საგრანტო პროექტის - "გადაწყვეტილების მიღების კომპიუტერული სისტემა საინვესტიციო პოლიტიკის შემუშავებისათვის" (074-13) - სრული ანგარიში

პროექტის მიზანია, შეიქმნას ინტერაქტიული პროგრამული სისტემა მაკროსისტემების იმიტაციური მოდელირებისა და მართვისთვის. მაკროსისტემას ვუწოდებთ ისეთ სისტემას, რომელიც, როგორც ერთიანი, ავლენს სხვა ბუნებასა და თვისებებს, ვიდრე მისი შემადგენელი ნაწილები. უმეტესწილად, სისტემა, როგორც ერთიანი, განეკუთვნება დეტერმინირებულ სისტემათა კლასს, მაშინ, როდესაც მისი შემადგენელი ნაწილების ქცევა სტოქასტიკურია. საინვესტიციო პორტფელის ფორმირების პროცესი, ურბანული სისტემები, სივრცული ეკონომიკური სისტემები, ბიოლოგიური, სოციალური, კომპიუტერულ ქსელებში ინფორმაციული ნაკადის გავრცელების დინამიკა და სხვა მიეკუთვნებიან მაკროსისტემათა კლასს.

თავი I. მაკროსისტემების-სივრცული ევოლუციური პროცესების- მათემატიკური მოდელირების მეთოდები

1.1 სივრცული ევოლუციური პროცესების მათემატიკური მოდელირება ენტროპიის მაქსიმიზაციის პრინციპის საფუძველზე.

ჩვენს მიერ შექმნილი მაკროსისტემების მათემატიკური მოდელების აგების პრინციპები არის უნივერსალური, რომელიც ეფუძნება ენტროპიის მაქსიმიზაციის პრინციპს და ინფორმაციის ანალიზის ენტროპიულ მეთოდს. აღნიშნულ მიმართულებას საფუძველი ჩაუყარა ლ. ბოლცმანმა და იგი მდგომარეობს შემდეგში.

განვიხილოთ 6-განზომილებიანი სივრცე L^i, რომელშიც მოძრაობს i ნაწილაკი. ნაწილაკის მდგომარეობა ამ სივრცეში ხასიათდება 6 განზომილებიანი ვექტორით. $x^i = \{q_1, q_2, q_3; p_1, p_2, p_3\}$; სადაც; $\overline{q} = (q_1, q_2, q_3)$ და $\overline{p} = (p_1, p_2, p_3)$ შესაბამისად ნაწილაკის კორდინატებისა და იმპულსების ვექტორებია. განვსაზღვროთ G სივრცე, როგორც L^i სივრცეთა პირდაპირი ნამრავლი, რომელიც მოიცავს n ერთგვაროვან ნაწილაკს.

განვიხილოთ G სივრცეში $\gamma(t)$ წერტილი, რომელსაც უწოდებენ ფაზურ წერტილს და რომელიც ახასიათებს n ერთგვაროვან ნაწილაკს დროის t მომენტისათვის.

თუ ჩვენ მოვახდენთ $\gamma(t)$ წერტილის პროექტირებას L^i სივრცეებზე მივიღებთ $\gamma_1(t),\ldots,\gamma_n(t)$ ფაზურ წერტილებს, რომლებიც ახასიათებს შესაბამისი ნაწილაკის მდგომარეობას დროის t მომენტისათვის.

თუ დავუშვებთ, რომ ყოველი L^i სივრცე ერთნაირადაა მოწყობილი, შეგვიძლია განვიხილოთ მხოლოდ ერთი სივრცე L და იქ განვათავსოთ ფაზური წერტილები $\gamma_1(t),\ldots,\gamma_n(t)$ დროის შესაბამისი მომენტებისათვის. ეს წერტილები წარმოქმნიან $S(t)$ სიმრავლეს, რომელიც შეიცავს n ფაზურ წერტილს. თითოეული წერტილი კი, ესაა ვექტორი $\{q(t), p(t)\}$.

დროის ფიქსირებული t მომენტისათვის $S(t)$ სიმრავლე შეიძლება დავახასიათოთ $f(q,p,t)$ განაწილების სიმკვრივის ფუნქციით. დროის სხვადასხვა t_1 და t_2 მომენტებისათვის გვექნება ფაზური წერტილების $S(t_1)$ და $S(t_2)$ სიმრავლეები. ფაზური წერტილის მოძრაობისას $S(t)$ სიმრავლე ევოლუციონირებს L სივრცეში.

ჯ. მაქსველმა დაამტკიცა, რომ გარკვეული დროის შემდეგ ფაზური წერტილების განაწილება $f(q,p,t)$,L სივრცეში მიისწრაფის სტაციონარული განაწილებისაკენ. აქედან გამომდინარე, $f^\circ(q,p)$ განაწილება წარმოადგენს განხილული სისტემის მაკრო მახასიათებელს.

ცხადია, თუ $f(q, p,t) \rightarrow f^\circ(q,p)$, როცა $t \rightarrow \infty$, მაშინ აღნიშნული გადასვლის პროცესი შეიძლება დავახასიათოდ რაღაც ფუნქციით, რომელიც აღწევს მაქსიუმუმს, როცა $f(q, p,t) = f^\circ(q,p)$. ამ $f(q,p,t)$ ფუნქციის ევოლუცია განისაზღვრება ბევრი სხვადასხვა ფაქტორების ზემოქმედებით, რაც განპირობებულია ნაწილაკების ურთიერთქმედებებით.

ლ. ბოლცმანს ეკუთვნის განაწილების $f(q,p,t)$ ფუნქციის ევოლუციის აღმწერი კინეტიკური განტოლება. ამ განტოლების ამოხსნათა თვისებების კვლევისათვის კი, გამოყენებულ იქნა ენტროპია H(t), რომელიც აღნიშნულ ტერმინებში გამოისახება შემდეგი სახით

$$H(t) = -\int f(q,p,t)\cdot \ln f(q,p,t)dpdq \ .$$

აღმოჩნდა, რომ ბოლცმანი კინეტიკური განტოლების ამოხსნათა სიმრავლეზე $dH / dt > 0$ და უდრის ნულს, როდესაც $f(q,p,t) = f^\circ(q,p)$. შესაბამისად, ბოლცმანის კინეტიკური განტოლებისათვის $H(t)$ ენტროპია არის ლიაპუნოვის ფუნქცია. ეს წარმოადგენს ბოლცმანის ცნობილ H-თეორემას, რომლის მიხედვითაც, ნაწილაკების განაწილება L-სივრცეში სტაციონარულია მაქსიმალური ენტროპიის დროს.

ახლა თუ წარმოვიდგენთ, რომ L – სივრცე წარმოადგენს $u_1,\ldots.u_m$ უჯრედების ერთობლიობას და თითოეულ მათგანს დროის ყოველ t მომენტში ვახასიათებთ მათში მოხვედრილი ნაწილაკების რაოდენობით $N_i(t)$, მაშინ ნაწილაკების განაწილება L – სივრცეში განისაზღვრება $N(t) = \{N_1(t),\ldots., N_m(t)\}$ ვექტორით. H-

თეორემის თანახმად, $N(t) \to N°$, როცა $t \to \infty$, ანუ ენტროპია, $H = -\sum_{i=1}^{m} N_i \ln N_i$ აღწევს მაქსიმუმს $N°$-ის დროს.

იმისდამიხედვით, თუ როგორია ნაწილაკების ქცევა მიკროდონეზე, მაკროსისტემები შეიძლება დავყოთ ორ ჯგუფად: მაკროსისტემები ნაწილაკების სტოქასტური ქცევით და მაკროსისტემები ნაწილაკების ე.წ. კონიუნქტურული ქცევით. პირველ ჯგუფს მიეკუთვნება ისეთი მაკროსისტემები, რომლებსაც გააჩნია m რაოდენობის ენერგეტიკული უჯრედი (u), რომლებიც ივსება გარკვეული რაოდენობის ერთგვაროვანი ნაწილაკებით (N_i). ამასთან ნაწილაკების განაწილებას უჯრედებში აქვს შემთხვევითი ხასიათი.

მაკროსისტემები კონიუნქტურული ქცევით ისეთი სისტემებია, როდესაც ნაწილაკების განაწილება ენერგეტიკულ უჯრედებში ხორციელდება გარკვეული სარგებლიანობის ფუნქციის შესაბამისად, რომლებიც მიეწერებათ ნაწილაკებს. ამასთანავე, ყოველი შემდეგი განაწილება დამოკიდებულია წინაზე.

არსებობს მაკროსისტემები სხვადასხვა სტატისტიკით.

ფერმის სტატისტიკის შემთხვევაში ყოველ u_i უჯრედში შეიძლება იმყოფებოდეს არაუმეტეს ერთი ნაწილაკისა ალბათობით a_i; ამასთან, თითოეული ნაწილაკის არჩევანი დამოუკიდებელია მეორე ნაწილაკის არჩევანისაგან; და ბოლოს, აპრიორული ალბათობები a_k ყველა ნაწილაკისათვის ერთნაირია.

აინშტაინის სტატისტიკის შემთხვევაში ყოველ უჯრედში შეიძლება იმყოფებოდეს ნაწილაკების ნებისმიერი სასრული რაოდენობა.

ბოლცმანის სტატისტიკა ფერმისა და აინშტაინის სტატისტიკათა ზღვრული შემთხვევაა. აქ u_i უჯრედის ტევადობა გაცილებით დიდია მათში ნაწილაკების რაოდენობის შესაძლო სიდიდეზე. ჩვენ განვიხილავთ მაკროსისტემებს ბოლცმანის სტატისტიკით.

ენტროპიული დეკომპოზიცია: ეკონომიკური პროცესის მამოძრავებელ ძალას წარმოადგენენ ეკონომიკური აგენტები, შუამავლები, რომელთა ურთერთქმედება არადეტერმინირებულია. ამასთანავე, ამ ურთიერთქმედებათა ერთობლიობა ქმნის ერთიანი სისტემის გარკვეულ მდგომარეობას. ანუ, ქცევის ინდივიდუალური ტიპი, გარდაიქმნება ქცევის კოლექტიურ ტიპად. ასეთ სისტემებს, როგორც ზემოთ აღვნიშნეთ, ვუწოდებთ მაკროსისტემებს.

ზემოთ მოყვანილი მათემატიკური მოდელირების მეთოდის შესაბამისად ეკონომიკური სივრცე E, იყოფა რეგიონებად N. თითოეული რეგიონი შეიცავს F_i რაიონს. ვაკეთებთ დაშვებას, რომ როგორც რეგიონები, ასევე რაიონები ხასიათდე-

ბა ერთგვაროვანი მაჩვენებლებით, შესაბამისად E_i და E_{ij}. ასეთ მაჩვენებლებად შეიძლება განვიხილოთ მოსახლეობის რაოდენობა, მომსახურების კონკრეტული სფეროების რაოდენობა, წარმოების ინტენსივობა და სხვა.

რეგიონულ და რაიონულ მაჩვენებლებს შორის ბალანსური თანაფარდობებს აქვს შემდეგი სახე:

$$E_i = \sum_{j=1}^{F_i} E_{ij} \ , \quad i = 1,2,\ldots,N \qquad (1.1)$$

(ვგულისხმობთ, რომ განსახილველ ეკონომიკურ სივრცეში აწარმოებენ r ტიპის პროდუქტს ("პროდუქტი" აქ განზოგადოებული ცნებაა).

ვგულისხმობთ, რომ i-ური რეგიონის „შესაძლებლობანი" k ტიპის პროდუქციის წარმოებისათვის დამოკიდებულია დანარჩენი ყველა სხვა რეგიონის მაჩვენებლებზე. ეს სიდიდე აღვნიშნოთ $Q_{ik}(E)$. ვგულისხმობთ, რომ ანალოგიური დაშვებებია მიღებული რაიონების მიმართაც. რაიონის საწარმოო შესაძლებლობანი აღვნიშნოთ $q_{ijk}\left(\overline{E}\right)$, სადაც $\overline{E} = \{E_{11},\ldots,E_{1F_1},\ldots,E_{N1},\ldots,E_{NF_N}\}$. შესაბამისად გვექნება.

$$Q_{ik}\left(E\right) = \sum_{j=1}^{F_i} q_{ijk}\,(\overline{E}) \quad i = 1,..,N;\ k = 1,\ldots,r \qquad (1.2)$$

ვთვლით, რომ ცნობილია რეგიონული მაჩვენებლები, ხოლო უცნობია რაიონული მაჩვენებლები $\overline{E}$, რომლებიც უნდა განვსაზღვროთ მიღებული ბალანსური თანაფართობების გათვალისწინებით. ზემოთ მიღებული დაშვებების დროს ბალანსურ თანაფარდობებს აკმაყოფილებს რაიონული მაჩვენებლების მთელი რიგი სიმრავლეები. ცხადია, ამ დროს გვაქვს ამოხსნათა არა ერთადერთი, არამედ მრავალი სიმრავლე. ისმის კითხვა, როგორ უნდა მოვძებნოთ ერთადერთი ამონახსნი. ამ ამოცანის გადაჭრისათვის ვიყენებთ ენტროპიის მაქსიმიზაციის პრინციპს. ამასთანავე, რადგანაც არსებობს გარკვეული განუზღვრელობა რეგიონული მაჩვენებლების განაწილებისათვის რაიონების მიხედვით, განუზღვრელობის საზომად და შესაბამისად კრიტერიუმად ავიღოთ ბოლცმანის განზოგადებული ინფორმაციული ენტროპია [1]

$$H(\overline{E}) = -\sum_{i=1}^{N}\sum_{j=1}^{F_i} E_{ij} \cdot \ln\frac{E_{ij}}{a_{ij}e} \qquad (1.3)$$

აქ $\overline{E}$ მატრიცაა, რომლის E_{ij} ელემენტები ახასიათებენ რაიონული მაჩვენებლების განაწილებას.

როდესაც ენტროპია $H(\overline{E})$ აღწევს მაქსიმუმს, მაშინ რაიონული მახასიათებლების განაწილება არის ყველაზე მეტად განუზღვრელი. ამ დროს სისტემაში მყარდება მდგრადი წონასწორობა. იმ შემთხვევაში, თუ გვაქვს დამატებითი ინფორმაცია რაონების შესახებ, მაშინ მათი გათვალისწინება შესაძლებელია a_{ij} - პარამეტრების საშუალებით.

ზემოთ აღწერილ პროცედურას - რეგიონალური მაჩვენებლების განაწილებას რაიონების მიხედვით - ვუწოდებთ ენტროპიიული დეკომპოზიციის მეთოდს [2]. აღნიშნული მიდგომის საფუძველზე, ჩვენს მიერ შექმნილია ინსვესტირების პროცესის მათემატიკური მოდელი:

დავუშვათ, საინვესტიციო სისტემა წარმოადგენს ერთმანეთთან დაკავშირებული n ქვესისტემას. თოთოეული ქვესისტემა – i, დროის მოცემულ t მომენტისათვის დავახასიათოთ იმ ინვესტიციებით $I_i(t)$ რომლებიც იდება ამ ქვესისტემაში. ინვესტიციების ნაკადი $I_i(t)$ თითოეულ ქვესისტემაში განისაზღვრება გამოსახულებით

$$I_i = \sum_{j=1,\, j\neq i}^{n} (x_{ji}(t) - x_{ij}(t)), \tag{1.4}$$

სადაც: x_{ij} - არის ინვესტიციის პორციაა i ქვესისტემიდან j ქვესისტემაში. ცხადია, ინვესტიციების პორციათა რაოდენობა არის შემთხვევითი და თოთოეული ქვესისტემაში მათი მიმართვა ხორციელდება გარკვეული აპრიორული ალბათობით a_{ij} ($\sum_{j=1}^{n} a_{ij} = 1.\ i = 1,2 ... n.$ აქ a_{ii} ის ალბათობაა, რომელიც განსაზღვრავს რა რაოდენობის ინვესტიცია დარჩება იქვე, ამავე i-ურ ქვესისტემაში). ენტროპიის სიდიდე დროის მოცემული t მომენტისათვის განისაზღვრება (3) გამოსახულებით, სადაც E_{ij}-ს მაგივრად გვაქვს x_{ij} ცვლადები. x_{ij}-ინვესტიციის პორციათა სიდიდეები შეირჩევა ისე, რომლის დროსაც $H(x)$ იყოს მაქსიმალური, შესაბამისი შეზღუდვების დროს.

აპრიორული ალბათობების შეფასებას (3) გამოსახულებაში ვახორციელებთ შემდეგი პროცედურით. ვთქვათ, რეგიონში ინვესტორთა სიმრავლეა $Q(q_1,...,q_n)$. რომელთა ინტერესების სფეროა $E(e_1,...,e_m)$ ეკონომიკის (წარმოების) სფეროები. კავშირს Q და E სიმრავლეებს შორის ვახასიათოთ ინციდენტურობის λ მატრიცით. $Q = \lambda E.$ სადაც λ ინციდენტურობის მატრიცაა, რომლის ელემენტებია

$$\lambda_{ij} = \begin{cases} 1, & \text{როცა } q_i \text{ ინვესტორს აქვს } e_j \text{ ინტერესი} \\ 0, & \text{წინააღმდეგ შემთხვევაშ ი} \end{cases}$$

თუ $E^q - q$ ინვესტორის ინტერესთა სიმრავლეა, ხოლო $E_k -$იმ ინტერესების სიმრავლე, რომელთა დაკმაყოფილება შეიძლება k-ურ რაიონში, მაშინ $E_k{}^q$ აღვნიშნოთ q ინვესტორის იმ ინტერესების სიმრავლე, რომელთა დაკმაყოფილება შესაძლებელია k რაიონში. ანუ, $E_k{}^q = E_k \bigcap E^q$. თუ a_i^q-თი აღვნიშნავთ $e_i^q \in E^q$ ინტერესის წონას $\left(\sum e_i^q = 1\right)$, მაშინ $E_k{}^q -$ინტერესების სიმრავლის წონა იქნება $\sum a_i^q$, რაც განსაზღვრავს აპრიორული ალბათობის მნიშვნელობას (1.3) გამოსახულებაში.

1.2 ევოლუციური განვითარების სისტემა - როგორც გეომეტრიული სტრუქტურა. ალგებრული და ტოპოლოგიური მთოდების გამოყენება მაკროსისტემების მათემატიკური მოდელირებისათვის

მაკროსისტემათა სივრცული ევოლუციური პროცესების ფუნქციონირებისას მათი გეომეტრიული სტრუქტურის მნიშვნელობას პირველად მიაქცია ყურადღება რ.ეტკინმა [3-5]. ირკვევა, რომ სისტემას ფუნქციონირების ხარისხი გარდა ფიზიკური პარამეტრებისა განისაზღვრება აგრეთვე მის ცალკეულ ნაწილთა „ბმულობითაც“, რომლის შესწავლა ხერხდება კომბინატორული ტოპოლოგიის აპარატის მეშვეობით.

კომბინატორული ტოპოლოგიის რამდენიმე ძირითადი ცნება.

