუშავებით.

საკმაო ლიტერატურის გადამუშავების შედეგად მივედით დასკვნამდე, რომ ასეთი სახის ამოცანებისთვის უკეთესია გამოვიყენოთ ევოლუციური ალგორითმებით მართვადი მულტი-რობოტული სისტემა. ჩვენ განვსაზღვრეთ ამ სისტემის პარამეტრები და შევიმუშავეთ მისი მართვის სტრატეგიები.

ნაშრომში განხილულია რამდენიმე ევოლუციური ალგორითმი, რომელთაგან ამოსავალ წერტილად ავიღეთ ნაწილაკების გროვის ოპტიმიზაციის მეთოდები. ნაშრომში წარმოვადგინეთ PSO-ზე დაფუძნებული ადაპტური ალგორითმი, რომელიც ორიგინალი მიდგომისგან განსხვავებით გლობალურ ოპტიმუმთან ერთად პოულობს ლოკალურ ოპტიმუმებსაც. ალგორითმში პატარ-პატარა ამოცანების გადასაწყვეტად კი მოვიშველიეთ ეფექტური და კარგად ნაცნობი კლასტერიზაციის, მოძრაობისა და ენერგიის დაზოგვის ალგორითმები. ჩვენ გვჯერა და ჩვენს მიერ ჩატარებულმა ექსპერიმენტის შედეგებმაც ცხადყო, რომ ჩვენს მიერ შემუშავებულ ალგორითმს შეუძლია სწრაფად მოძებნოს ოპტიმალური შედეგი გარემო პირობების მუდმივად ცვლილების დროსაც კი.

ნაშრომის ექსპერიმენტულ ნაწილში მოცემულია ნაწილაკთა გროვის ოპტიმიზაციის მოდიფიცირებული ალგორითმის ექსპერიმენტის შედეგები და ანალიზი. გამოთვლების შედეგებით დადასტურდა ნაწილაკებზე და კლასტერების შიგნით ფიტნეს ფუნქციის გაუმჯობესება, რაც ავტომატურად ნიშნავს იტერაციების რაოდენობის და დროის დაზოგვას. ექსპერიმენტების შედეგებმა ცხადყო შემუშავებული ალგორითმის ეფექტურობა, მაღალი სიზუსტე და კრებადობა, კლასიკურ მეთოდებთან შედარებით, რაც იყო ჩვენი კვლევის მიზანი და წარმოადგენს დისერტაციის მიღწევას.

გამოყენებული ლიტერატურა

1. E. Shaw, "The schooling of fishes," Sci. Am., vol. 206, pp. 128-138, 1962.
2. E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems. NY: Oxford Univ. Press, 1999.
3. R. Arkin, Behavior-Based Robotics. Cambridge, MA: MIT Press, 1998.
4. G. Beni and J. Wang, "Swarm intelligence in cellular robotics systems," in Proceeding of NATO Advanced Workshop on Robots and Biological System, 1989.
5. M. Pachter and P. Chandler, "Challenges of autonomous control," IEEE Control Systems Magazine, pp. 92-97, April 1998.
6. G. Flake, The Computational Beauty of Nature. Cambridge, MA: MIT Press, 1999.
7. C. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," Comp. Graph, vol. 21, no. 4, pp. 25-34, 1987.
8. M. Resnick, Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. Cambridge, MA: MIT Press, 1994.
9. D. Terzopoulos, X. Tu, and R. Grzeszczuk, "Artificial fishes with autonomous locomotion, perception, behavior, and learning in a simulated physical world," in Artificial Life I, p. 327, MIT Press, 1994.
10. K. Sims, "Evolving 3d morphology and behavior by competition," in Artificial Life I, p. 353, MIT Press, 1994.
11. G. Dudek and et al., "A taxonomy for swarm robots," in IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, (Yokohama, Japan), July 1993.
12. S. Hackwood and S. Beni, "Self-organization of sensors for swarm intelligence," in IEEE Int. Conf. on Robotics and Automation, (Nice, France), pp. 819-829, May 1992.
13. R. Brooks, "Intelligence without reason," tech. rep., Artificial Intelligence Memo. No. 1293, 1991.