ვთქვათ, $a_0, a_1, \ldots, a_k$ ევკლიდეს n-განზომილებიანი სივრცის წერტილებია. ამასთან, სისტემა

$$a_0 - a_1,\ a_2 - a_0, \ldots,\ a_k - a_0$$

წრფივად დამოუკიდებელია. R^n სივრცის იმ x წერტილთა A სიმრავლეს, რომელთათვისაც

$$x = \lambda_0 a_0 + \lambda_1 a_1 + \ldots + \lambda_k a_k \tag{1.5}$$

სადაც $\lambda_0, \lambda_1, \ldots, \lambda_k$ ისეთი ნამდვილი რიცხვებია, რომ

$$\lambda_0 + \lambda_1 + \ldots + \lambda_k = 1 \tag{1.6}$$

$$\lambda_0, \lambda_1, \ldots, \lambda_k \geq 0 \tag{1.7}$$

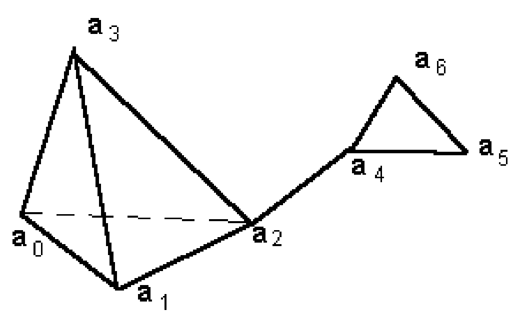
ეწოდება k-განზომილებიანი სიმპლექსი, რომელსაც ავღნიშნავთ [a_0, a_1,..., a_k] სიმბოლოთი. მტკიცდება, რომ $\lambda_0,\lambda_1,...,\lambda_k$ რიცხვები ცალსახად განისაზღვრება x წერტილით და მათ ეწოდებათ x-ის ბარიცენტრული კოორდინატები. a_0, a_1,..., a_k წერტილებს ეწოდებათ A სიმპლექსის წვეროები, ხოლო {a_0, a_1,..., a_k} სიმრავლის ნებისმიერი s+1 ელემენტიანი ქვესიმრავლის მიერ განსაზღვრულ სიმპლექსს ეწოდება A სიმპლექსის s-განზომილებიანი წახნაგი. 0-განზომილებიანი სიმპლექსი წერტილია, 1-განზომილებიანი სიმპლექსი წარმოადგენს მონაკვეთს, 2-განზომილებიანი სიმპლექსი_ სამკუთხედს და 3_განზომილებიანი სიმპლექსი _ ტეტრაედრს.

ვიტყვით, რომ ორი სიმპლექსი წესიერადაა განლაგებული, თუ ისინი არ გადაიკვეთებიან, ან მათი თანაკვეთა ორივეს წახნაგს წარმოადგენს.

R^n სივრცის სიმპლექსების რაიმე K არაცარიელ სიმრავლეს ეწოდება გეომეტრიული სიმპლიციალური კომპლექსი, თუ სრულდება პირობები:

1) თუ $A \in K$ და A' არის A-ს წახნაგი, მაშინ $A' \in K$,

2)თუ $A, B \in K$, მაშინ ისინი წესიერადაა განლაგებულნი.



სურ. 1

K-ში შემავალი სიმპლექსების განზომილებებს შორის უდიდესს ეწოდება K-ს განზომილება. K-ს 0-განზომილებიან სიმპლექსებს ეწოდებათ K-ს წვეროები. სურ. 1-ზე გამოსა-ხულია R^3-ში 3-განზომილებიანი კომპლექსის მაგალითი. ეს კომპლექსი შესდგება 7 წვეროსაგან: a_0, a_1, a_2, a_3, a_4, a_5, a_6; 10 1-განზომილებიანი სიმპლექსისაგან: [a_0,a_1], [a_0,a_2], [a_0,a_3], [a_1,a_2], [a_1,a_3], [a_2,a_3], [a_2,a_4], [a_4,a_5], [a_4,a_6], [a_5,a_6]; 5 2-განზომილებიანი სიმპლექსისაგან: [a_0,a_1,a_2], [a_0,a_1,a_3], [a_0,a_2,a_3], [a_1,a_2,a_3], [a_4,a_5,a_6] და ერთი 3-განზომილებიანი [a_0,a_1, a_2,a_3] სიმპლექსისაგან.

ადვილი დასანახია, რომ ყოველი გეომეტრიული სიმპლიციალური კომპლექსი მოიცემა თავისი წვეროებით და წვეროების ერთობლიობების გარკვეული სიმრავლით. აქედან გამომდინარე, ყოველ გეომეტრიულ სიმპლიციალურ კომპლექსს

შეესაბამება სიმპლიციალური (ზოგჯერ ამბობენ აბსტრაქტული სიმპლიციალური) კომპლექსი, რომელიც განისაზღვრება შემდეგნაირად.

სიმპლიციალური კომპლექსი არის წყვილი (X* K), სადაც X არის რაიმე სასრული სიმრავლე, ხოლო K არის X-ის ქვესიმრავლეების სიმრავლე და სრულდება პირობები:

1) X-ის ყოველი ერთელემენტიანი სიმრავლე შედის K-ში*

2) თუ $\sigma \in X$ და $\sigma' \subset \sigma$, მაშინ $\sigma' \in X$.

X-ის ელემენტებს ეწოდება კომპლექსის წვეროები, ხოლო K-ს ელემენტებს - სიმპლექსები. თუ σ სიმპლექსია და $\sigma' \subset \sigma$, მაშინ σ'-ს ეწოდება σ-ის წახნაგი. თუ σ-ში შემავალი ელემენტების რაოდენობაა r, მაშინ σ-ს ეწოდება (r-1)-განზომილებიანი სიმპლექსი. K-ში შემავალი სიმპლექსების განზომილებებს შორის უდიდესს ეწოდება სიმპლიციალური კომპლექსის განზომილება.

როგორც აღვნიშნეთ, ყოველ გეომეტრიულ m-განზომილებიან K კომპლექსს შეესაბამება m-განზომილებიანი $\overline{K}$ აბსტრაქტული სიმპლიციალური კომპლექსი. ასეთ დროს ამბობენ, რომ K არის $\overline{K}$-ის გეომეტრიული რეალიზაცია. მტკიცდება, რომ ყოველ m-განზომილებიან აბსტრაქტულ სიმალიციალურ კომპლექსს აქვს გეომეტრიული რეალიზაცია R^{2m+1} სივრცეში. ქვემოთ ყველგან ვიხმართ ტერმინს „სიმპლიციალური კომპლექსი“ და კონტექსტიდან გასაგები იქნება გეომეტრიულ თუ აბსტრაქტულ სიმპლიციალურ კომპლექსზეა ლაპარაკი.

ვთქვათ, K m-განზომილებიანი სიმპლიციალური კომპლექსია და 0≤q≤m. შემოვიღოთ K-ს იმ სიმპლექსების K_q სიმრავლეზე, რომელთა განზომილება არ არის q-ზე ნაკლები, $\sim_q$ ექვივალენტობის მიმართება შემდეგნაირად: ვიტყვით, რომ σ და σ' სიმპლექსები ექვივალენტურია, თუ არსებობს სიმპლექსების ისეთი

$$\sigma_1, \sigma_2, ..., \sigma_s \tag{1.8}$$

მიმდევრობა, რომ

$$\sigma_1 = \sigma, \ \sigma_s = \sigma'$$

და

$$\dim(\sigma_i \cap \sigma_{i+1}) \geq q, \quad i = 1,2,...,(s-1).$$

ცხადია, რომ განსაზღვრული მიმართება მართლაც ექვივალენტობის მიმართებაა. $\sim_q$-ს ექვივალენტობის კლასების რაოდენობა აღვნიშნოთ Q_q-თი.

ვექტორს $(Q_0, Q_1,..., Q_m)$ ეწოდება K-ს სტრუქტურული ვექტორი, ხოლო თვით Q_q რიცხვების პოვნის პროცედურას ეწოდება კომპლექსის Q-ანალიზი. მაგალითად, სურ. 1-ზე გამოსახული კომპლექსის სტრუქტურული ვექტორია (1, 3, 2, 1). ამ კომპლექსში რომ არ შესულიყო ერთადერთი 3-განზომი-ლებიანი სიმპლექსი, მაშინ სტრუქტურული ვექტორი იქნებოდა (1, 3, 5).

$\sim_0$-ს ექვივალენტობის კლასებს ეწოდებათ ბმულობის კომპონენტები ამრიგად, Q_0 არის ბმულობის კომპონენტების რაოდენობა. მისი პოვნა შეიძლება კიდევ სხვანაირად, Q -ანალიზის ჩატარების გარეშეც, ე.წ. 0-ანი ჰომოლოგიების ჯგუფის გამოთვლით. მოვიყვანოთ სიმპლიციალური კომპლექსის ჰომოლოგიების ჯგუფების განსაზღვრება.

ვთქვათ, $\sigma=[a_0, a_1...., a_k]$ არის K კომპლექსის k-განზომილებიანი სიმპლექსი. განვიხილოთ σ-ს წვეროების ყველა გადანაცვლებების I სიმრავლე და მასზე შემოვიღოთ ექვივალენტობის მიმართება: ξ_1 გადანაცვლება ექვივალენტურია ξ_2 გადანაცვლებისა, თუ $\xi_2^{-1}\xi_1$ ლუწია. ამ ექვივალენტობის მიმართებით I სიმრავლე დაიყოფა ორ კლასად. თითოეულ კლასს ეწოდება σ-ს ორიენტაცია. ვიტყვით, რომ σ ორიენტირებულია თუ ფიქსირებულია მისი ორიენტაცია. ორიენტირებულ σ სიმპლექსებს ჩავწერთ

$$(a_{i_0}, a_{i_1}, ..., a_{i_k})$$

სიმბოლოთი, სადაც $(i_0, i_1, ... i_k)$ არის σ-ს ფიქსირებული ორიენტაციის ერთ-ერთი წარმომადგენელი.

ყოველი $0 \le q \le m$-სათვის განვიხილოთ აბელური $C_q(k)$ ჯგუფი, რომელიც წარმოქმნილია კომპლექსის ყველა q-განზომილებიანი ორიენტირებული სიმპლექსით და რომელშიც სრულდება თანაფარდობები:

$$\sigma' + \sigma'' = 0$$

სადაც σ' და σ'' არის ერთი და იგივე სიმპლექსი სხვადასხვა ორიენტაციით. $C_q(K)$ -ს ელემენტებს უწოდებენ q-განზომილებიან ჯაჭვებს. ცხადია რომ ყოველი ჯაჭვი მოიცემა სახით

$$\alpha=k_1\sigma_1+k_2\sigma_2+...+k_s\sigma_s$$

სადაც $\sigma_1, \sigma_2, ..., \sigma_s$ K .ყველა q-განზომილბიანი სიმპლექსია რაიმე ფიქსირებული ორიენტაციით, ხოლო $k_1, k_2, ..., k_s \in Z$. ამასთან, თუ

$$\alpha'=k_1'\sigma_1+k_2'\sigma_2+...+k_s'\sigma_s$$

მაშინ

$$\alpha+\alpha'=(k_1+k_1')\sigma_1+(k_2+k_2')\sigma_2+...+(k_s+k_s')\sigma_s$$

ხოლო თუ $\sigma_1^{/}, \sigma_2^{/}, ..., \sigma_s^{/}$ არის იგივე სიმპლექსები, რაც შესაბამისად $\sigma_1, \sigma_2, ..., \sigma_s$, ოღონდ საწინააღმდეგო ორიენტაციით, მაშინ

$$\alpha=(-k_1)\sigma_1'+(-k_2)\sigma_2'+...+(-k_s)\sigma_s'$$

თუ $q<0$ ან $q>m$ მაშინ მივიღოთ, რომ $C_q(K)=0$.

ყოველი $q\in Z$-სათვის განვსაზღვროთ ჯგუფების ჰომომორფიზმი

$$C_q(K) \xrightarrow{\Delta_q} C_{q-1}(K)$$

შემდეგნაირად:

თუ $q\leq 0$ ან $q>m$, მაშინ $\Delta_q=0$.

თუ $1\leq q\leq m$, მაშინ

$$\Delta_q(a_0, a_1,..., a_k)=\sum_{i=0}^{q} (-1)^i(a_0, a_1,...a_{i-1},a_{i+1},.., a_k) \qquad (1.9)$$

სადაც $(a_0, a_1,..., a_k)$ არის ნებისმიერი ორიენტირებული q-განზომილ. სიმპლექსი, ხოლო

$$\Delta_q(k_1 \sigma_1+k_2\sigma_2+...+k_s\sigma_s)= k_1 \Delta_q(\sigma_1)+ +k_2 \Delta_q(\sigma_2)+...+k_s \Delta_q (\sigma_s) \qquad (1.10)$$

შეიძლება შემოწმდეს, რომ (9)-(10) კორექტულადაა განსაზღვრული).

$\Delta_q(\alpha)$-ს ეწოდება α ჯაჭვის საზღვარი, ხოლო თვით Δ_q-ს ეწოდება საზღვრის ჰომომორფიზმი. ჯაჭვს ეწოდება ციკლი, თუ მისი საზღვარი უდრის 0-ს. ჯაჭვს ეწოდება საზღვარი, თუ ის რაიმე ჯაჭვის საზღვარია. შეიძლება შემოწმდეს, რომ

$$\Delta_{q-1} \Delta_q=0$$

საიდანაც გამომდინარეობს, რომ ყოველი საზღვარი ციკლიცაა, q-განზომილ. ციკლების სიმრავლე აღვნიშნოთ Z_q სიმბოლოთი, ხოლო q-განზ. საზღვრების სიმრავლე - B_q სიმბოლოთი. გვაქვს $B_q\subset Z_q$. ამასთან ცხადია, რომ B_q და Z_q $C_q(K)$-ს ქვეჯგუფებია. ფაქტორჯგუფს Z_q/B_q ეწოდება K კომპლექსის q-განზომილ. ჰომოლოგიების ჯგუფი და აღინიშნება $H_0(K)$ სიმბოლოთი. ამრიგად,

$$H_q(K)= Z_q/B_q$$

მტკიცდება , რომ ყოველი K სიმპლიციალური კომპლექსისათვის $H_0(K)$ ჯგუფი თავისუფალია და მისი რანგია Q_0%

$$H_0(K) \approx \underbrace{Z \oplus Z \oplus \ldots \oplus Z}_{Q_0-\text{ჯერ}} \tag{1.11}$$

უფრო მაღალ განზომილებებში საზოგადოდ $H_q(K)$ ჯგუფი არა არის თავისუფალი _ მასში პერიოდული ელემენტებიც შეიძლება არსებობდეს. (შესაბამისი მაგალითი იხ. [1]-ში).

აბელური სასრულად წარმოქმნილი ჯგუფებისათვის სტრუქტურული თეორემის თანახმად ყოველი $H_q(K)$ იშლება პირდაპირ ჯამად

$$H_q(K) \approx \underbrace{Z \oplus Z \oplus \ldots \oplus}_{p_q-\text{ჯერ}} Z \oplus Z/\tau_1 Z \oplus Z/\tau_2 Z \oplus \ldots \oplus Z/\tau_{s_q} Z$$

და ყოველი τ_{j+1} იყოფა τ_j-ზე. p_q-ს ეწოდება K კომპლექსის q-ური ბეტის რიცხვი, ხოლო $\tau_1, \tau_2, \ldots$ რიცხვებს _ q-ური გრეხვის კოეფიციენტები. (11)-დან ჩანს, რომ Q_0 სინამდვილეში არის 0-ანი ბეტის რიცხვი. საზოგადოდ, სხვა q-სათვის Q_q განსხვავდება q-ური ბეტის რიცხვისაგან. განვიხილოთ მაგალითად სიმპლიციალური კომპლექსი სურ. 2-ზე, რომელიც შესდგება სამი წვეროსაგან და ორი 1-განზ. სიმპლექსისაგან. ადვილი შესამოწმებელია, რომ $Q_1=2$. ვიპოვოთ 1-ლი ბეტის რიცხვი. ამისათვის აღვნიშნოთ, რომ ყოველი სიმპლიციალური კომპლექსისათვის სამართლიანია ე.წ. ეილერ-პუანკარეს ფორმულა:

$$\sum_{q=0}^{m} (-1)^q \alpha_q = \sum_{q=0}^{m} (-1)^q p_q$$

ნახ 2

სადაც m არის კომპლექსის განზომილება, α_q არის კომლექსისის q-განზომილებიანი წახნაგების რაოდენობა, p_q _ q-ური ბეტის რიცხვი. ჩვენს შემთხვევაში ფორმულა მიიღებს სახეს:

$$\alpha_0 - \alpha_1 = p_0 - p_1$$

გვაქვს $\alpha_0=3, \alpha_1=2, p_0=Q_0=1$. ამიტომ $p_1=0$.

X და Y სიმრავლეებს შორის ყოველ მიმართებას $\rho\subset X\times Y$ შევუსაბამოთ შემდეგი K_ρ სიმპლიციალური კომპლექსი: K_ρ-ს წვეროებია X-ის ელემენტები, ხოლო X' არის სიმპლექსი მაშინ და მხოლოდ მაშინ, როცა არსებობს ისეთი $y\in Y$ რომ ყოველი $x'\in X'$ -სათვის $(x';y)\in\rho$.

ყოველი y-სათვის K_ρ-ს სიმპლექსს:

$$\{x \mid (x;y)\in\rho\}$$

ვუწოდებთ y-სიმპლექსს.

ახლა დავუბრუნდეთ ისევ სისტემებს და შევთანხმდეთ, რომ სისტემის ქვეშ ჩვენ ვგულისხმობთ ოთხეულს

$$(X;Y;\rho;\pi) \tag{1.12}$$

სადაც X და Y სიმრავლეებია, $\rho\subset X\times Y$, ხოლო π ასახვაა

$$\pi\colon K_\rho \to R$$

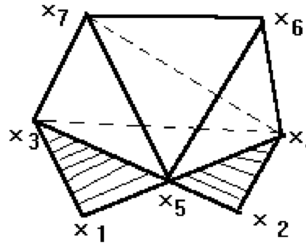
ე.ი. შესაბამისობა, რომლის დროსაც ყოველ სიმპლექსს შეესაბამება ნამდვილი რიცხვი. π–ს ეწოდება მოდელი K_ρ სიმპლიციალურ კომპლექსზე.

π-ს მოცემა შეიძლება მაგალითად $f\colon\rho\to Z$ ასახვის მეშვეობით

$$\pi[x_0;x_1;...;x_k]=\sum_{\forall j(x_j;y)\in\rho}\sum_{j=0}^{k} f(x_j;y) \tag{1.13}$$

f და π-ს აქვს შემდეგი ფიზიკური შინაარსი: $f(x;y)$ არის სისტემაში $(x;y)$ კავშირის განხორციელებაზე გაწეული დანახარჯები, $\pi(\sigma)$ არის σ-სთან ყველა კავშირების დამყარებაზე გაწეული დანახარჯები. ვნახოთ მაგალითი. ვთქვათ, X არის ქალაქის რაიონების სიმრავლე, ხოლო Y - საავადმყოფოების სიმრავლე. ρ მიმართება შემდეგია- $(x;y)\in\rho$ მაშინ და მხოლოდ მაშინ, როცა y ემსახურება x რაიონს. თუ მაგალითად ρ-ს ინციდენტობის მატრიცაა:

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$



ნახ 3

მაშინ შესაბამისი K_ρ 3-განზომილებიანი კომპლექსია ნახ.3-ზე. ვთქვათ, f ფუნქცია, რომელიც ასახავს ხარჯებს რაიონების მიერ საავადმყოფოების დაფინანსებაზე მოიცემა მატრიცით

$$(a_{ij})_{1\le i\le 6,\ 1\le j\le 7}$$

მაშინ, მაგალითად

$$\pi[x_1;x_3]=a_{11}+a_{13},\ \pi\,[x_5;x_4;x_7]=(a_{55}+a_{54}+a_{57})+(a_{65}+a_{64}+aq_{67}) \qquad (1.14).$$

სისტემისათვის X სიმრავლის ყოველ ელემენტს შევუსაბამოთ რაიმე x_i ცვლადი. K_ρ-ზე ყოველ π მოდელს შევუსაბამოთ $x_1,x_2,...,x_n$ ცვლადების შემდეგი პოლინომი:

$$\sum_{\{x_{i_1};x_{i_2};...;x_{i_k}\}\in K_\rho} \pi\left[x_{i_1};x_{i_2};...;x_{i_k}\right]x_{i_1}x_{i_2}...x_{i_k}$$

ფიქსირებული ρ-ს შემთხვევაში ეს შესაბამისობა იძლევა K_ρ-ზე მოდელების სიმრავლესა და $R[x_1;...;x_k]$-დან ისეთი პოლინომების სიმრავლეს შორის ბიექციას, რომლებიც

1) არ შეიცავენ წევრს $ax_{i_1}^{n_1}x_{i_2}^{n_2}...;x_{i_k}^{n_k}$, თუ ერთი მაინც $n_j>1$;

2) არ შეიცავენ წევრს $ax_{i_1}x_{i_2}...x_{i_k}$, თუ $\left\{x_{i_1};x_{i_2};...;x_{i_k}\right\}\notin K_\rho$.

ამრიგად, ყოველ მოდელს შეიძლება ვუყუროთ როგორც პოლინომს, და ზოგიერთ პოლინომს - როგორც მოდელს. იმის გამო, რომ პოლინომთა აღნიშნული სიმრავლე ჩაკეტილია შეკრებისა და გამოკლების ოპერაციების მიმართ, მოდელებზეც შეიძლება განვსაზღვროთ შესაბამისი ოპერაციები თუ ჩავთვლით,

რომ $x_1, x_2,..., x_n$ ცვლადები წარმოადგენენ რაიმე ისეთი R-ალგებრის ელემენტებს, რომელშიც სრულდება თანაფარდობები%

$$x_i x_j = -x_j x_i$$

ყველა $1 \le i,j \le n$-სათვის, საიდანაც კერძოდ გამომდინარეობს, რომ

$$x_i^2 = 0 \tag{1.15}$$

ყველა $1 \le i \le n$-სათვის, მაშინ შეიძლება განვსაზღვროთ ასეთი პოლინომების ნამრავლიც, რომელიც აკმაყოფილებს 1)-ს, მაგრამ საზოგადოდ არ აკმაყოფილებს 2)-ს. ამრიგად, მოდელების ნამრავლი საზოგადოდ არ გვაძლევს მოდელს.

სისტემაში ყოველი ცვლილება გამოიხატება, $(X;Y;\rho;\pi)$ ოთხეულის ახალი $(X_1;Y_1;\rho_1;\pi_1)$ ოთხეულით შეცვლაში. მაგალითად, თუ ზემოთ განხილულ სიტუაციაში ცვლილება მდგომარეობდა იმაში, რომ რემონტზე დაიხურა პირველი საავადმყოფო, მაშინ ახალი $\rho_1 \subset X \times Y_1$ მოიცემა მატრიცით

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

ცხადია, შესაბამისად შეიცვლება K_ρ და π-ც. ცვლილება მოხდება იმ შემთხვევაშიც, თუ მაგალითად, სავადმყოფოების რაოდენობა დარჩება იგივე, მაგრამ შეიცვლება მომსახურების ფასები და ა.შ.

სხვაობას $\delta\pi = \pi_1 - \pi$ ეწოდება π მოდელის ნაზრდი.

განვიხილოთ კიდევ ერთი მაგალითი. ვთქვათ, გვაქვს რეგიონი, რომელიც შესდგება N რაიონისაგან. რეგიონში მცხოვრებ მოსახლეობას აქვს ინტერესების J სიმრავლე. მივიღოთ , რომ ξ_{ij} არის i-იურ რაიონში j ინტერესის დაკმაყოფილების პარამეტრი და η_j _ მოსახლეობის j ინტერესის დაკმაყოფილების აუცილებლობის წონა. გვაქვს სისტემა

$$(J;I;\rho;\pi)$$

სადაც I და J შესაბამისად რაიონების და ინტერესების სიმრავლეებია, ხოლო

$$(j;i) \in \rho \Leftrightarrow \xi_{ij} \neq 0$$

ანუ რაიმე ინტერესი და რაიმე რაიონი იმყოფებიან ρ მიმართებაში მაშინ და მხოლოდ მაშინ, როცა აღნიშნულ რაიონში აღნიშნული ინტერესის დაკმაყოფილება შესაძლებელია. π მოდელი განვსაზღვროთ (13) ფორმულის ანალოგიურად, კერძოდ,

$$\pi[x_0;x_1;...;x_k]= \sum_{\forall i, \xi_{ij} \neq 0} \sum_{j=0}^{k} \xi_{ij} \eta_i$$

მივიღოთ შეთანხმება, რომ თუ რეგიონში მცხოვრებელს აქვს ინტერესების $J' \subset J$ სიმრავლე, ხოლო რაიმე i რაიონში კმაყოფილდება J'_ის მხოლოდ ნაწილი $J'' \neq J'$, მაშინ ასეთი მცხოვრებლისათვის რაიონი საერთოდ არ წარმოადგენს არჩევანის კანდიდატს. ამის შემდეგ, თუ ინტერესების ყოველ სიმპლექსს შევხედავთ როგორც ვინმე მაცხოვრებლის ინტერესების სიმრავლეს, π მოდელს შეგვიძლია მივცეთ შემდეგი ფიზიკური ინტერპრეტაცია: π(σ) არის რეგიონში σ-ს დაკმაყოფილების პარამეტრი. თუ კიდევ მივიღებთ შეთანხმებას, რომ ყოველი $i_1 \neq i_2$ რაიონებისათვის

$$\{j|\ \xi_{i_1 j} \neq 0\} \not\subset \{j|\ \xi_{i_2 j} \neq 0\}$$

მაშინ. i რაიონის მიერ ინტერესების დაკმაყოფილების λ პარამეტრი წარმოადგენს π მოდელის მნიშვნელობას i - სიმპლექსზე.

თავი 2. პროცესების ფენომენოლოგია და საინვესტიციო პოლიტიკა

2.1 ევოლუციური, სივრცული ეკონომიკა

თავისუფალი კონკურენციის პირობებში ეკონომიკური საქმიანობის რეგულირების ძირითად მექანიზმს წარმოადგენს ბაზარი. აქედან გამომდინარე “ბაზრის” შესწავლა არის ეკონომიკური მეცნიერების ძირითადი საგანი კაცობრიობის ისტორიის მთელ მანძილზე. ბაზრის პოლიტეკონომიკური კონცეფცია გულისხმობს ეკონომიკური აგენტების რაციონალურ ქცევას, რომელსაც მივყავართ ბაზრის წონასწორულ მდგომარეობამდე. ამ კონცეფციის რეალიზაცია ხორციელდება სპეციალური საბაზრო რეგულატორებით-ადმინისტრაციული, საგადასახადო, საბიუჯეტო, სამართებლივი და სხვა. ამ მექანიზმების დანიშნულებაა დაეხმარონ ეკონომიკურ აგენტებს ინდივიდუალური გეგმების კოორდინირებაში და ოპერატიულ საქმიანობაში. იმ მიზნით, რომ ეკონომიკური სისტემა მივიდეს წონასწორულ მდგომარეობამდე.

მაგრამ, რეალურ ეკონომიკურ პროცესებს თუ დავაკვირდებით, ზემოთ მოყვანილი ბაზისური კონცეფცია საკმარისი არაა იმ დესტაბილიზაციის, დეზინტეგრაციის, და კოორდინაცის დარღვევათა ახსნისათვის, რომლებსაც ხშირად მივყავართ წონასწორობისმიღმა განვითარებულ ეგზოტიკურ ეკონომიკურ დინამიკამდე. სადაც უკვე, ზემოთ მოყვანილი საბაზრო რეგულატორები

„უსუსურები" არიან მიიყვანოონ ეკონომიკური სისტემა წონასწორულ მდგომარეობამდე.

აქედან გამომდინარე, ცხადია, აუცილებელია ისეთი ზოგადი კონცეფციის შექმნა, რომელიც ახსნის, როგორც ეკონომიკური წონასწორობის წარმოშობის ასევე ეკონომიკური განვითარების, მისი ზრდის მექანიზმებს. ასეთ ზოგად კონცეფციას უწოდებენ „ევოლუციურ ეკონომიკას".

ეკონომიკური მეცნიერების ეს მიმართულება შედარებით ახალია და მისი თეორიული ბაზა ახლა ყალიბდება. პრობლემა განპირობებულია იმით, რომ წონასწორული მდგომარეობა და მასთან დაკავშირებული პროცესების ანალიზი და ახსნა იოლია, ვიდრე სხვადასხვა დროით მასშტაბებში მიმდინარე ურთიერთმოქმედი დინამიკური ეკონომიკური პროცესებისა, რომელთა აღმოჩენა და დაკვირვება შეუძლებელია როგორც წონასწორულ მდგომარეობაში, ასევე ეკონომიკური სისტემების ურთიერთქმედებისას მაკროსისტემასთან]. იმ ნაშრომებში, რომლებიც ევოლუციურ ეკონომიკას ეხება, ძირითადად განიხილავენ სამ მოვლენას, რომლებიც მიიჩნევიან საბაზრო წონასწორობის მადესტაბილიზებელ ფაქტორებად.

ინოვაციური ქმედება ეკონომიკური სუბიექტის ის ფაქტორია, რომელიც წარმოადგენს ეკონომიკის განვითარების მიზეზს და წყაროს. ეს ის ფაქტორია, რომელიც უწყვეტად აგზავნის იმპულსებს და უბიძგებს ეკონომიკას შეჯიბრებადობისაკენ, რესტრუქტურიზაციისაკენ. აქედან გამომდინარე ევოლუციური ეკონომიკის ძირითადი კვლევის საგანია-როგორ იბადება ინოვაცია, ანუ: პირადი შემოქმედებითი პრობლემატიკა, ახალი ტექნიკის ტექნოლოგიების გაჩენა, პროდუქციის ხარისხის და თვისებების ცვლილება, ახალი ობიექტების გაჩენა და სხვა. საბაზრო არეში ინოვაციური ქმედებების და შეჯიბრებადობის გავრცობა დიფუზიის საშუალებით, ეკონომიკრ სისტემაში წარმოქმნიან მაკროეფექტებს, რომლებიც აიძულებენ ეკონომიკურსისტემას გადავიდეს ახალ წონასწორულ მდგომარეობაში.

ეკონომიკურ სისტემას გააჩნია თვითორგანიზაციის ელემენტები. ესაა მისი უნარი დაბადოს ახალი დინამიკური რეჟიმები, როდესაც სისტემა ერთი წონასწორული მდგომარეობიდან გადადის მეორეში. ეკონომიკური სისტემის ეს თვისება რეალიზდება საბაზრო და სახელმწიფო რეგულატორების საშუალებით, რომლებიც ხელს უწყობენ იმ სიახლეებს, რომლებიც აიძულებენ სისტემას გადავიდეს ახალ სტაციონალურ მდგომარეობაში. ხელს უწყობს ამ სიახლეების დანერგვას და ამკვიდრებს სიახლეთა ახალ ფორმებს და სტრუქტურებს. ამ დროს ჩნდება ახალი დინამიკა მაკოორდინებელი და დემაკორდინებელი პროცესებით. არამდგრადი შუალედური წონასწორობებით და ფაზური გადასვლებით.

მესამე მოვლენა, რომელიც იწვევს საბაზრო წონასწორობის დესტაბილიზაციას დაკავშირებულია ეკონომიკური სისტემის ურთიერთქმედებასთან სხვა სისტემებთან,

რომლებსაც სხვა ფუნქციური დანიშნულება აქვს, როგორებიცაა: პოლიტიკური, ბუნებრივი დასხვა.

ზემოთ მოყვანილი მოვლენების, ფაქტორების საერთო და მნიშვნელოვანი განსაკუთრებულობებია:

- სივრცული განაწილება
- თითოეულ მათგანში არსებობს ისეთი შიგა პროცესები, რომლებსაც გააჩნიათ სხვადასხვა დროითი მახასიათებლები, რელაქსაციები.

ეს უკანასკნელი საშუალებას გვაძლევს კლასიკური იდეა-წონასწორული ბაზრისა, ტრანსფორმირებული იქნას ლოკალურ საბაზრო წონასწორობათა პრინციპად, რომლებიც ერთმანეთს ენაცვლებიან ძირითადი ევოლუციური ეკონომიკური პროცესის განვითარებისას.

* სივრცული ეკონომიკური სისტემის ქვეშ იგულისხმება რეგიონი, რომლის რაიონებს შორის ხდება საქონლის, პროდუქციის, ინვესტიციების ურთიერთგაცვლა.

** ინვესტიციები გაიგება, როგორც ნაკადური ცვლადი, ანუ ინვესტიციების მოცულობა დროის ერთეულში.

2.2 ქმედებათა მოტივაცია საინვესტიციო პოლიტიკის შემუშავებისას.

ეკონომიკურ თეორიაში ერთერთი ყველაზე გავრცელებული ჰიპოთეზა, რომელიც უკავშირდება ა.სმიტს, მდგომარეობს იმაში, რომ ინვესტორის, ეკონომიკური აგენტის ქმედებას განსაზღვრავს მოგების მაქსიმიზაცია. ასეთ ქმედებას უწოდებენ რაციონალურს, შესაბამის მოდელს კი რაციონალური ქმედების (ქცევის) მოდელს.

შემდეგში რაციონალური ქცევის ცნებად შემოღებული იქნა უფრო ფართო ცნება-სარგებლიანობის. ახლა უკვე, მწარმოებლისათვის სარგებლიანობა გაიგება როგორც მისი მოგება, ხოლო მომხმარებლისათვის – როგორც კმაყოფილება. ცხადია, პრაქტიკული გამოყენებისათვის (მითუმეტეს, როდესაც საქმე ეხება მაქსიმიზაციას), საინვესტიციო პოლიტიკის შემუშავებისათვის, ორივე მაჩვენებელი მოითხოვს რაოდენობრივ შეფასებას, მიუხედავად იმისა, რომ „სარგებლიანობა“ თითქოს ხარისხობრივია ხოლო მოგება „უფრო“ რაოდენობრივი.

~~სარგებლიანობის მაქს~~იმიზაცია საშუალებას იძლევა შევადაროთ ერთმანეთს ეკონომიკური ალტერნატივები. განვიხილოთ ეს პრობლემა შემდეგი მარტივი შემთხვევისათვის, ანუ როდესაც გვაქვს ალტერნატივების სასრული სიმრავლე x={$x^1, \dots, x^s$}, და თითოეული ალტერნატივა მკაფიოდაა ფორმულირებული. თითოეული ალტერნატივა ესაა n - განზომილებიანი ვექტორი x^i={$x_1^i, \dots, x_n^i$}, R^n – სივრციდან. მწარმოებლისათვის ალტენატივების სიმრავლე შეიძლება იყოს საწარმოო პროცესების ტექნოლოგიები, ხოლო კომპონენტები x_j^i, i ტექნოლოგიის მახასიათებლები.

თითოეული ალტერნატივის სარგებლიანობის რაოდენობრივ მახასიათებლებად შეიძლება ავიღოთ სარგებლიანობის ფუნქცია $u(x)$, რომელიც განსაზღვრულია X - ზე. მოხერხებულობისთვის (ზოგადობის შეუზღუდავად) ვთვლით, რომ U-ს მნიშვნელობები ნორმირებულია, ანუ მისი მნიშვნელობების არე ერთეულ ინტერვალშია $U = [0,1]$.

სარგებლიანობის ფუნქცია საშუალებას გვაძლევს, შევადაროთ ერთმანეთს ალტერნატივები: x^i ალტერნატივა „უფრო სარგებლიანია“ x^j - ალტერნატივაზე. ანუ $x^i \succ x^j$, თუ $u(x^i) > u(x^j)$ (2.1)

მკაცრი უტოლობა ნიშნავს იმას, რომ ყველა განსახილველი ალტერნატივიდან არსებობს ერთადერთი x^{io}. რომლისთვისაც სარგებლიანობის ფუნქცია მაქსიმალურია:

$$x^{io} = arg \max_{1 \le i \le s} u\left(x^i\right) \qquad (2.2)$$

ინვესტორის, საერთოდ ყველა ჩვენს ქმედებას საფუძვლად უდევს გარკვეული მოტივაცია, მიზანი. რადგანაც, ბუნებრივია, ყოველი ქმედების შედეგად ჩვენ გარკვეულ სარგებელს ველოდებით: მოგება, კმაყოფილება. ჩვენი ქცევის მოდელირებისათვის აღნიშნული საკითხი ყველაზე ნათლად ჩანს ეკონომიკური გადაწყვეტილების მიღებისას. ამიტომ ქვემოთ ამ საკითხს განვიხილავთ, ზოგადობის შეუზღუდავად, ეკონომიკური თეორიის ფარგლებში.

განვიხილოთ შემთხვევა, როდესაც ალტერნატივის მახასიათებელი პარამეტრები ($x^i \in R^n$ვექტორის პარამეტრები) ცნობილია, მაგრამ ზუსტად არ არის ცნობილი მათი მნიშვნელობები. ამ შემთხვევაში R^n სივრცეში შეიძლება გამოვყოთ V^i - სიმრავლე, რომელიც წარმოადგენს ალტერნატივების შესაძლებელ მნიშვნელობათა სიმრავლეს. ამ შემთხვევაში ჩვენ შეგვიძლია განვიხილოთ მხოლოდ ის ალტერნატივები, რომლის მახასიათებლები არიან $x^i \in V^i \subset R^n$ ვექტორები. ჩვეულებრივ V^i - სიმრავლე წარმოადგენს $n-$ რიგის პარალელეპიპედს ანუ

$$V^i = \{x^i : a_k^i \leq x_j^i \leq b_k^i, i = 1,2, \dots, n\}.$$

ანალოგიურად წინა შემთხვევისა, შემოვიტანოთ სარგებლიანობის ფუნქცია $u(x)$ განსაზღვრული $V = \bigcup_{i=1}^{S} V^i$ სიმრავლეზე, მნიშვნელობებით U- ინტერვალში. ამ შემთხვევაში ინვესტორის, ეკონომიკური აგენტის რაციონალური არჩევანი, როდესაც სარგებლიანობის ფუნქციით შესაძლებელია ალტერნატივების სელექცია (იხ. გამოსახულება (1)) განისაზღვრება ალტერნატივით

$$x^* = arg \max_X (u(x) \mid x \in V). \tag{2.3}$$

რაციუნალური ქცევის მოდელები დაფუძნებულია სარგებლიანობის ფუნქციის არსებობაზე, რომლის საშუალებითაც შეიძლება რაოდენობრივად შევაფასოთ ალტერნატივების სარგელიანობის სიდიდეები. ქვემოთ განვიხილოთ სარგებლიანობის ფუნქციის ზოგიერთი კლასები და მათი აგების მეთოდები.

2.3 სარგებლიანობის ადიტიური ფუნქციები.