14. T. Ueyama, T. Fukuda, and F. Arai, "Configuration of communication structure for distributed intelligent robot system," in Proc. IEEE Int. Conf. on Robotics and Automation, pp. 807-812, 1992.
15. M. Mataric, "Minimizing complexity in controlling a mobile robot population," in IEEE Int. Conf. on Robotics and Automation, (Nice, France), May 1992.
16. T. Fukuda, T. Ueyama, and T. Sugiura, "Self-organization and swarm intelligence in the society of robot being," in Proceedings of the 2nd International Symposium on Measurement and Control in Robotics, 1992.
17. T. Fukuda, G. Iritani, T. Ueyama, and F. Arai, "Optimization of group behavior on cellular robotic system in dynamic environment," in Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pp. 1027-1032, 1994.
18. G. Giralt, "The Remote-Operated Autonomous-Robot concept: from deep sea to outer space applications," Intl. Symp. Missions Technologies and Design of Planetary Mobile Vehicles, Toulouse, France, 1992.
19. D.M. Wilson, S. Hoyt, J. Janata, K. Booksh, and L. Obando, "Chemical Sensors for Portable, Handheld Field Instruments" IEEE Sensors J., Vol. 1, No. 4, pp. 256-274, 2001.
20. A.M. Zoubir, I.J. Chant, C.L. Brown, B. Barkat, and C. Abeynayake, "Signal Processing Techniques for Landmine Detection Using Impulse Ground Penetrating Radar" IEEE Sensors J., Vol. 2, No. 1, pp. 41-51, 2002.
21. Luís M. L. Oliveira and Joel J. P. C. Rodrigues ., "Wireless Sensor Networks: a Survey on Environmental Monitoring," Journal of Communications, vol. 6, no. 2, April 2011.
22. D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," In Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCOM '99), August 1999, Seattle, Washington.

23. D. Estrin, L. Girod, G. Pottie, M. Srivastava, "Instrumenting the world with wireless sensor networks," In Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001), Salt Lake City, Utah, May 2001.
24. Alcherio Martinoli, Kjerstin Easton, Modeling Swarm Robotic Systems, California Institute of Technology, pp 1-2.
25. Ron Fulbright, Larry M. Stephens, CLASSIFICATION OF MULTIAGENT SYSTEMS, Electrical and Computer Engineering Department, University of South Carolina, Columbia, pp 1-2.
26. Toshiyuki Yasuda, Multi-Robot Systems, Trends and Development, InTech, 2011.
27. Ming Tan, MultiAgent Reinforcement Learning: Independent vs. Cooperative Agents, Waltham, pp 1-2.
28. J.T. Feddela, C. Lewis, D.A. Schoenwald, "Decentralized Control of Cooperative Robotic Robots : Theory and Application", IEEE Transactions on Robotics and Automation, Special Issue on Advances in Multirobot Systems, Vol. 18, N 5, October 2002, pp. 852-863.
29. A.K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Speltzer, C. J. Taylor, "A Vision-Based Formation Control Framework", IEEE Transactions on Robotics and Automation, Special Issue on Advances in Multirobot Systems, Vol. 18, N 5, October 2002, pp. 813-825.
30. Moises Lejter, Thomas Dean, A Framework for the Development of Multi-Agent Architectures, Department of Computer Science, Brown University, pp 1-2.
31. Onn Shehory, Architectural Properties of MultiAgent Systems, The Robotics Institute Carnegie Mellon University, Pittsburgh, December 1998, pp 3-10.
32. David Kinny, Michael George, Modelling and Design of Multi-Agent Systems, Australian Artificial Intelligence Institute, November 1996, pp 2-12.