განვიხილოთ x ალტერნატივების სიმრავლე $V \in R^n$, სადაც x - ვექტორის კომპონენტები $x_1, , \dots . x_n$, წარმოადგენენ ალტერნატივის მახასიათებლებს ნატურალურ მაჩვენებლებში. სარგებლიანობის ადიტიურ ფუნქციათა კლასი აღიწერება შემდეგ სახის გამოსახულებით

$$u(x) = \sum_{i=1}^{n} u_i (x_i) \tag{2.4}$$

სადაც $u_i(x_i)$ – x - ალტერნატივის ცალკეული კომპონენტების სარგებლიანობის კერძო ფუნქციებია. $u(x)$- ის ყველაზე გავრცელებული სახეა წრფივი ფუნქცია

$$u_i(x_i) = a_i x_i + b_i, \ i = 1, \dots n \tag{2.5}$$

ასეთ დროს (4) იღებს შემდეგ სახეს

$$u(x) = \sum_{i=1}^{n} (a_i x_i + b_i) \tag{2.6}$$

მაქსიმალური სარგებლიანობის ალტერნატივის არჩევის ამოცანა დადის (3)-ის წრფივი პროგრამების ამოცანამდე. მაგრამ, ამისათვის აუცილებელია წინასწარ განვსაზღვროთ კოეფიციენტები a, b, (6) გამოსახულებაში. მოვიყვანოთ ამ კოეფიციენტების განსაზღვრის შემდეგი მაგალითი.

ვთქვათ, ვაშენებთ ახალ სატრანსპორტო მაგისტრალს. ალტერნატიულ პროექტებს ვახასიათებთ: დანახარჯებით-K, რომელიც საჭიროა მაგისტრალის მშენებლობისათვის, მოგებით -P, და ერთ სულ მოსახლეზე მგზავრობის დროის შემცირებაზე - Δ. ამ შემთხვევისათვის (6) მიიღებს შემდეგ სახეს

$$u = a_1 P + a_2\, K + a_3\, \Delta$$

ამ შემთხვევაში არ ვიხილავთ კოეფიცინტ b (იხ. გან. (6)), რადგანაც ის არ მოქმედებს ალტერნატივის არჩევაზე, რომელიც მოგვცემს მაქსიმალურ სარგებლიანობას. a_1 და a_2 კოეფიციენტები შეიძლება მივიჩნიოთ 1-ის ტოლად, რადგანაც დანახარჯები და პროექტში ჩადებული მოგების ნორმა დამოკიდებულნი არიან ინვესტირების პირობებზე და პოტენციური დამკვეთის შესაძლებლობებზე. კოეფიციენტი a_3 მიზანშეწონილია დაუკავშიროთ იმ N მოქალაქეს, რომლებიც ღებულობენ სარგებელს ამ პროექტს რეალიზაციისას ანუ მივიღოთ $a_3 = c\,N$. სადაც c არის მოქალაქის ერთეულოვანი თავისუფალი დროის ფასი. აქვე უნდა შევნიშნოთ, რომ ასეთი სიტუაციები გვაქვს იშვიათად.

ხშირად იყენებენ სარგებლიანობის ფუნქციებს, რომლებთაც აქვთ არაწრფივი ხასიათი, კერძოდ მათ აქვთ ექსპონენციალური და ლოგარითმული ფუნქციების სახე:

$$u_i(x_i) = exp\{a_i x_i + b_i\} \tag{2.7}$$

$$u_i(x_i) = l_n\{a_i x_i + b_i\} \tag{2.8}$$

2.4 მულტიპლიკატური სარგებლიანობის ფუნქციები.

განვიხილოთ შემთხვევა, როდესაც ყოველი ალტერნატივა ხასიათდება მხოლოდ ორი x_1 და x_2 მაჩვენებლებით. ვთვლით, რომ ორი ალტერნატივა ტოლფასიანია

(ექვივალენტურია), როდესაც ნებისმიერი ერთ-ერთი მახასიათებლის გაუარესება კომპენსირდება მეორე ალერნატივის სხვა მახასიათებლის გაუმჯობესებით, ანუ

$$\frac{\Delta x_1}{x_1} = -\frac{\Delta x_2}{x_2} \quad (2.9)$$

სარგებლიანობის ფუნქციის ეს თვისება შეიძლება წარმოვადგინოთ შემდეგი დიფერენციალური ფორმის სახით:

$$d_u(x) = x_2 dx_1 + x_1 dx_2 \quad (2.10)$$

ამ გამოსახულებიდან ჩანს, რომ (12) ტოლობა განსაზღვრავს $u(x_1x_2)$=x_1x ფუნქციის სრულ დიფერენციალს.

ეს პრინციპი ვრცელდება ზოგად შემთხვევისათვისაც, როდესაც x ვექტორს აქვს n კომპონენტი.

$$\sum_{i=1}^{n} \frac{\Delta x_i}{x_i} = 0 \quad (2.11)$$

აქედან ვღებულობთ სარგებლიანობის ფუნქციის მულტიპლიკატორულ ფორმას

$$u(x) = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} \quad (2.12)$$

მულტიპლიკატური ფუნქციის ერთ-ერთი დადებითი განსაკუთრებულობა ისაა, რომ საუკეთესო ალტერნატივის არჩევა დამოკიდებული არ არის იმაზე, თუ რა ერთეულებში იზომება x_i - კომპონენტები.

2.5 კომპრომისული ქცევის მოდელები, საინვესტიციო პოლიტიკის გატარებისას

ეკონომიკის განვითარებას თან სდევს ეკონომიკური საქმიანობის დივერსიფიკაცია (მრავალფეროვნება). მასში მიმდინარე პროცესების დინამიკურობა და მონაწილეთა რაოდენობის ზრდა. ყოველივე ეს პრობლემებს უქმნის ინვესტორებს, ეკონომიკური აგენტებს, რათა ამ დროს მათი ქცევა იყოს რაციონალური. ეს განპირობებულია იმით, რომ გადაწყვეტილების მიღების მომენტი და ამ გადაწყვეტილების შედეგად მიღებული მოგების დრო საკმაოდ დაშორებულები არიან ერთმანეთისაგან. ამ დროს ყოველთვის არსებებს რისკი იმისა, რომ დაგეგმილი მოგება მიუღწეველი იქნება. რისკი, შეიძლება ითქვას, ესაა კრებსითი ცნება, რადგანაც ის ერთდროულად მოიცავს, როგორც რაოდენობრივ ასევე ხარისხობრივ, თვისობრივ მაჩვენებლებს- რომელთა გაზომვა შეუძლებელია (მაგალითად ფორს – მაჟორული სიტუაცია). თუ მწარმოებლის რეაქცია მოგებაზე

ცალსახაა რაც შეიძლება „მეტი“, მისი რეაქცია რისკზე არაერთგვაროვანია. ის დამოკიდებულია მწარმოებლის თვისებებზე, ზოგადად -ეკონომიკურ აგენტზე.

შეიძლება ითქვას, რომ გადაწყვეტილების ალტერნატივები ხასიათდებიან ორი „სარგებლიანობის ფუნქციით“: მოგება და რისკი. სარგებლიანობის ფუნქციის არაერთმნიშვნელიანობა წარმოადგენს ეკონომიკური აგენტის პრინციპიალურ თვისებას.

კვლავ განვიხილოთ x ალტერნატივების სიმრავლე $V \in R^h$.

დავუშვათ, ალტერნტივად განიხილება ერთ-ერთი საცხოვრებელი ბინის მშენებლობა. ხოლო x ვექტორის კომპონენტები ამ პროექტის პარამეტრებია, რომლებიც რაოდენობრივად ახასიათებენ x ალტერნატივებს $Q_1,(x),\dots,Q_n(x)$, რომლებიც წარმოადგენენ $Q(x)$ ვექტორული კრიტერიუმის კომპონენტებს. ცხადია ისეთი გადაწყვეტილების მიღება, რომლის დროსაც ყველა კრიტერიუმი აღწევს თავის მაქსიმალურ (ექსტრემალურ) მნიშვნელობებს, თითქმის შეუძლებელია. რადგანაც კრიტერიუმები, როგორც წესი წარმოადგენენ ურთიერთკონკურენტულ სიდიდეებს. მაგალითად ჩვენს შემთხვევაში კრიტერიუმი Q_1 - მოგება, კრიტერიუმი Q_2 - რისკი. ისეთი ალტერნატივის მოძებნა, რომლისთვისაც მოგება მაქსიმალურია, ხოლო რისკი მინიმალური თითქმის შეუძლებელია. ბუნება ისეა მოწყობილი, რომ მაქსიმალურ მოგებას შეესაბამება მაქსიმალური რისკი. აქედან გამომდინარე უნდა ვიფიქროთ <u>კომპრომისულ</u> ალტერნატივაზე: ნაკლები რისკი, მოგება არამაქსიმალური.

კომპრომისის ცნება უფრო მეტად არამკაფიოა, ვიდრე სარგებლიანობის. რადგანაც ჩვენ გვინდა ეკონომიკური აგენტების კომპრომისული ქცევის მოდელირება, ამისათვის აუცილებელია ფორმალურად აღვწეროთ განსახილველი კომპრომისების შინაარსი.

<u>**წონითი კომპრომისი:**</u> განვიხილოთ კრიტერიუმების წრფივი კომბინაცია:

$$F(Q(x)) = \sum_{k=1}^{m} Q_k(x)\lambda k \tag{2.13}$$

აქ $\lambda_\lambda, \lambda_2, \dots \lambda_m$ - წონითი კოეფიციენტები ნორმირებულია:

$$\sum_{k=1}^{m} \lambda_k = 1. \tag{2.14}$$

საუკეთესო ალტერნატივა x^* განისაზღვრება გამოსახულებით

$$x^* = arg\max_x \big(F\big(Q(x)\big) | x \in V\big) \tag{2.15}$$

ამ მოდელში იგულისხმება, რომ ვექტორული კრიტერიუმის კომპონენტების ფასეულობა განისაზღვრება წონითი კოეფიციენტებით $\boldsymbol{\lambda}$, რომლებიც წინასწარ მოიცემა. კოეფიციენტების განსაზღვრის მრავალი რეკომენდაცია არსებობს, მაგრამ ყველა მეთოდის ამოსავალია ექსპერტების შეფასებანი.

<u>პირობითი კომპრომისი;</u> ამ შემთხვევაში ვექტორული კრიტერიუმის კომპონენტებიდან ირჩევენ ყველაზე მნიშვნელოვანს $Q_{ko}(x)$, ხოლო დანარჩენებისათვის შემოაქვთ ინტერვალური შეზღუდვები $[a_k b_k]$, $k = 1, \dots, m$; $k \neq k_o$. ამ სემტხვევაში, პირობითი კომპრომისის შესაბამის ალტერნატივას x^* აქვს შესმდეგი სახე

$$x^* = arg\max_x (Q_{k0}(x) | a_k \leq Q_k(x) \leq b_k \quad k = 1,, \dots, m \quad k \neq k_o \quad), \tag{2.16}$$

<u>პარეტო-ოპტიმალური, საუკეთესო კომპრომისი:</u> შეზღუდული რესურსების პირობებში შეუძლებელია მოსახლეობის ყველა ფენის მდგომარეობის ერთდროულად გაუმჯობესება. ამას უწოდებენ პარეტო – ოპტმალური კომპრომისს, ანუ არაგაუმჯობესებად კომპრომისს ამ პრინციპის აღმომჩენის, პარეტოს საპატივცემულოდ.

აქვე უნდა შევნიშნოთ, რომ ზემოთ განხილული კომპრომისული ალტერნატივების არჩევის მოდელები ეყრდნობოდა ვექტორული კრიტერიუმის გარდაქმნას სკალარულში. მაგრამ არცერთი ვერ ასახავს ეკონომიკური ალტერნატივის „სარგებლიანობის" ვექტორული შეფასების არსს. ამის საილუსტრაციოდ განვიხილოთ ორ კომპონენტიანი ვექტორული კრიტერიუმი $Q(x) = \{Q,(x), Q,(x)\}$. $Q_1(x) \geq 0$, $Q_2(x) \geq 0$. და სასურველია მათი Q_1 და Q_2 მაქსიმიზაცია ალტერნატივათა დასახასიათებლად ავირჩიოთ აგრეთვე ორი მაჩვენებელი, $x = \{x_1, x_2\}$. ეს ყოველივე მხოლოდ იმისათვის, რომ შეგვეძლოს ყოველივეს გრაფიკულად ჩვენება.

ნახ. 1-ზე ნაჩვენებია ალტერნატივების დასაშვები სიმრავლე V. თითოეული ალტერნატივას x, V - სიმრავლედან შეესაბამება ვექტორული კრიტერიუმის კომპონენტების გარკვეული მნიშვნელობები $Q_1(x)$, $Q_2(x)$ ანუ, ვექტორული

კრიტერიუმის მიშვნელობათა სივრცეში ნახ.2.1ბ თუ ჩვენ მოვახდენთ ყველა ალტერნატივების განხილვას V - სიმრავლიდან მივიღებთ ვექტორული კრიტერიუმის სივრცეში P სიმრავლეს, საზღვრით P. დავუშვათ მას აქვს ნახ. 2.ბ გამოსახული კონფიგურაციის სახე. განვიხილოთ ორი უბანი საზღვარზე. $P{:}\,AB$ და BC.

V - სიმრავლედან ავირჩიოთ x^1 ალტერნატივა, ისეთი, რომ მას შეესაბამებოდეს წერტილი P სიმრავლის შიგა არეში. მაგალითად წერტილი $Q^1 = Q(x^1)$. რადგანაც Q^1 წერტილი არის P სიმრავლის შიგა არეში და გვინდა ვექტორული კრიტერიუმის კომპონენტების მაქსიმიზირება, მაშინ, როგორც ვხედავთ არსებობს სხვა ალტერნატივა, მაგალითად x^2, რომლის შესაბამისი შეფასებები ვექტორული კრიტერიუმის $Q(x)$ - ის მიხედვით აღემატება Q^1. ნახ1 – ნახაზიდან ჩანს, რომ არსებობს ალტერნატივების სიმრავლე V_1, რომელთა შეფასებები აღემატება ალტერნატივა x^1 - მნიშვნელობას. ეს შეფასებები მდებარეობენ იმ არეში, რომელიც შემოსაზღვრულია Q^1 წვეროთი და იმ წრფეებით, რომლებიც ერთმანეთთან ქმნიან მართ კუთხეს. ცხადია AB მონაკვეთზე შეიძლება იყოს x^1 ალტერნატივის უკეთესი შეფასების ალტერნატივები.

ახლა განვიხილოთ V - სიმრავლეზე ის ალტერნატივები, რომლის შეფასებები მდებარეობენ BC მონაკვეთზე. მაგალითად x^3 და x^4, რომლის შესაბამისი წერტილებია $Q^3 = Q(x^3)$, $Q^4 = Q(x^4)$. იმ ალტერნატივების ვექტორული შეფასებები, რომლებიც ხვდებიან BC

მონაკვეთზე, ისეთებია, რომ მათთვის შეუძლებელია ერთდროულად ვექტორული კრიტერიუმების ორვე კომპონენტის მნიშვნელობათა ზრდა: თუ იზრდება ერთი კომპონენტი, მაშინ მცირდება მეორე. ასსეტ ალტერნატივებს უწოდებენ პარეტო – ოპტიმალურებს. მათი ქვესიმრავლე აღვნიშნოთ $V_p \subset V$. შესაბამისად ასეთი ალტერნატივების ვექტორული შეფასებები ეკუთვნიან P_p ქვესიმრავლეს, პარეტო – ოპტიმალის შეფასებებს. ჩვენს შემთხვევაში P_p – ესაა BC მრუდი, რომელიც ეკუთვნის P საზღვარს.

როგორვ ვხედავთ, ალტერნატივების ხარისხიანობის შეფასებისას ვექტორული კრიტერიუმის გამოყენებას კომპრომისი დაიყვანება არაგაუმჯობესებადი არჩევამდე ალტერნატივაში, ყველა კომპონენტების მიხედვით, მაგრამ ასეთი ალტერნატივები ბევრია, V_p სიმრავლე. ყველა ალტერნატივები ამ სიმრავლიდან ტოლფასოვანები არიან. ამისათვის მათი სელექციისათვის აუცილებელია დამატებითი ინფორმაციის მოპოვება.

ახლა განვიხილოთ ზოგადი ამოცანა არაგაუმჯობესებადი კომპრომისის განსაზღვრისათვის, როდესაც ალტერნატივათა ხარისხი ფასდება კრიტერიუმის n - კომპონენტიანი ვექტორის მიხედვით.

$$Q(x) = \{Q_1(x), \ldots, Q_1(x)\}.$$

x^* - ალტერნატივა წარმოადგენს არაგაუმჯობესებადი კომპრომისის ალტერნატივის, თუ ყველა დანარჩენი ალტერნატივებისათვის - $x \in V$, შესრულებული იქნება შემდეგი პირობა: ან $Q_i(x) \leq Q_i(x^*)$, i=1,,..., n, ან მოიძებნება ისეთი io ინდექსი, ისეთი, რომ $Q_{io}(x^*) > Q_{io}(x)$.

ცხადია, ამ განსაზღვრებას შეესაბამება ალტერნატივების მთელი სიმრავლე. ამისათვის პარეტო – ოპტომალური შეფასებებს, პარეტოს სიმრავლის აგებისათვის გამოიყენებენ დამხმარე ოპტიმიზაციის მეთოდებს [6].

თავი. 3. საბაზრო წონასწორობა, მდგრადობა სივრცულ ეკონომიკურ პროცესებში

საბაზრო წონასწორობის პრობლემების კვლევისას მომხმარებელი და მწარმოებელი განიხილებიან როგორც კონკურენტული ეკონომიკური აგენტები. კონკურენტული ბაზრის კლასიკური მოდელი ხასიათდება შემდეგი ნიშნებით:

1. ყველა ფირმა აწარმოებს ერთგვაროვანი მოხმარების საგნებს. გამყიდვლის თვალით ყველა მომხმარებელი ერთგვაროვანია, ანუ არ არსებობს პრივილეგირებული მომხმარებელი.
2. არსებობს მრავალი მწარმოებელი და მომხმარებელი, რაც გამორიცხავს მათ უშუალო კონტაქტებს. მათი ურთიერთქმედება ხორციელდება მხოლოდ ფასის მეშვეობით. მომხმარებლები განსაზღვრავენ შესყიდული საქონლის რაოდენობას, ხოლო გამყიდველები გაყიდული საქონლის რაოდენობას მოცემული ფასის დროს. ამასთანავე ვგულისხმობთ, რომ შესყიდვების და გაყიდვების რაოდენობა არ მოქმედებენ ფასზე.
3. მწარმოებლებს და მომხმარებლებს აქვთ სრული ინფორმაცია ფასზე და მიმდინარე შემოთავაზებაზე. ისინი ყველა საშუალებით ცდილობენ თავისი შემოსავლების ან სარგებლიანობისას გაზრდას შესყიდული საქონლის შესყიდვით.

მესამე ნიშანი ფაქტიურად აფიქსირებს მწარმოებლების და მომხმარებლების ეკონომიკური ქცევის რაციონალურობას, რომელსაც მივყავართ წონასწორულ ფასამდე.

თავდაპირველად განვიხილოთ K მომხმარებლის ქცევა. დავუშვათ, ბაზარზე გვაქვს S რაოდენობის მოხმარების საქონელი. თითოეულის ფასია $P_1, P_2, \ldots, P_s$.

თითოეულ k მომხმარებელს $k = 1, 2, \ldots, K$ აქვს B_k - ბიუჯეტი, რომლითაც შეუძლიათ შეიძინონ შესაბამისად $q_1^k, \ldots, q_S^k$ ერთეულის საქონელი. ამასთანავე ვიღებთ, რომ ბიუჯეტი იხარჯება მთლინად, ანუ

$$B_k - \sum_{i=L}^{S} P_i\, q_i^k = 0, \qquad k = 1, \ldots, K \tag{3.1}$$

რაციონალური ქცევის ჰიპოთეზის მიხედვით მომხმარებლის სარგებლიანობის ფუნქციაა

$$u_k\left(q_1^k, \ldots, q_S^k\right) > 0, \quad k = 1, 2, \ldots, K \tag{3.2}$$

რომელიც წარმოადგენს სარგებლიანობის ზომას.

თითოეული მომხმარებელი იძენს იმ რაოდენობის საქონელს, რომლის დროს მისი სარგებლიანობის ფუნქცია აღწევს მაქსიმუმს, მუდმივი ბიუჯეტის შემთხვევაში. ანუ

$$u_k\left(q_1^k, \ldots, q_S^k\right) \to \max_q \tag{3.3}$$

$$\sum_{i=L}^{S} P_i\, q_i^k = B_k, \quad k = 1, 2, \ldots, K$$

რადგანაც მომხმარებლები ბაზარზე მოქმედებენ ერთმანეთისგან დამოუკიდებლად, ჩვენ გვაქვს K დამოუკიდებელი ამოცანა პირობით ექსტრემუმზე. მაქსიმუმის აუცილებელი პირობები განისაზღვრება ლაგრანჟის ფუნქციის სტაციონალურობით ყოველი მომხმარებლის მიმართ

$$L_k = u_k\left(q_1^k, \ldots, q_S^k\right) + \lambda_k\left(\sum_{i=1}^{S} B_k - P_i\, q_i^k\right) \tag{3.4}$$

ვღებულობთ შემდეგ განტოლებათა სისტემას

$$\frac{\partial L_k}{\partial q_i^k} = \frac{\partial u_k}{\partial q_i^k} - \lambda_k P_i = 0, \qquad i = 1, 2, \ldots, S$$

$$\frac{\partial L_k}{\partial \lambda} B_k - \sum_{i=L}^{S} P_i\, q_i^k = 0 \tag{3.5}$$

ეს სისტემა განსაზღვრავს ოპტიმალურ “კალათას”, რომლის დროსაც მომხმრებლის სარგებლიანობა მაქსიმუმია, ფიქსირებული ფასისა და ბიუჯეტის შემთხვევისათვის.

$$\hat{q}_i^k = fk_i(p, B_k); \quad k = 1, \dots, K; \quad i = 1, \dots, S \tag{3.6}$$

ახლა თუ შევაჯამებთ ინდივიდუალურ მომხმარებლების მოთხოვნებს, მივიღებთ ჯამურ მოთხოვნას თითოეულ საქონელზე

$$\hat{q}_i = \sum_{k=1}^{K} \hat{q}_i^k = F_i(p, B_k) \tag{3.7}$$

ახლა განვიხილოთ N მწარმოებლის ქცევა. მიუხედავად იმისა, რომ მწარმოებლის ქცევის ფორმალური ანალიზი, თითქმის იგივეა, როგორიც მომხმარებლების, მაინც არსებობს გარკვეული განსხვავებანი მათ შორის.

მწარმოებელმა შეიძლება აწარმოოს სხვადასხვა სახის პროდუქცია და გააჩნია უფრო მეტი საშულება წარმოების ოპტიმიზაციისათვის ვიდრე მომხმარებელმა გაუკეთოს ოპტიმიზაცია თავის ბიუჯეტს. მაგალითად, მას შეუძლია დაწიოს პროდუქციის თვითღირებულება, მოგების მაქსიმიზაციისათვის.

განვიხილოთ N მწარმოებელი, რომლებიც S საქონლის წარმოებისათვის გამოიყენებენ $x = \{x_1, \dots, x_m\}$ პროდუქტს. საწყისი პროდუქტის (ნედლეულის) გარდაქმნა გასაყიდ საქონლად აღიწერება მწარმოებლის ე.წ. ტექნოლოგიური ფუნქციით $\varphi^n(q, x)$. ამ ფუნქციის დახმარებით ჩაიწერება გარკვეული ბალანსი - შეზღუდვები, რომლებიც აღწერენ დასაშვები კავშირების არეს ნედლეულსა და წარმოებულ პროდუქციას შორის.

$$\varphi^n(q, x) = 0, \qquad n = 1, \dots, N \tag{3.8}$$

წარმოების ოპტიმიზაციის პროცედურა ორ ეტაპიანია. პირველ ეტაპზე განისაზღვრება “ნედლეულის” ოპტიმალური ვექტორი x^*, რომლის დროსაც წარმოებული “საქონლის ” მინიმლური ხარჯის ვექტორი q^* მინიმალურია. თუ საწყისი ნედლეულის ფასებია $c_1, \dots, c_m$, მაშინ ნედლეულის ფასია

$$C = \sum_{i=1}^{m} c_i x_i \tag{3.9}$$

პირველ ეტაპზე იხსნება შემდეგი ამოცანა:

$$C(c, x) = \sum_{i=1}^{m} c_i x_i \to \min_x, \quad \varphi^n(q, x) = 0, \tag{3.10}$$

მისი ამოხსნისათვის შემოგვაქვს ლაგრნჟის ფუნქცია

$$L^n(x, c, q, \lambda) = C(c, x) - \mu_n \varphi^n(q, x) \tag{3.11}$$

აქ n - მწარმოებლის რიგითი ნომერია.

აქ, ცვლადები c, q განიხილებიან როგორც მუდმივი სიდიდეები. ამიტომ ლაგრანჟის ფუნქციის სტაციონალურობის პირობები x და μ_n მიხედვით განისაზღვრება შემდეგი გამოსახულებებით

$$c_i + \mu_n \frac{\partial \varphi^n(q,x)}{\partial x_i} = 0. \qquad i = 1, 2, \dots, m$$

$$\varphi^n(q,x) = 0 \tag{3.12}$$

ეს განტოლებები განსაზღვრავენ x^* ნედლეულის კომპონენტების მოცულობებს q ოდენობის საქონლის მოცულობის და c ფასების დროს ანუ

$$x^* = g^n(q,c) \tag{3.13}$$

საბოლოოდ n - რიგის მწარმოებლის ხარჯვის ფუნქციას აქვს სახე

$$C^n = \sum_{i=1}^{m} c_i g_i{}^n(q,c) = C^n(q,c) \tag{3.14}$$

მეორე ეტაპზე, იგულისხმება, რომ მწარმოებელი ირჩევს თავისი წარმოების ვექტორს- q ისე, რომ მაქსიმიზაცია გაუკეთოს თავის მოგებას. მოგება G განისაზღვრება როგორც სხვაობა: გაყიდვებიდან შემოსავალსა და ყველა ხარჯს შორის:

$$G^n(q,c,p) = \sum_{i=1}^{S} p_i q_i - C^n(q,c) > 0 \tag{3.15}$$

აქ p - გაყიდვების ფასია.

თუ არ არსებობს არავითარი შეზღუდვები მოგების სიდიდეზე ან ნორმაზე, მაშინ მოგების მაქსიმუმის პირობები განისაზღვრება შემდეგი განტოლებათა სისტემით:

$$p_i - \frac{\partial C^n(q,c)}{\partial q_i} = 0 \tag{3.16}$$

$$-\frac{\partial^2 C^n(q,c)}{\partial q_i, \partial q_j} < 0 \qquad i,j = 1, \dots, S$$

წარმოებულები $\frac{\partial C^n(q,c)}{\partial q_i}$, თავისი შინაარსით წარმოადგენენ i-ური საქონლის წარმოების ზღვრულ ფასებს. მაშინ პირველი პირობიდან გამომდინარეობს, რომ გაყიდვების ფასი უნდა უდრიდეს i-ური საქონლის ზღვრულ ფასს.

3.2 ინვესტორების, ეკონომიკური აგენტების ინოვაციური ქმედებები

ინვესტორების ეკონომიკური აგენტების ინოვაციური საქმიანობა ხშირად მიზეზია წონასწორული ბაზრის შერყევისა და წონასწორობის მდგომარეობიდან გადახრისა. საბაზრო ფასების ბალანსირების პროცესი ხორციელდება საკმაოდ სწრაფად ინოვაციური და სხვა ეკონომიკურ პროცესებთან შედარებით. შემოვიტანოთ "სწრაფი" დროის სკალა ε, და "ნელი" დროის სკალა τ.

განვიხილოთ ფასწარმოქმნის უმარტივესი მოდელი ერთგვროვანი ეკონომიკისათვის. ამ მოდელის არსი ისაა, რომ საბაზრო ფასის $\varphi(\varepsilon)$ ფარდობითი ცვლილება პროპორციულია ჭარბი მოთხოვნის ფარდობითი ცვლილებისა. ამასთანავე ვთვლით, რომ მომხმარებლის ბიუჯეტი B, ნედლეულის შესყიდვის ფასი C და სხვა ის ფაქტები, რომლებიც მოქმედებენ ჭარბ მოთხოვნზე, მუდმივი რჩებიან საბაზრო ფასის დროს რელაქსაციის განმავლობაში, მაგრამ შეიძლება შეიცვალონ "ნელი" დროის სკალის მიხედვით. აქედან გამომდინარე ჭარბი მოთხოვნის ფუნქცია დამოკიდებულია "ნელ" დროზე τ, ანუ $\Delta = \Delta(p(\varepsilon), \tau)$.

ერთპროდუქტიანი ეკონომიკისათვის ვღებულობთ შემდეგ დიფერენცილურ განტოლებას საბაზრო ფასისათვის, "სწრაფი" დროის სკალაში ε.

$$\frac{dp(\varepsilon)}{d\varepsilon} = k\frac{F(p(\varepsilon), \tau) - Y(p(\varepsilon), \tau)}{Y(p(\varepsilon), \tau)} \tag{3.22}$$

სადაც k პარამეტრია, რომლის განზომილებაა 1/დროის ერთეული;

$F(p(\varepsilon), \tau)$ - მოთხოვნაა ნატურალურ ერთეულებში;

$Y(p(\varepsilon), \tau)$ - მიწოდება ნატურალურ ერთეულებში.

რადგანაც τ - დრო წარმოადგენს (3.22) განტოლების მარჯვენა ნაწილში, მიზანშეწონილია არა "წონასწორულ" ფასზე არამედ კვაზიწონასწორულ ფასზე.

(3.22) განტოლებიდან გამომდინარეობს, რომ კვაზიწონასწორული ფასი არის $P^*(\tau)$, რომელიც წარმოადგენს შემდეგი განტოლების ამონახსნს:

$$\Delta(p, \tau) = 0 \tag{3.23}$$

ვაჩვენოთ ინოვაციის როლი საბაზრო ფასის განსაზღვრაში.

განვიხილოთ გამრტივებული სიტუაცია: საქონელს აწარმოებს ეკონომიკა, რომელიც მოიხმარს მხოლოდ ერთ რესურსს - ერთგვროვან სამუშაო ძალას, მაგრამ იყენებს სხვადასხვა ტექნოლოგიებს.

ტექნოლოგიების რაოდენობრივ მახასიათებლად ავიღოთ $\lambda(\tau)$ დანახარჯები მუშახელზე, რომელიც საჭიროა ერთეულოვანი პროდუქციის გამოსაშვებად.

დავუშვათ, გამოიყენება სხვადასხვა ტექნოლოგიები

$$\lambda_{min}(\tau\ ) \leq \lambda(\tau\ ) \leq \lambda_{max}(\tau\ ) > 0$$

$\lambda_{min}(\tau\ )$ - ესაა საუკეთესო ტექნოლოგია τ დროის შემთხვევაში.

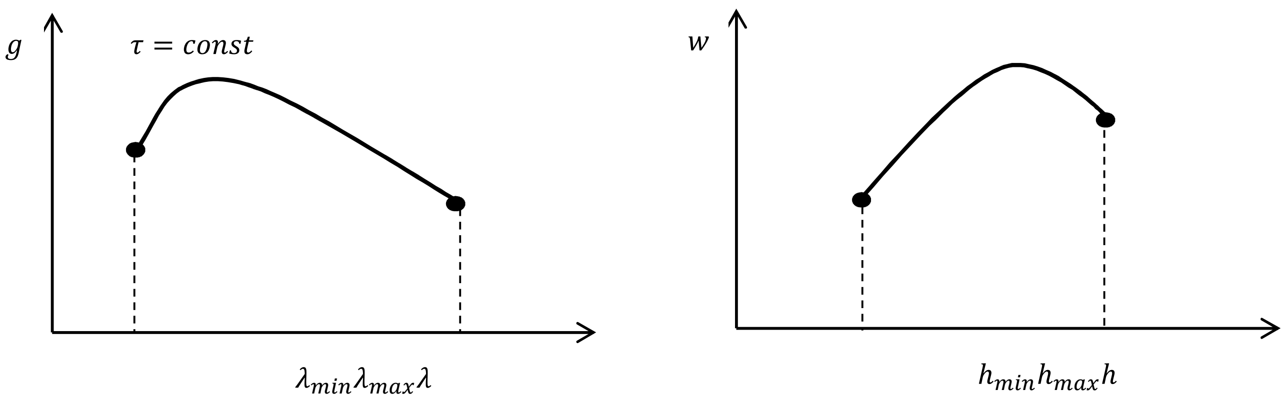
შემოვიღოთ ტექნოლოგიური დონის ცნება შემდეგი სახით:

$$h(\tau) = \frac{1}{\lambda(\tau)} \tag{3.24}$$

შესაბამისად მაქსიმლური დონე

$$h_{max}(\tau) = \frac{1}{\lambda_{min}(\tau)} \tag{3.25}$$

ეკონომიკის მდგომარეობა დროის τ მომენტისათვის დავახასიათოთ წარმოების სიმძლავრეებით ტექნოლოგიების გააწილების მიხედვით $g(\lambda(\tau),\tau)$ ან ტექნოლოგიების დონეთა მიხედვით $w(h(\tau),\tau)$. ამ განაწილებათა ხარისხობრივ მაჩვენებლებს აქვთ შემდეგი სახე (ნახ. 3.1).



ა) ბ)

ნახ. 3.1

ეკონომიკის ჯამური საწარმოო სიმძლავრე

$$M(\tau) = \int_{\lambda\ \ (\tau)}^{\lambda_{max}(\tau)} g(\lambda,\tau)dx = \int_{h_{min}(\tau)}^{h_{max}(\tau)} w(h,\tau)dh \tag{3.26}$$

g და w განაწილებები წარმოვადგინოთ შემდეგი სახით:

$$g(\lambda,\tau) = M(\tau)\psi(\lambda,\tau)$$
$$w(h,\tau) = M(\tau)\varphi(h,\tau) \quad (3.27)$$

აქ: $\psi(\lambda,\tau)$ - ტექნოლოგიური სტრუქტურის ფუნქცია;

$\varphi(h,\tau)$ - ტექნოლოგიური ეფექტურობის ფუნქციაა.

ამ მახასიათებლებს შორის არსებობსშემდეგი სახის თანაფარდობა:

$$\varphi(h,\tau) = \psi\left(\frac{1}{h},\tau\right), \psi(\lambda,\tau) = \varphi\left(\frac{1}{\lambda},\tau\right) \quad (3.28)$$

განვიხილოთ ეკონომიკური მდგომარეობა, სადაც

$$\psi(\lambda,\tau) = \frac{\lambda_{min}(\tau)}{\lambda^2}, \quad w(h,\tau) = \frac{1}{h_{max}(\tau)}h^2 \quad (3.29)$$

რაც ნიშნავს იმას, რომ ტექნოლოგიური მაჩვენებლები იცლება "საუკეთესოთი" (λ_{min}) ან მაქსიმალური ტექნოლოგიური დანართი ($h_{max}(\tau)$).

განვიხილოთ დროითი ინტერვალი $\tau \geq t \geq 0$ და ამ ინტერვალზე ტექნოლოგიური დონის $h_{max}(\tau)$ – ცვლილება. ეს ცვლილებები შეიძლება განხორციელდეს ორი პროცესის ზემოქმედებით. ესაა: ცვეთა, რაც იწვევს $h_{max}(\tau)$ - შემცირებას. მეორე ესაა ტექნოლოგიური პარკის განახლება, რაც იწვევს $h_{max}(\tau)$ - ზრდას.

υ - აღვნიშნოთ ტექნოლოგიური დანადგარების ცვეთის ინტენსიობა.

$I_T(\tau)$ - ინვესტიციები (ნატურალურ გამოსახულებაში)

μ - ინვესტიციების გამოყენების ეფექტურობაა.

ქვემოთ განვიხილოთ საინვესტიციო პოლიტიკის ზოგიერთი სტრატეგიები.

3.3 გარე ინვესტიციები

_დავუშვათ, წარმოებაში ხორციელდება ინვესტიციები მხოლოდ გარედან. ზემოთ მიღებული პირობებისათვის ტექნოლოგიური დონის მაქსიმალური მნიშვნელობის ზრდისათვის $\tau \geq t$-თვის გვექნება

$$\frac{dh_{max}(\tau)}{d\tau} = -\upsilon h_{max}(\tau) + \mu I_T(\tau), \quad h^0{}_{max} = h_{max}(t) \quad (3.30)$$

თუ დროის $\tau \geq t$ ინტერვალში $I_T(t)$ მუდმივაა. ამ განტოლების ამონახსნია

$$h_{max}(\tau)=\frac{\mu I_T(t)}{\upsilon}(1-exp\{-\upsilon(\tau-t)\})+h_{max}(t)exp\{-\upsilon(\tau-t)\} \quad (3.31)$$

“საუკეთესო” ტექნოლოგია განისაზღვრება გამოსახულებით

$$\lambda_{min}(\tau)=\left(\frac{\mu I_T(t)}{\upsilon}(1-exp\{-\upsilon(\tau-t)\})+h_{max}(t)exp\{-\upsilon(\tau-t)\}\right)^{-1} \quad (3.32)$$

საბაზრო ეკონომიკის პირობებში მხოლოდ ის წარმოებები მუშაობენ, რომლებიც რენტბელურები არიან. ყველაზე მარტივ შემთხვევაში, ანუ თუ გავითვალისწინებთ მხოლოდ “ადამიანურ” დანახარჯებს, რენტაბელობის პირობა ჩაიწერება სახეში

$$P(\tau)-\lambda(\tau)S(\tau)\geq 0 \quad (3.33)$$

აქ: $S(\tau)$ - ხელფასებია.