33. Meparishvili B., Synergy-based Approach to Social System Modeling, Georgian Technical University, Transactions, Automated Control Systems ISSN 1512-3979, #1(10)2011, 154-157 p.
34. Tucker Balch, Hierarchic Social Entropy: An Information Theoretic Measure of Robot Group Diversity The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3891, 2-8 p.
35. Kervalishvili P., Meparishvili B., and Janelidze G. “*Self-Organization Modelling Of Multi-Agent Systems*”. SynEnergy Forum (S.E.F.) -2 The conference for International Synergy in Energy, Environment, Tourism and Information Technology. Spetses, Greece, 2009.
36. T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet, “Novel type of phase transition in a system of self-driven particles,” *Physical Review Letters*, vol. 75, no. 6, 1995.
37. Shannon, C.E. *The Mathematical Theory of Communication*, University of Illinois Press, 1949.
38. Georgian Technical University, Transactions, Automated Control Systems ISSN 1512-3979, # 2(18)2014, 7-11 p.
39. Paata Kervalishvili, Synergetic Modeling of Sustainable Development, 2015, 3-6 p.
40. Evangelou IE, Hadjimitsis DG, Lazakidou AA, (2001), Clayton C, Data Mining and Knowledge Discovery in Complex Image Data using Artificial Neural Networks, Workshop on Complex Reasoning an Geographical Data, Cyprus.
41. Elham Karoussi, Data Mining, K-Clustering Problem, University of Agder, Faculty of Engineering and Science 2012, p 23-24.
42. https://en.wikipedia.org/wiki/Cluster_analysis, უკანასკნელად იქნა გადამოწმებული - 01.02.2016.
43. https://en.wikipedia.org/wiki/Voronoi_diagram, უკანასკნელად იქნა გადამოწმებული - 01.02.2016.

44. ADAM DOBRIN, A REVIEW OF PROPERTIES AND VARIATIONS OF VORONOI DIAGRAMS, p 2-3.
45. Franz Aurenhammer, Rolf Klein, Institut für Grundlagen der Informationsverarbeitung, Technische Universität Graz, Austria, FernUniversität Hagen, Praktische Informatik VI, Germany, p 1-2.
46. Roberto Tamassia, Computational Geometry, 1993, p 1-5.
47. Diane Souvaine, Michael Horn, Julie Weber, Computational Geometry, Tufts University, Spring 2005, p 1-3.
48. Franz Aurenhammer, Rolf Klein, Institut für Grundlagen der Informationsverarbeitung, Technische Universität Graz, Austria, FernUniversität Hagen, Praktische Informatik VI, Germany, p 6-16.
49. <http://bug.kpi.ua/stud/work/RGR/DATAMINING/neuralnetskhohonencards.html>, უკანასკნელად იქნა გადამოწმებული - 13.06.2014.
50. <http://www.intuit.ru/studies/courses/6/6/lecture/180>, უკანასკნელად იქნა გადამოწმებული - 13.06.2015.
51. https://en.wikipedia.org/wiki/K-means_clustering, უკანასკნელად იქნა გადამოწმებული - 01.03.2016.
52. Elham Karoussi, Data Mining, K-Clustering Problem, University of Agder, Faculty of Engineering and Science 2012, p 26-28.
53. https://en.wikipedia.org/wiki/Fuzzy_clustering, უკანასკნელად იქნა გადამოწმებული - 01.03.2016.
54. https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm, უკანასკნელად იქნა გადამოწმებული - 01.03.2016.
55. <https://en.wikipedia.org/wiki/K-means%2B%2B>, უკანასკნელად იქნა გადამოწმებული - 01.03.2016.
56. https://en.wikipedia.org/wiki/K-d_tree, უკანასკნელად იქნა გადამოწმებული - 01.03.2016.