აქედან გამომდინარეობს, რომ დროის τ მომენტისათვის მხოლოდ ის წარმოება იმუშავებს, რომლისთვისაც “ადამიანური” დანახარჯები პატარაა ვიდრე

$$\lambda^*(\tau)=\frac{P(\tau)}{S(\tau)} \quad (3.34)$$

ტექნოლოგიურ დონეს

$$h^*(\tau)=\frac{1}{\lambda^*(\tau)} \quad (3.35)$$

უწოდებენ <u>რენტაბელობის დონეს.</u>

$\lambda^*(t)$ საშუალებას გვძლევს განვსაზღვროთ საქონლის მოთხოვნა ბაზარზე

$$Y(P,\tau)=M(\tau)\int_{\lambda_{min}(\tau)}^{\lambda^*(\tau)}\psi(\lambda,\tau)d\tau \quad (3.36)$$

და იმ ადამიანური რესურსების სიდიდე, რომელიც საჭიროა მისი წარმოებისათვის.

$$R^E(P,\tau)=M(\tau)\int_{\lambda_{min}(\tau)}^{\lambda^*(\tau)}\lambda\psi(\lambda,\tau)d\tau \quad (3.37)$$

თუ ამ გამოსახულებებში შევიტანთ (3.29), გვექნება

$$Y(P,\tau)=M(\tau)\left(1-\frac{\lambda_{min}(\tau)}{\lambda^*(\tau)}\right) \quad (3.38)$$

$$R^E(P,\tau) = M(\tau)\lambda_{min}(\tau)ln\frac{\lambda^*(\tau)}{\lambda_{min}(\tau)} \tag{3.39}$$

ეს ტოლობები განსაზღვრავენ კავშირის ხასიათს წარმოების საშუალებებსა (წარმოების სიმძლავრესთნ $M(\tau)$), საჭირო მუშა (ადამიანურ) რესურსებს $R^E(P,\tau)$ და გამოშვებული პროდუქციის რაოდენობას შორის $Y(P,\tau)$.

განვითარებადი საბაზრო ეკონომიკის დროს მთლიანი მოთხოვნის ძირითადი ნაწილი განისაზღვრება საჭირო მუშა (ადამიანური) რესურსის მიხედვით. მივიღოთ, რომ მუშა რესურსების საჭირო რაოდენობა სრულიად დაკმაყოფილებულია, ხოლო ყიდვითუნარიანობა ხასიათდება ფარდობით ხელფასსა და საქონლის ფასს შორის, ზოგადი მოთხოვნა პირველი მიახლოებით შეგვიძლია წარმოვადგინოთ შემდეგი სახით

$$F(P,\tau) = \varepsilon R^E(P,\tau)\frac{S(\tau)}{P(\tau)} \tag{3.40}$$

აქ ε - მოთხოვნის ხვედრითი წილის კოეფიციენტია, რომელიც იზომება საქონლის რაოდენობით (ნატურალურ ერთეულებში), რომელიც მოდის ერთ მოსამსახურეზე.

თუ ჩვენ (3.38) - (3.40) და (3.34) ჩავსვამთ (3.23), რომელიც განსაზღვრავს კვაზისტაციონალურ ფასს, როდესაც $\tau = const$, გვექნება:

$$\varepsilon\lambda_{min}(\tau)ln\left(\frac{P(\tau)}{S(\tau)\lambda_{min}(\tau)}\right)\frac{S(\tau)}{P(\tau)} - \left(1 - \frac{\lambda_{min}(\tau)S(\tau)}{P(\tau)}\right) = 0 \tag{3.41}$$

შემოვითანოთ აღნიშვნა

$$x = \frac{\lambda_{min}(\tau)S(\tau)}{P(\tau)} \tag{3.42}$$

მაშინ (3.41) შეიძლება წარმოვადგინოთ შემდეგი სახით:

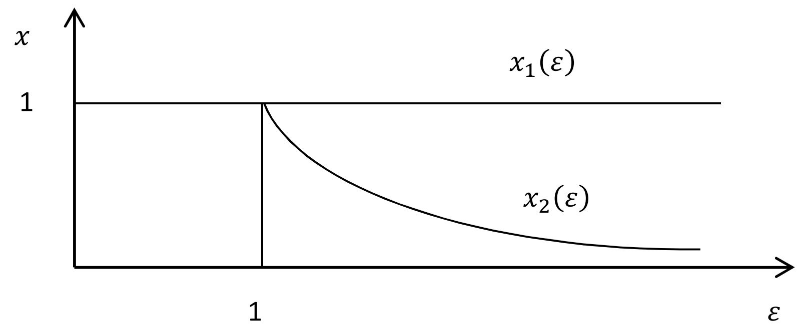
$$x = 1 + \varepsilon x lnx = \varphi(x,\varepsilon) \tag{3.43}$$

ამ განტოლებას აქვს ორი ამონახსნი: $x_1(\xi) = 1$, ყოველი $\varepsilon > 0$-თვის. მეორე ამონახსნის მნიშვნელობა დამოკიდებულია ε-ზე მოცემულ ცხრილში 3.1.

ცხრილი 3.1

ε	1	2	3	4	5	6	7	8	9	10
x_2	1	0.28	0.15	0.1	0.07	0.05	0.04	0.04	0.03	0.027

$x_1(\varepsilon)$ და $x_2(\varepsilon)$ ამონახსნების გრაფიკული გამოსახულება ნაჩვენებია ნახ. 3.2



ნახ. 3.2

(3.43) გამოსახულების ამონახსნების და (3.42) საშუალებით შესაძლებელია გავიგოთ კვაზიწონასწორული ფასები.

თუ მოთხოვნის ხვედრი წილი დიდი არ არის $(0 \leq \varepsilon \leq 1)$, მაშინ არსებობს კვაზიწონასწორული ფასის ერთადერთი მნიშვნელობა

$$P^*(\tau) = \lambda_{min}(\tau)S(\tau) = \frac{S(\tau)}{h_{max}(\tau)} \qquad (3.44)$$

ამ გამოსახულებიდან ჩანს, რომ მოთხოვნის ხვედრითი წილი პატარა სიდიდეა და ხელფასი ფიქსირებული კვაზიწონასწორული ფასი უკუპროპორციულია მაქსიმალური ტექნიკური დონის $h_{max}(\tau)$.

ეს დასკვნა სრულიად თანხმობაშია პოლიტეკონომიკის ერთ-ერთ კონცეფციასთან, სადაც დეკლარილებულია ფასის შემცირება ტექნოლოგიური დონის ამაღლების შემთხვევაში. ეს ასეა მრავალი პროდუქციისათვის.

თუ (3.44)-ში ჩავსვამთ (3.31) მოვიღებთ:

$$P^*(\tau) = S(\tau)\left(\frac{\mu I_T(t)}{\upsilon}(1 - exp\{-\upsilon(\tau - t)\}) + h_{max}(t)exp\{-\upsilon(\tau - t)\}\right)^{-1} \qquad (3.45)$$

ეს გამოსახულება განსაზღვრავს კვაზიწონასწორული ფასის "მდორე" ევოლუციას და მის დამოკიდებულებას $S(\tau)$ -ხელფასზე და ინვესტიციებზე, რომლებიც უცვლელია $\tau \geq t$ ინტერვალის ფარგლებში.

თუ მოთხოვნის ხვედრითი წილი $\varepsilon > 1$, მაშინ არსებობს ორი კვაზიწონასწორული ფასები

$$P_1{}^*(\tau) = \frac{S(\tau)}{h_{max}(\tau)}$$

$$P_2{}^*(\tau) = \frac{S(\tau)}{x_2 h_{max}(\tau)} \qquad (3.46)$$

ახლა ვნახოთ როგორია რენტაბელობს დონის $\lambda^*(\tau)$ ქცევა კვაზისტაციონალური ფასების შემთხვევაში.

(3.34) და (3.46) თანახმად

$$\lambda_1{}^* = \lambda_{min}, \qquad \lambda_2{}^* = \frac{\lambda_{min}}{x_2(\varepsilon)} \qquad (3.47)$$

სადაც $x_2(\varepsilon)$ განისაზღვრება ცხრილი 3.1 - მიხედვით.

როგორც ადვილად დავასკვნით, კვაზისტაციონალური ფასის P^* - შემთხვევაში ფუნქციონირება შეუძლიათ მხოლოდ იმ საწარმოებს, რომლებსაც გააჩნიათ “საუკეთესო” ტექნოლოგიები, ხოლო კვაზისტაციონალური $P_2{}^*$- ფასის შემთხვევაში რენტაბელური წარმოების დიაპაზონი უფრო ფართოა.

შედეგად მნიშვნელოვანია საკითხი კვაზიწონასწორული ფასების (3.44) და (3.46) მდგრადობის შესახებ.

როგორც ზემოთ ვაჩვენეთ (3.21), მდგრადობის პირობას აქვს შემდეგი სახე

$$0 < \frac{\partial F(P,\tau)}{\partial P} < \frac{\partial Y(P,\tau)}{\partial P} \tag{3.48}$$

(3.38)-(3.40) და (3.34) გათვალისწინებით, გვექნება შემდეგი მდგრადობის პირობა

$$\varepsilon\left(1 - ln\frac{P}{S\lambda_{min}(\tau)}\right) < 1 \tag{3.49}$$

$P_1{}^*$ (3.46) კვაზიწონასწორული ფასისათვის მდგრადობის არე

$$0 < \varepsilon < 1 \tag{3.50}$$

ხოლო კვაზიწონასწორული ფასის $P_2{}^*(\tau)$ (3.46) მდგრადობისათვის, აუცილებელია შემდეგი უტოლობის შესრულება

$$0 < \varepsilon\big(1 + lnx_2(\varepsilon)\big) < 1, \qquad \varepsilon > 1 \tag{3.51}$$

ცხრილი 3.1 გათვალისწინებით მივიღებთ, რომ ეს პირობა სრულდება $\varepsilon > 1$-სათვის, შესაბამისად, კვაზიწონასწორული ფასი $P_2{}^*$ მდგრადია.

3.4 შიგა ინვესტიციები

ახლა განვიხილოთ ე.წ. ჩაკეტილი ეკონომიკა, სადაც ტექნოლოგიების განახლება ხორციელდება საკუთარი რესურსეით. ამ შემთხვევაში მთელი პროდუქციის $Y(P,\tau)$ გარკვეული ნაწილი იხარჯება ინვესტირებაში ახალი ტექნოლოგიების შეძენისათვის, ანუ

$$I_T(t) = \beta(\tau)Y(P,\tau)$$

$\beta(\tau) < 1$- კოეფიციენტი შეიძლება იცვლებოდეს დროის განხილულ ინტერვალში. მათემატიკური გარდაქმნების გამარტივებისათვის ქვემოთ ყველგან ვგულისხმობთ, რომ ინვესტიციის აბსოლუტური მნიშვნელობა მუდმივია განსახილველ დროით ინტერვალზე $\tau > t$, ანუ

$$I_T(\tau) = I_T(t) = \beta(\tau)Y(P,\tau) \qquad (3.52)$$

მიღებული დაშვებების შედეგად მაქსიმალური ტექნოლოგიური დონის ცვლილება $\tau \geq t$ ინტერვალში აღინიშნება შემდეგი განტოლებით

$$\frac{dh_{max}(\tau)}{d\tau} = -\upsilon h_{max}(\tau) + \mu\beta(\tau)Y(P,\tau)$$

$$h^0{}_{max} = h_{max}(\tau) \qquad (3.53)$$

ამ განტოლების ამონახსნია

$$h_{max}(\tau) = \frac{\mu\beta(\tau)Y(P,\tau)}{\upsilon}(1 - exp\{-\upsilon(\tau - t)\}) + h_{max}(t)exp\{-\upsilon(\tau - t)\} = L \qquad (3.54)$$

“საუკეთესო” ტექნოლოგიის მნიშვნელობა იქნება

$$\lambda_{min}(\tau) = (L)^{-1} \qquad (3.55)$$

რადგანაც ამ შემთხვევაში შემოსავლების ნაწილი მოდის ტექნოლოგიების განახლებაში, რენტაბელობის პირობას ექნება შემდეგი სახე

$$P(\tau) - \beta(\tau)Y(P,\tau) - \lambda(\tau)S(\tau) \geq 0 \qquad (3.56)$$

აქედან გამომდინარე, ამ შემთხვევისათვის მხოლოდ ის საწარმოები შეძლებენ ფუნქციონირებას, რომლებშიც დანახარჯები “ადამიანურ” რესურსზე იქნება ნაკლები ვიდრე

$$\lambda^*(\tau) = \frac{\gamma(t)P(\tau)}{S(\tau)}, \quad 0 < \gamma(t) = 1 - \beta(\tau) < 1 \qquad (3.57)$$

თუ ჩავსვამთ ამ სიდიდეს გამოსახულებებში (3.38)-(3.40), (3.59) განსაზღვრავს საქონლის საერთო მოწოდების სიდიდეს

$$Y(P,\tau) = M(\tau)\left(1 - \frac{\lambda_{min}(\tau)S(\tau)}{\gamma(t)P(\tau)}\right) \tag{3.58}$$

ხოლო მოთხოვნილება სამუშაო ძალაზე განისაზღვრება შემდეგი გამოსახულებით

$$R^E(P,\tau) = M(\tau)\lambda_{min}(\tau)ln\frac{\gamma P(\tau)}{\lambda_{min}(\tau)S(\tau)} \tag{3.59}$$

ხოლო მოთხოვნის სიდიდე

$$F(P,\tau) = \varepsilon M(\tau)\frac{\lambda_{min}S(\tau)}{P(\tau)}ln\frac{\gamma(t)P(\tau)}{\lambda_{min}(\tau)S(\tau)} \tag{3.60}$$

(3.23) პირობის შესაბამისად, კვაზიწონასწორული ფასი განისაზღვრება შემდეგი განტოლების სახით:

$$\varepsilon\frac{\lambda_{min}\,(\tau)S(\tau)}{P(\tau)}ln\frac{\gamma(t)P(\tau)}{\lambda_{min}(\tau)S(\tau)} = 1 - \frac{\lambda_{min}\,(\tau)S(\tau)}{\gamma(t)P(\tau)} \tag{3.61}$$

(3.57) და (3.42)-ის გათვალისწინებით მივიღებთ

$$\frac{x}{\gamma} = 1 + \varepsilon x ln\frac{x}{\gamma}, \quad \gamma = 1-\beta, \quad \varepsilon > 0, \quad 0 < \beta < 1 \tag{3.62}$$

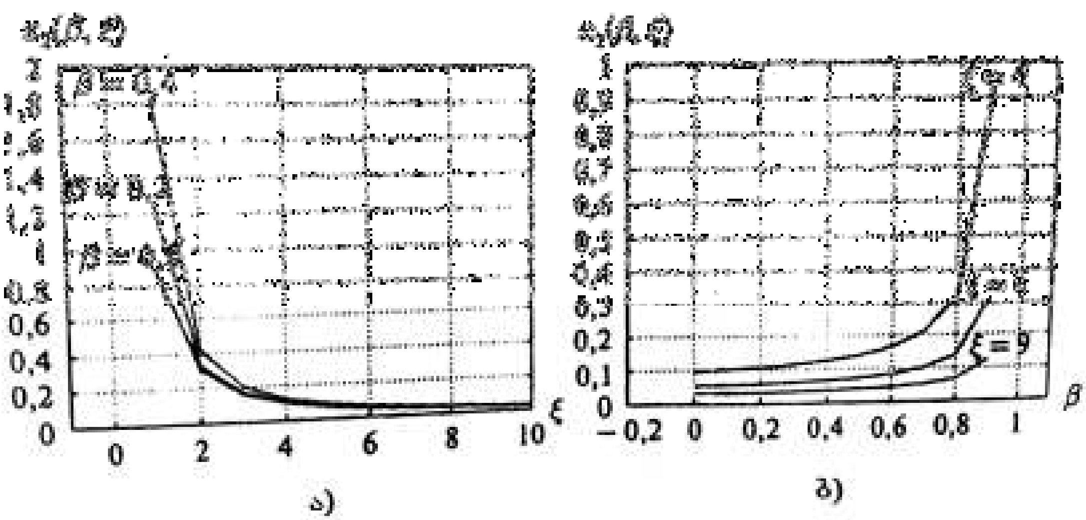
სადაც x ცვლადი განსაზღვრულია ტოლობით - (3.42). (3.62) განტოლების ერთ-ერთი ამონახსნია $x_1 = \gamma$, ნებისმიერი მოთხოვნის ხვედრითი წილის დროს. მეორე ამონახსნის მნიშვნელობის სხვადასხვა ინვესტიციის სიდიდეების შემთხვევისათვის (β) და მოთხოვნის ხვედრი წილის ε სხვადასხვა მნიშვნელობებისათვის მოცემულია ცხრილი 3.2.

ცხრილი 3.2

ξ / β	1	2	3	4	5	6	7	8	9	10
0	1,000	0,285	0,149	0,097	0,070	0,054	0,044	0,036	0,031	0,027
0.1	0,900	0,306	0,158	0,102	0,073	0,056	0,045	0,038	0,032	0,028
0.2	1,273	0,333	0,169	0,108	0,077	0,059	0,048	0,039	0,034	0,029
0.3	1,499	0,369	0,184	0,116	0,082	0,063	0,050	0,042	0,035	0,030
0.	1,850	0,4	0,	0,	0,	0,	0,	0,	0,	0,

4		21	204	127	089	068	054	044	038	032
0.5	2,461	0,500	0,233	0,142	0,099	0,074	0,059	0,048	0,041	0,035
0.6	3,726	0,636	0,281	0,166	0,114	0,085	0,066	0,054	0,045	0,039
0.7	7,338	0,925	0,372	0,210	0,140	0,102	0,079	0,063	0,053	0,045
0.8	28,665	1,863	0,617	0,318	0,200	0,140	0,106	0,083	0,068	0,057
0.9	2201,646	14,332	2,446	0,931	0,492	0,308	0,214	0,159	0,124	0,100

ნახ. 3.3 - ნაჩვენებია $x_2(\beta,\varepsilon)$ ფუნქციის გრაფიკები.



ნახ. 3.3

თუ ჩვენ გავითვალისწინებთ (3.42), (3.25), ჩაკეტილი ეკონომიკის შემთხვევაში გვექნება კვაზიწონასწორული ფასები:

$$P_1{}^*(\tau) = \frac{S(\tau)}{(1-\beta)h_{max}(\tau)}$$

$$P_2{}^*(\tau) = \frac{S(\tau)}{x_2(\beta,\varepsilon)h_{max}(\tau)} \tag{3.63}$$

თუ ჩვენ გავაერთიანებთ (3.63) და (3.54), მივიღებთ გამოსახულებას, რომელიც ახასიათებს კვაზიწონასწორულ ფასებს "ნელი" დროით სკალაში

$$P_1{}^*(\tau) = \frac{S(\tau)}{(1-\beta)}\left(\frac{\mu\beta(\tau)Y(P,\tau)}{\upsilon}(1-exp\{-\upsilon(\tau-t)\}) + h_{max}(t)exp\{-\upsilon(\tau-t)\}\right)^{-1}$$

$$P_2^*(\tau) = \frac{S(\tau)}{x_2(\beta,\varepsilon)}\left(\frac{\mu\beta(t)Y(t)}{\nu}(1-exp\{-\nu(\tau-t)\}) + h_{max}(t)exp\{-\nu(\tau-t)\}\right)^{-1} \tag{3.64}$$

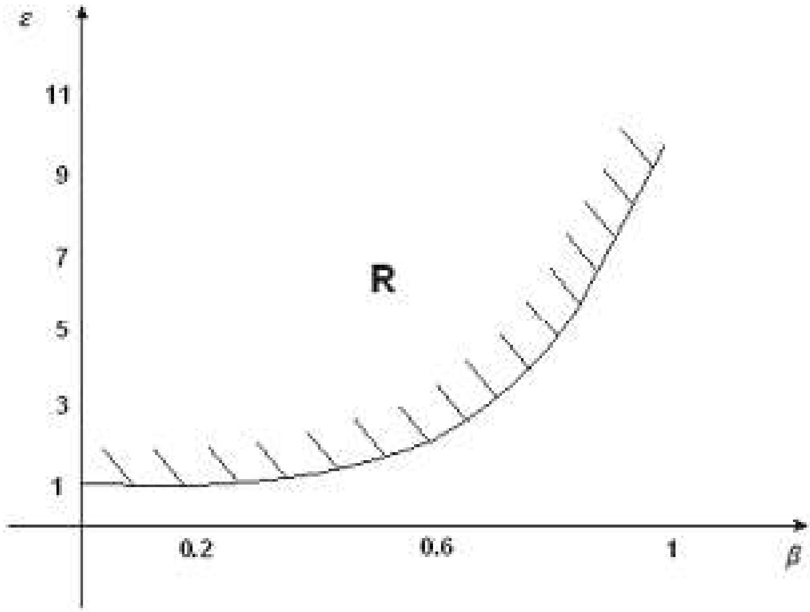
ახლა ვნახოთ როდის გვექნება რენტაბელობის დონე კვაზიწონასწორული ფასებისათვის. (3.58), (3.63) თანახმად

$$\lambda_1^* = \lambda_{min}, \qquad \lambda_2^* = \frac{1-\beta}{x_2(\beta,\varepsilon)}\lambda_{min} \tag{3.65}$$

როგორც ვხედავთ ისევე როგორც გარე ინვესტიციების შემთხვევაში კვაზიწონასწორული ფასის $P_1{}^*$ შემთხვევაში, რენტაბელურები არიან მხოლოდ ის წარმოებები, რომლებიც აღჭურვილნი არიან "საუკეთესო" ტექნოლოგიებით. $P_2{}^*$კვაზიწონასწორული ფასის შემთხვევაში რენტაბელობის დიაპაზონი ფართოვდება, იმ შემთხვევაში თუ პარამეტრები (β,ε) მიეკუთვნებიან "გაფართოებული რენტაბელობის" სიმრავლეს.

$$R = \{(\beta,\varepsilon): (1-\beta) \geq (\beta,\varepsilon), \qquad \beta \in (0,1), \qquad \varepsilon > 0\} \tag{3.66}$$

R სიმრავლე ნაჩვენებია ნახ. 3.4.



ნახ. 3.4

კვაზიწონასწორული ფასების მდგრადობის კვლევისათვის გამოვიყენოთ ე.წ. ვალრასის პირობა (3.21). (3.21), (3.61) მიხედვით გვექნება შემდეგი პირობები:

− P_1^* - კვაზიწონასწორული ფასისათვის

$$(1-\beta)\varepsilon < 1 \tag{3.67}$$

− P_2^* - კვაზიწონასწორული ფასისათვის

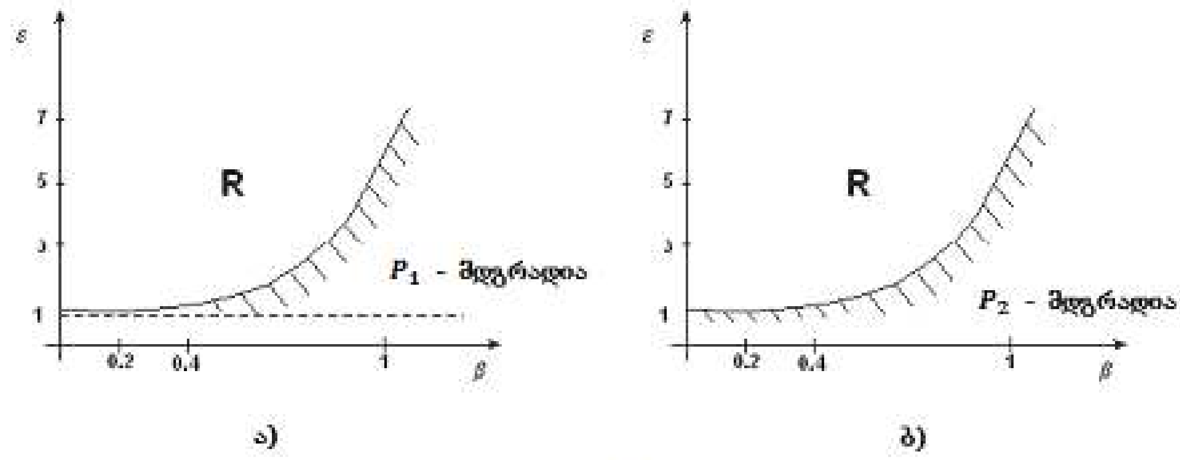
$$(1-\beta)\varepsilon\left(1 + ln\frac{x_2(\beta,\varepsilon)}{1-\beta}\right) < 1 \tag{3.68}$$

შესაბამისი მდგრადობის არეების საზღვრები განისაზღვრება შემდეგი განტოლებებით:

$$\psi_1(\beta,\varepsilon) = (1-\beta)\varepsilon = 1 \tag{3.69}$$

$$\psi_2(\beta,\varepsilon) = (1-\beta)\varepsilon\left(1 + ln\frac{x_2(\beta,\varepsilon)}{1-\beta}\right) = 1$$

ნახ. 3.5 ნაჩვენებია კვაზიწონასწორული ფასების P_1^* და P_2^* მდგრადობის არეების



ნახ. 3.5

დამოკიდებულება მოთხოვნის ხვედრითი წილთან და $\varepsilon > 0$ ინვესტიციების წილთან მიმართებაში.

თავი 4. სივრცული ეკონომიკური პროცესების დინამიკური ანალიზი საინვესტიციო პოლიტიკის განსაზღვრისათვის

სივრცული ევოლუციური ეკონომიკური სისტემის ქვეშ იგულისხმება რეგიონი, რომლის რაიონებს შორის ხდება საქონლის, პროდუქციის, ინვესტიციების ურთიერთგაცვლა.

თანამედროვე ეკონომიკა – სივრცული ეკონომიკაა. ტრადიციულ ეკონომიკაში ეკონომიკური პროცესის ანალიზი დაიყვანება წონასწორული მდგომარეობების პოვნაზე. როდესაც გარემო პირობების ცვლილება ეკონომიკურ პროცესზე უმნიშვნელოა, როდესაც ეკონომიკური სისტემა მცირედ გადაიხრება წონასწორული მდგომარეობიდან, მის მდგრადობას უზრუნველყოფენ საბაზრო მექანიზმები. სივრცული ეკონომიკის დროს კი გვაქვს ურთიერთდამოკიდებული და ურთიერთმოქმედი დინამიკური ეკონომიკური პროცესები, რომლებიც მიმდინარეობენ დროის სხვადასხვა მასშტაბებში. ამ პროცესების აღმოჩენა და მათზე დაკვირვება შეუძლებელია როგორც წონასწორულ მდგომარეობაში, ასევე მეგასისტემასთან ურთიერთობისას. როდესაც გარემო პირობები მნიშვნელოვნად იცვლება, ხშირად საბაზრო მექანიზმები უძლურები არიან შეინარჩუნონ ეკონომიკური სისტემის წონასწორობა „ძველი“ მდგომარეობის ან „ახლის“ მიმართ. სივრცული ევოლუციური ეკონომიკის შესწავლისა და მართვისათვის აუცილებელია შესაბამისი მათემატიკური მოდელების შექმნა, რომლებიც სტრუქტურულად წარმოადგენენ განსაზღვრულ სივრცულ არეებში ლოკალიზებულ ურთიერთმოქმედ ეკონომიკების მოდელების ერთობლიობებს.

აქვე უნდა შევნიშნოთ, რომ სივრცე-დროითი არესათვის დამახასიათებელი უწყვეტი მათემატიკის პრინციპები (დიფერენციალური და ინტეგრალური განტოლებები), ხშირ შემთხვევაში არასაკმარისია ინტელექტუალური სისტემების – წარმოების და საინვესტიციო პოლიტიკის მართვის სისტემების – მოდელირებისა და მართვისათვის. და მოითხოვს ალგებრული და ტოპოლოგიური მეთოდების გამოყენებას.

წარმოდგენილ ნაშრომში მათემატიკის ორივე პრინციპია (უწყვეტი და დისკრეტული მათემატიკა) გამოყენებული. სივრცული ეკონომიკის მოდელის სტრუქტურა აიგება მაკროსისტემურ მიდგომაზე, სადაც ინვესტიციების ნაკადის ფორმირება ხდება ენტროპიის მაქსიმიზაციის პრინციპის საფუძველზე. ეს საშუალებას გვაძლევს განვსაზღვროთ საწყისი, საბაზისო მონაცემების სივრცეში ის არეები, რომლებიც განაპირობებენ საინვესტიციო პოლიტიკის ”თვისობრივ მახასიათებლებს”. შემდეგ, ინვესტორის მიერ მოპოვებული ინფორმაციის რაოდენობრივი მახასიათებელი (ენტროპია) ფასდება იმ საბაზისო ინფორმაციის კლასებთან მიმართებაში, რომელიც მიეცემა ინვესტორს მაკროსისტემური მიდგომის შედეგად. ეს საშუალებას აძლევს ინვესტორს შეამციროს რისკი ახალი ინფორმაციის შეფასების საფუძველზე.

განვიხილოთ ეკონომიკური სივრცე რომელიც შედგება N ქვესისტემისაგან (რაიონისაგან). დავუშვათ, განსახილველ ეკონომიკურ სივრცეში აწარმოებენ გარკვეული ტიპის პროდუქტს. ვგულისხმობთ, რომ m რაიონის შესაძლებლობანი აღნიშნული ტიპის პროდუქციის წარმოებისათვის დამოკიდებულია დანარჩენ ყველა რაიონის ეკონომიკურ მაჩვენებლებზე.

განვიხილოთ ინვესტირების პროცესი. თითოეული m ქვესისტემა (რაიონი) დროის მოცემულ t მომენტისათვის დავახასითოთ იმ ინვესტიციებით $I_m(t)$, რომლებიც იდება ამ ქვესისტემაში. ინვესტიციების ქვეშ ვგულისხმობთ ნაკადურ ცვლადს, ანუ, დროის ერთეულში განხორციელებულ ინვესტიციების მოცულობას. $I_m(t)$ ინვესტიციების ნაკადი თითოეულ ქვესისტემაში განვსაზღვროთ შემდეგი დიფერენციალური განტოლების საშუალებით:

$$\frac{dI_m(t)}{dt} = \alpha_m \cdot I_m(t) + \sum_{s=1} x_{sm}(t) - \sum_{s=1} x_{ms}(t)$$

(4.1)

სადაც x_{sm} ინვესტიციის ნაკადია განხორციელებული s რაიონიდან m რაიონში.

$\alpha_m = \alpha_m(x; y)$ - ორი ცვლადის ფუნქციაა, რომელიც ზრდადია პირველი ცვლადის (შემოსავლების) მიმართ და კლებადი მეორე ცვლადის (დანახარჯები, ამორტიზაცია) მიმართ.

განვიხილოთ შემთხვევა, როცა m რაიონიდან სხვა რაიონებში განხორციელებული ჯამური ინვესტიცია პროპორციულია $I_m(t)$:

$$\sum_{s \neq m} x_{ms}(t) = \varphi(c_m, \mu) \cdot I_m(t) \tag{4.2}$$

სადაც $\varphi(c_m;\ \mu)$ არის ორი ცვლადის ფუნქცია, რომელიც კლებადია პირველი ცვლადის და ზრდადია მეორე ცვლადების მიმართ. c_m არის m რაიონის მახასიათებელი პარამეტრი (რაც უფრო უკეთესი საინვესტიციო პირობებია m რაიონში, მით უფრო დიდია c_m). μ სისტემური პარამეტრია [7], საერთო ყველა ქვესისტემისათვის (რაიონისათვის), რომელიც ახასიათებს ქვესისტემებს შორის ეკონომიკურ ურთიერთკავშირს. $0 \le \mu \le 1$ (როდესაც რაიონებს შორის არ არსებობს ურთიეთკავშირი, მაშინ $\mu = 0$; როცა ყველა რაიონი ერთმანეთთანაა დაკავშირებული, მაშინ $\mu = 1$).

ენტროპიის მაქსიმიზაციის პრინციპის თანახმად, უალბათესია $x_{ms}(t)$ სიდიდეების ის მნიშვნელობები, რომლებიც აკმაყოფილებენ (2) პირობას და რომლებზედაც ფუნქცია:

$$S(t) = -\sum_{m,s} x_{ms}(t) \cdot \ln \frac{x_{ms}(t)}{v_s(t)} \tag{4.3}$$

აღწევს მაქსიმუმს ($v_s(t)$არის რაიონის მიზიდვის ფუნქცია, რომელიც ახასიათებს საინვესტიციო გარემოს აღნიშნულ რაიონში და წარმოადგენს ინვესტიციების წილს m რაიონში [7,8]). პირობითი ექსტრემუმის პოვნის, ლაგრანჟის მეთოდის თანახმად (2), (3) ამოცანის ამონახსნია [9]:

$$x_{ms}(t) = \frac{v_s(t)}{\sum_{k=1} v_k(t)} \cdot \varphi(c_m, \mu) \cdot I_m(t) \tag{4.4}$$

(2) და (4)-ის გათვალისწინებით (1) განტოლება მიიღებს შემდეგ სახეს:

$$\frac{dI_m(t)}{dt} = \left(\alpha_m - \varphi(c_m(t), \mu)\right) \cdot I_m(t) + \frac{v_m(t)}{\sum_{k=1} v_k(t)} \cdot \sum_{k=1} \varphi(c_k(t), \mu) \cdot I_k(t) \tag{4.5}$$

განვიხილოთ შემთხვევა, როცა რაიონის მიზიდვის ფუნქცია წარმოადგენს ლოგისტიკურ ფუნქციას:

$$v_m(t) = I_m(t)\left(V_m - I_m(t)\right) \tag{4.6}$$

სადაც V_m არის m რაიონში ინვესტიციების ზღვრული მნიშვნელობა

(6)-ის გათვალისწინებით (5) ტოლობა ასე გადაიწერება:

$$\frac{dI_m(t)}{dt} = \left(\alpha_m - \varphi(c_m(t),\mu)\right)\cdot I_m(t) + \frac{I_m(t)\left(V_k - I_m(t)\right)}{\sum_{k=1} I_k(t)\left(V_k - I_k(t)\right)}\cdot \sum_{k=1}\varphi(c_k(t),\mu)\cdot I_k(t) \qquad (4.7)$$

(7) ტოლობიდან გამომდინარე, რეგიონში წონასწორობის აუცილებელი პირობაა:

$$\frac{I_m(t)\left(V_m - I_m(t)\right)}{\sum_{k=1} I_k(t)\left(V_m - I_k(t)\right)}\cdot \sum_{k=1}\varphi(c_k(t),\mu)\cdot I_k(t) = \left(\varphi(c_m(t),\mu) - \alpha_m\right)\cdot I_m(t) \qquad (4.8)$$

ადგილი აქვს შემდეგ თეორემას [10]: თუ რომელიმე $m\epsilon 1,N$- თვის სრულდება დადებითი წონასწორობის არსებობის აუცილებელი პირობა, მაშინ იგივე პირობა შესრულდება ნებისმიერი $m\epsilon 1,N$-

ისმის კითხვები: 1. თუ პარამეტრების რომელიმე მნიშვნელობებისათვის სისტემას აქვს წონასწორობის ერთზე მეტი მდგომარეობა, მაშინ რომელში აღმოჩნდება სისტემა? 2. პარამეტრების „უმნიშვნელო“ ცვლილება რა დროს იწვევს კატასტროფას - სისტემის მდგომარეობის „მკვეთრ“ ცვლილებას?

აღნიშნულ კითხვებზე პასუხის გასაცემად ვისარგებლოთ ტომის საკლასიფიკაციო თეორემით [5]. ცვლადთა გარდაქმნის შედეგად (8) განტოლებები შეიძლება ჩაიწეროს შემდეგი სახით:

$$\left(Z_m\right)^3 + \lambda_m\cdot Z_m + \delta_m = 0 \qquad (4.9)$$

სადაც λ_m და δ_m კოეფიციენტები განისაზღვრებიან განტოლება (8) პარამეტრების საშუალებით. განტოლება (9) მიხედვით გვაქვს „წყობის“ ტიპის კატასტროფა [11].

$\alpha_m,\ \mu,\ \{c_k\}_{k=\overline{1,N}};\ \{V_k\}_{k=\overline{1;N}}$ პარამეტრების ცვლილებით იცვლება λ_m და δ_m სიდიდეები და (9) ზედაპირზე მიიღება რაღაც წირი. ტომის საკლასიფიკაციო თეორემის თანახმად, იმისდამიხედვით, ჰკვეთს თუ არა $(\lambda_m;\delta_m)$ სიბრტყეზე მიღებული შესაბამისი წირი არეს, რომელიც შემოსაზღვრულია $16\cdot\left(\lambda_m\right)^3 + 27\cdot\left(\delta_m\right)^2 = 0$ ტოლობით მოცემული წირით, შეგვიძლია ვუპასუხოთ კითხვებს: I) შესაძლებელია თუ არა წონასწორობის ერთი მდგომარეობიდან მეორეშე გადასვლა კატასტროფის გარეშე; II) თუ კი ასეთი გადასვლა შესაძლებელია, მაშინ როგორ უნდა ვცვალოთ სისტემის პარამეტრები ასეთი გადასვლის განსახორციელებლად?

ახლა როდესაც საწყის მონაცემთა, ბაზისური სივრცე დაყოფილია კლასებად, რომლებიც განაპირობებენ საინვესტიციო პოლიტიკის „თვისობრივ მახასიათებლებს''. ინვესტორის მიერ მოპოვებული ინფორმაციის რაოდენობრივი

მახასიათებელი (ენტროპია) ფასდება იმ საბაზისო ინფორმაციის კლასებთან მიმართებაში, რაც საშუალებას აძლევს ინვესტორს შეამციროს რისკი ახალი ინფორმაციის ანალიზის საფუძველზე. ასეთი კვლევის აუცილებლობა განსაკუთრებით მაშინ დგება, როდესაც რეგიონში მიმდინარე ეკონომიკური პროცესის მახასიათებელი პარამეტრების მნიშვნელობები უახლოვდებიან იმ არეს, რომლის დროსაც, მათი უმნიშვნელო ცვლილებას შეიძლება მოყვეს ეკონომიკური სიტუაციის მკვეთრი ცვლილება.

ქვემოთ განვიხილოთ ე.წ. მესერული სიმრავლეების თეორიის ძირითადი ელემენტები [12], რომლის საფუზველზე შეგვიძლია კორექტირება გავუკეთოთ ჩვენს გადაწყვეტილებას ინვესტირების სიდიდის თაობაზე.

ვთქვათ X რაიმე სიმრავლეა. X სიმრავლის ქვესიმრავლეთა A სიმრავლეს ეწოდება X სიმრავლის მესერი თუ სრულდება შემდეგი ოთხი პირობა.

1) $X \in A$, $\varnothing \in A$

2) თუ $a \in A$ და $b \in A$ მაშინ $a \cup b \in A$.

3) თუ $a \in A$ და $b \in A$ მაშინ $a \cap b \in A$.

4) თუ $a \in A$ და $b \in A$ მაშინ $a - b \in A$.

ვიტყვით, რომ A მესერის a_1 და a_2 ელემენტები არიან დიზიუნქციურები, თუ $a_1 \cap a_2 = \varnothing$.

ვთქვათ A არის X სიმრავლის მესერი. $a \in A$ ელემენტს ეწოდება A მესერის ატომი, თუ $a \neq \varnothing$ და A-ში არ არსებობს ისეთ $b \neq \varnothing$ ელემენტი, რომ $b \neq a$ და $b \subset a$.