57. Renato Cordeiro de Amorim, Constrained Clustering with Minkowski Weighted K-Means, Department of Computer Science and Information Systems, Birkbeck University of London, p 1-2.
58. Renato Cordeiro de Amorim, Boris Mirkin, Selecting the Minkowski exponent for intelligent K-Means with feature weighting, p 1-4.
59. Thomas Weise, Global Optimization Algorithms - Theory and Application, 2009, pp 1-50.
60. Kalyanmoy Deb, Multi-Objective Optimisation using Evolutionary Algorithms, Indian Institute of Technology, India, pp 1-45.
61. Weise T. Global Optimization Algorithms - Theory and Application. 2009, pp 141-142.
62. Xinjie Yu, Mitsuo Gen. Introduction to Evolutionary Algorithms. London: Springer, 2010, pp 6-8.
63. Sivanandam S.N., Deepa S.N. Introduction to Genetic Algorithms. Berlin, Heidelberg: Springer, 2008, pp 15-16.
64. Mitchell Melanie. An Introduction to Genetic Algorithms. London, England: Cambridge, Massachusetts, 1999, pp 117-118.
65. Sivanandam S.N., Deepa S.N. Introduction to Genetic Algorithms. Berlin, Heidelberg: Springer, 2008, pp 29-31.
66. Mitchell Melanie. An Introduction to Genetic Algorithms. London, England: Cambridge, Massachusetts, 1999, pp 124-128.
67. Mitchell Melanie. An Introduction to Genetic Algorithms. London, England: Cambridge, Massachusetts, 1999, pp 128-130.
68. Weise T. Global Optimization Algorithms - Theory and Application. 2009, pp 147-150.
69. <http://www.obitko.com/tutorials/genetic-algorithms/crossover-mutation.php>,
უკანასკნელად იქნა გადამოწმებული - 20.06.2014.
70. http://en.wikipedia.org/wiki/Genetic_algorithm, უკანასკნელად
გადამოწმებული - 20.06.2014.

71. <http://www.obitko.com/tutorials/genetic-algorithms/index.php>,
უკანასკნელად იქნა გადამოწმებული - 20.06.2014.
72. http://en.wikipedia.org/wiki/Genetic_programming, უკანასკნელად იქნა
გადამოწმებული - 20.06.2014.
73. http://en.wikipedia.org/wiki/Genetic_programming#Representation,
უკანასკნელად იქნა გადამოწმებული - 20.06.2014.
74. [http://cswwww.essex.ac.uk/staff/poli/gp-field-
guide/22InitialisingthePopulation.html#dx1-79](http://cswwww.essex.ac.uk/staff/poli/gp-field-guide/22InitialisingthePopulation.html#dx1-79), უკანასკნელად იქნა
გადამოწმებული - 20.06.2014.
75. Weise T. Global Optimization Algorithms - Theory and Application. 2009, pp
162-165.
76. Girish Kumar Jha. ARTIFICIAL NEURAL NETWORKS AND ITS
APPLICATIONS. Delhi, 9 p.
77. Girish Kumar Jha. ARTIFICIAL NEURAL NETWORKS. Delhi: Indian
Agricultural Research Institute, 10 p.
78. Nedjah N. Mourelle L.M. Swarm Intelligent Systems. Berlin: Springer, 2006, pp
7-7.
79. <http://tracer.uc3m.es/tws/psso/neighborhood.html#Watt99>, უკანასკნელად
იქნა გადამოწმებული - 20.11.2013.
80. Swagatam Das. Ajith Abraham. Amit Konar. Particle Swarm Optimization and
Differential Evolution Algorithms: Technical Analysis, Applications and
Hybridization Perspectives. pp 6-9.
81. Nedjah N. Mourelle L.M. Swarm Intelligent Systems. Berlin: Springer, 2006, pp
5-6.
82. Maurice Clerc. Particle Swarm Optimization. Great Britain: ISTE Ltd, 2006, pp
37-38.
83. Dorigo M. Stutzle T. Ant Colony Optimization. United States of America:
Massachusetts Institute of Technology, 2004, pp 2-5.

84. Oded Maimon, Lior Rokach. Soft Computing for Knowledge Discovery and Data Mining, Springer US, 2008, pp 284-286.
85. <http://www.emeraldinsight.com/journals.htm?articleid=1740598>,
უკანასკნელად იქნა გადამოწმებული - 13.06.2014.
86. [http://www.scielo.br/scielo.php?pid=S1516-14392014005000023](http://www.scielo.br/scielo.php?pid=S1516-14392014005000023&script=sci_arttext)
&script=sci_arttext, უკანასკნელად იქნა გადამოწმებული - 13.06.2014.