ვთქვათ A არის X სიმრავლის მესერი. X სიმრავლის ქვესიმრავლეთა B სიმრავლეს ეწოდება A მესერის შკალა, თუ A მესერის ყოველი ელემენტი შეიძლება გამოისახოს B სიმრავლის ელემენტების გაერთიანების, თანაკვეთის და სხვაობის საშუალეებით.

A მესერის B შკალას ეწოდება საბაზო, თუ B-ს არც ესეთი ელემენტი არ შეიძლება გამოისახოს B სიმრავლის სხვა ელემენტების გაერთიანების, თანაკვეთის და სხვაობის საშუალებით.

A მესერის BS შკალას ეწოდება ატომური შკალა თუ მისი ყოველი ელემენტი არის ატომი.

ვთქვათ X ელემენტთა (ინფორმაციათა) ნებისმიერი სიმრავლეა, A არის $X-$ ის მესერი, BS კი A მესერის ატომური შკალა: $BS=\{b_k,\ k\in K\}$, სადაც K არის სასრული ან თვლადი სიმრავლე.

დავუშვათ, რომ ყოველ $x_0\in X$ ელემენტისათვის მოცემულია სანდოობის დისკრეტული განაწილების BS შკალაზე, ე.ი. მოცემულია არაუარყოფითი რიცხვებით ფუნქცია $p(b_k;x_0)$ $p(b_k;x_0)\geq 0$ ამასთან სრულდება ნორმირების პირობა .

$$\sum_{b_k\in BS} P(b_x,x_0)=1$$

მოცემულ სანდოობას დისკრეტულ განაწილებას ეწოდება $x_0\in X$ ელემენტის სანდოობას საწყისი განაწილება A მესერის მიმართ (რადგან ყოველ მესერს აქვს ერთადერთი ატომური შკალა, ამიტომ ეს განმარტება არის კორექტული).

ყოველი $\sigma\in A$ სიმრავლისათვის სიდიდეს

$$P=\sum_{b_k\in\delta} P(b_k;x_0)$$

ეწოდება სანდოობა (ალბათობა) ფაქტისა, რომ $x_0\in\sigma$.

რადგან $p(b_k;x_0)\geq 0$ ყოველი $b_k\in BS-$ თვის, და

$$\sum_{b_k\in BS} P(b_x,x_0)=1$$

ამიტომ ცხადია, რომ $0\leq P\leq 1$. P=1 შეიძლება გავიგოთ როგორც აუცილებელი ხდომილობა ფაქტორს $x_0\in$ ნ, ხოლო P=0 კი როგორც შეუძლებელი ხდომილობა ფაქტისა $x_0\in$ ნ

საწყისი განაწილების კონკრეტიზაცია დამოკიდებულია მომხმარებელზე. და ახასიათებს x_0 წერტილის შესახებ მასთან არსებულ ინფორმაციას. თუ x_0 წერტილის შესახებ რაიმე საწყისი ინფორმაცია არ არსებობს მაშინ განვიხილავთ თანაბარ განაწილებას ან ნორმალურ განაწილებას.

ვთქვათ $x_0\in X$ წერტილისათვის ცნობილია სანდოობის საწყისი განაწილება. ყოველი ნ$(x_0)\in B$-თვის განვმარტოთ x_0 წერტილის სანდოობის მეორადი განაწილება ბაიესის ფორმულის საშუალებით.

$$p_\sigma(b_k;\, x_0)=\begin{cases}0; & \text{თუ} \quad b_k\in\sigma\\ \dfrac{p(b_k;\, x_0)}{\sum\limits_{b_i\in\sigma}p(b_i;\, x_0)}, & \text{თუ} \quad b_k\notin\sigma\end{cases}$$

$x_0\in X$ წერტილის სანდოობის საწყისი განაწილების შესაბამისი ენტროპია გამოითვლება ფორმულით:

$$H(X(x_0))=-\sum_{b_k\subset X}p(b_k;x_0)\cdot\log_2 p(b_k;x_0),$$

ხოლო x_0 წერტილის სანდოობის მეორადი განაწილების შესაბამისი ენტროპია კი გამოითვლება ფორმულით:

$$H(\sigma(x_0))=-\sum_{b_k\subset\sigma}p_\sigma(b_k;x_0)\cdot\log_2 p_\sigma(b_k;x_0).$$

საწყისი და მეორადი განაწილების შესაბამისი ენტროპიების სხვაობას ვუწოდოთ σ სიმრავლის მიერ მოტანილი ინფორმაციის რაოდენობა.

შევნიშნოთ, რომ ეს სხვაობა შესაძლებელია იყოს როგორც დადებითი ასევე 0, ან უარყოფითი. σ სიმრავლის მიერ მოტანილი ინფორმაციის რაოდენობა (ენტროპიების სხვაობა) არის უარყოფითი ნიშნავს, რომ σ კონფლიქტშია საწყის განაწილებასთან.

მოვიყვანოთ მაგალითი რომელიც გვიჩვენებს, თუ როგორ შეიძლება გამოვიყენოთ სიმრავლეთა მესერის თეორია საინვესტიციო პოლიტიკის შემუშავებისათვის.

ვთქვათ გადაწყვეტილია საინვესტიციო თანხის განთავსება $b_1; b_2; \dots b_{10}$ რაიონებში. თითოეული $b_k,\, k\in\overline{1;10}$ რაიონისათვის ცნობილია ინვესტიციის განთავსების ალბათობა $p(b_k)$, ჩვენს შემთხვევაში $p(b_k)=\dfrac{\nu_k(t)}{\sum\limits_s \nu_s(t)}$.

ზოგადობის შეუზღუდავად დაუშვათ, რომ გვაქვს შემდეგი საწყისი განაწილება:

$$p(b_k)=\begin{cases}c_1, & \text{თუ} \quad k\in\overline{1;\,6}\\ c_2, & \text{თუ} \quad k\in\overline{7;\,10}\end{cases}$$

ნორმირების პირობის ძალით $6\cdot c_1+4\cdot c_2=1$.

გამოვთვალოთ საწყისი განაწილების შესაბამისი ენტროპია:

$$H(X) = -\sum_{k=1}^{10} p(b_k) \cdot \log_2 p(b_k) = -6 \cdot c_1 \cdot \log_2 c_1 - 4 \cdot c_2 \cdot \log_2 c_2 .$$

ვთქვათ, ჩვენთვის ცნობილი გახდა რაღაც σ ინფორმაცია, რომლის გამო გადავწყვიტეთ, რომ ინვესტიცია ჩავდოთ $b_1; b_2; b_3; b_4; b_5; b_6$ რაიონიდან ერთ-ერთში. σ ინფორმაციის შესაბამისი სანდოობის მეორადი განაწილება მოიცემა ფორმულით:

$$p_\sigma(b_k) = \begin{cases} \frac{1}{6}, & \text{თუ} \quad k \in \overline{1;\,6} \\ 0, & \text{თუ} \quad k \in \overline{7;\,10} \end{cases}$$

მეორადი განაწილების შესაბამისი ენტროპიაა:

$$H(\sigma) = -\sum_{k=1}^{10} p_\sigma(b_k) \cdot \log_2 p_\sigma(b_k) = \log_2 6 .$$

σ ცნობის მიერ მოტანილი ინფორმაციის რაოდენობაა:

$$I(\sigma) = -6 \cdot c_1 \cdot \log_2 c_1 - 4 \cdot c_2 \cdot \log_2 c_2 - \log_2 6 .$$

(I) თუ $c_1 = c_2 = 0{,}1$ (თუ საწყისი განაწილება არის თანაბარი განაწილება), მაშინ

$$I(\sigma) = \log_2 10 - \log_2 6 = \log_2 \frac{5}{3} .$$

(II) თუ $c_1 = \frac{1}{6}$, $c_2 = 0$, მაშინ

$I(\sigma) = \log_2 6 - \log_2 6 = 0$ - σ ცნობა არავითარ ახალ ინფორმაციას არ გვაწვდის;

(III) თუ $c_1 = 0; c_2 = \frac{1}{4}$, მაშინ

$I(\sigma) = \log_2 4 - \log_2 5 = \log_2 \frac{4}{5} < 0$ - σ ცნობა კონფლიქტშია საწყის განაწილებასთან.

მაშასადამე, σ ცნობის მიერ მოტანილი ინფორმაციის რაოდენობა საშუალებას გვაძლევს შევაფასოთ თუ რამდენად ღირებულია σ ცნობა შემუშავებული საინვესტიციო პოლიტიკის ცვლილებისათვის.

თავი 5. ქალაქის ფუნქციონალური ქვესისტემების იერარქიული განთავსება

ქალაქის ფუნქციონალური ქვესისტემების სივრცულ ურთიერთქმედებათა მოდელის აგებისას ქალაქი წარმოადგენს n რაიონში განლაგებულ ფუნქციონალური ქვესისტემების ერთობლიობას:

— "ქალაქის წარმომქმნელი" ბაზა, რომელიც ხასიათდება რაიონებში განლაგებული B ობიექტების სიმრავლით, რომელსაც შემდგომში აღვნიშნავთ $G = \{G_j, j = \overline{1,n}\}$.

— მომსახურების ცენტრები – K, რომელიც შეესაბამება K- სხვადასხვა მომსახურების სფეროებს. მომსახურების ცენტრები დავახაისათოთ მათში მომსახურე პერსონების საშუალებით, რასაც აღვნიშნავთ შესაბამისი ვექტორით $M^K = \{M_j^K, j = \overline{1,n}\}, K = \overline{1,K}$.

— მაცხოვრებლები, რომლებსაც დავახასიათებთ მათი საცხოვრებელი ადგილის მიხედვით: $N = \{N_i, i = \overline{1,n}\}$.

ქალაქის ქვესისტემების ურთიერთქმედება რეალიზდება მოსახლეობის გადაადგილებით. ამასთანავე, მივიღეთ, რომ:

G^B – განაწილება ფიქსირებულია.

M^K – განაწილება დამოკიდებულია N, G^B,

N^S, $s = \overline{1,K}$ (სხვადასხვა ტიპის კლიენტები მომსახურების სფეროში)

განაწილება N დამოკიდებულია იმ მოქალაქეების რაოდენობაზე, რომლებიც მუშაობენ. მომუშავეთა საერთო რაოდენობა აღვნიშნოთ M-ით.

$$M = G^B + \sum_{\cap K} M^K$$

ვთვლით, რომ თუ მომსახურების ცენტრების განაწილება ფიქსირებულია, მაშინ მოსახლეობის გადაადგილება არის დამოუკიდებელი შემთხვევითი სიდიდეები a_{ij}.

მოსახლეობის გადაადგილების პერიოდული ციკლი (დღე-ღამობაში) ხასიათდება შემთხვევითი მდგომარეობათა მატრიცით $X = \{x_{ij}\}$.

ამოცანის ასეთი ფორმულირება საშუალებას გვაძლევს განვსაზღვროთ ერთადერთი მდგრადი მდგომარეობა $\overset{0}{X}$, რომელიც შეესაბამება ენტროპიის მაქსიმუმს

$$E(X) = -\sum_{a_{i,j}} x_{ij} \, ln \frac{x_{ij}}{a_{ij}}$$

იმ შეზღუდვების დროს, რომელიც ედება X.

a_{ij} სიდიდეები განისაზღვრებიან იმ კომუნიკაციის რაობით, რომელიც არსებობს i და j შორის.

მაგალითად, გადაადგილების დრო, ღირებულება, მანძილი და ა. შ. სხვა), რომლებიც მიიღებიან სტატისტიკური მონაცემების საშუალებით, $f(h)$ განაწილების მიხედვით, როდესაც გადაადგილება ხორციელდება გარკვეული მიზნით.

მოსახლეობის განთავსების მოდელი მდგომარეობს შემდგომში:

დავუშვათ, გვაქვს სხვადასხვა ტიპის – r რაიონების სიმრავლე $I^r, r = \overline{1,R}$.

ცნობილია მოსახლეობის რაოდენობა თითოეულში N_i^r – და მათი რაოდენობა დაფიქსირებულია. დანარჩენ i რაიონში $i = I^H$ მოსახლეობის რაოდენობების სიდიდეებმა უნდა დააკმაყოფილონ შემდეგი უტოლობები

$$g_i \leq N_i \leq d_i \tag{5.1}$$

აქ g_i – იმ მოსახლეობის რაოდენობაა, რომლებიც იკავებენ არასარეკონსტრუქციო საცხოვრებელ ფონდს.

d_i – მოსახლეობის დასაშვები რაოდენობაა, რომელიც შეესაბამება იმ ტერიტორიას, რომელიც რჩება მას შემდეგ, რაც იქ განთავსდება მომსახურების ობიექტები.

$$d_i = Z_i^N \left(A_i - \sum_k M_i^k / Z_i^k \right);$$

A_i - i რაიონის ტერიტორია, რომელზეც უნდა განთავსდეს მოსახლეობა და მომსახურების ცენტრები.

Z_i^N, Z_i^k – ქვესისტემების განთავსების სიმკვრივეებია.

მომსახურე პერსონალის მდგრადი გადაადგილების მნიშვნელობები j სამუშაო ადგილიდან i საცხოვრებელი ადგილას არის შემდგომი ამოცანის ამონახსნი

$$\max_{ji} - \sum_{ji} y_{ji} ln \frac{y_{ji}}{\mu_{ji}}$$

$$\sum_i y_{ji} = E; \qquad j = \overline{1,n}$$

$$\sum_{y_{ji}} = c^r N_i^r, \ \ i \in I^r, \ \ r = \overline{1,R} \tag{5.2}$$

$$c^H g_i \le \sum_j y_{ji} \le c^H d_i, \qquad i \in I^h$$

აქ c^r, c^H – მოსახლეობის დასაქმებულობის კოეფიციენტია სხვადასხვა ტიპის რაიონებისათვის. აქვე უნდა შევნიშნოთ, რომ ქალაქის პროექტირება ხასიათდება ე.წ. ტერიტორიული იერარქიულობით. იერარქიულობის ერთ-ერთი ინტერპრეტაცია შეიძლება იყოს შემდეგი: განაშენიანება ხორციელდება რამოდენიმე ეტაპად, რაც საშუალებას იძლევა, შემდგომ ეტაპებზე გათვალისწინებული იყოს ყველა უნებლიეთ გამოტოვებული დეტალები, გათვალისწინებულ იქნას მიმდინარე რეალიები და დაშვებული შეცდომები.

ქვემოთ განვიხილოთ მოსახლეობის განთავსების ორ ეტაპიანი იერარქიული პროცედურა.

თავდაპირველად გამოვყოთ მოსახლეობის ის ჯგუფი, რომლებიც უნდა განვათავსოთ რაიონებში, რომლებიც წარმოადგენენ სხვადასხვა ტიპის რაიონებს (ასე მაგალითად სამრეწველო რაიონი, ცენტრალური რაიონი და ა.შ.). შემდგომ ამ ჯგუფს მოსახლეობისას გავანაწილებთ ყველა რაიონში, სადაც მოსახლეობის რაოდენობა ფიქსირებული არ არის, ეს რაიონებია $i \in I^H$.

ფორმალურად ჩვენ მოსახლეობის "სასტარტო" და "საფინალო" წერტილების რაოდენობები ცნობილია, მაგრამ დავაწესებთ შეზღუდვებს $R+1$ "საფინალო" წერტილებზე, რომლებიც შეესაბამება როგორც ფიქსირებულ, ისე არაფიქსირებულ რაიონებს მოსახლეობისას. ასეთი სახით ფორმალიზებული ამოცანა მიიღებს შემდეგ სახეს:

$$\max_{uji} \sum_{ji} U_{ji} \frac{\mu_{ji}}{U_{ji}}$$

$$\sum_i U_{ji} = E_j, \quad j = \overline{1,n} \tag{5.3}$$

$$\sum_{i \in I^r} \sum_j U_{ji} = c^r \bar{N}^r, \quad r = \overline{1,R}$$

$$\sum_{i \in I^H} \sum_j U_{ji} = c^H \bar{N}^H, \quad U_{ji} \ge 0$$

აქ მიღებულია ასეთი აღნიშვნები

$$\bar{N}^r = \sum_{i \in I^r} N_i^r, \quad r = \overline{1,R}$$

$$\bar{N}^H = \sum_{i \in I^H} N_i = \bar{N} - \sum_r \bar{N}^r, \quad r = \overline{1,R}$$

ანუ აქ $\bar{N}$ – ქალაქის სრული მოსახლეობაა.

(2) ამოცანის ამონახსნი $\overset{0}{U}_{ji}$ – შეიძლება მიღებული იყოს სტანდარტული გზით [1].

ზოგადად $\overset{0}{U}_{ji}$ მნიშვნელობები არ აკმაყოფილებენ იმ შეზღუდვებს, რომლებიც დადებული არის "სტარტზე" ამოცანა (2)-ში.

რადგანაც მოსახლეობის რაოდენობა N_i^r, $i \in I^r$, $r = \overline{1,R}$, იმ რაიონებში, სადაც მოსახლეობა ფიქსირებულია და განსაზღვრულია აპრიორულად, ამიტომ ჩვენ განვიხილავთ იმ ნაკადების $\overset{0}{U}_{ji}$ ასეთ რაიონებში $i \in I^H$. მაშინ

$$T_i = \frac{1}{c^H} \sum_j \overset{0}{U}_{ji}, \ i \in I^H \tag{5.4}$$

შეიძლება გაგებული იქნას როგორც "პოტენციალური" რაიონები სადაც მოსახლეობას უნდა გადასვლა. ხოლო $T_i/\bar{N}^H$ – კი შეიძლება ინტერპრეტირებული იქნას როგორც ალბათობები იმ მოსახლეობისა, რომლებიც ცხოვრობენ რაიონებში არაფიქსირებული მაცხოვრებლებით i.

ახლა უკვე შესაძლებელია ავაგოთ სქემა – მოსახლეობისაგან შემთხვევითი შერჩევა რაიონებისა – "შენაკადისა" $i \in I^H$ ალბათობებით $T_i/\bar{N}^H$, შეზღუდვებისას (5.1)

აქედან გამომდინარე მეორე ეტაპზე გვაქვს ამოცანა:

$$\max_{N_i} \sum_{i \in I^H} N_i \, ln \frac{T_i}{N_i}$$

$$\sum_{i \in I^H} N_i = \bar{N}^H \tag{5.5}$$

$$g_i \le N_i \le d_i, \qquad i \in I^H$$

T_i პოტენციალების განსაზღვრა ხორციელდება [1] მოცემული სქემის მიხედვით. მივიღებთ:

$$\overset{0}{U}_{ji} = \begin{cases} a_j h_r \mu_{ji}, & i \in I^r,\ r = \overline{1,R} \\ a_j h_H \mu_{ji}, & i \in I^{/H},\ j = \overline{1,n} \end{cases} \tag{5.6}$$

a_j, h_r, h_Hმამრავლები მიიღება იმ განტოლებათა სისტემების საშუალებით, რომელიც მიიღება (6) ჩასმით (3) შეზღუდვებში.

თუ გამოვიყენებთ აღნიშვნებს

$$\rho_i^H \sum_{i \in I^H} \mu_{ji}, \qquad \rho_i^r \sum_{i \in I^r} \mu_{ji}, \quad r = \overline{1,R}$$

მივიღებთ განტოლებებს, რომლითაც განისაზღვრება (6) გამოსახულების პარამეტრები

$$a_j \left(\sum_r h_r \rho_i^r + h_H \rho_i^H \right) = E_j, \quad j = \overline{1,n}$$

$$h_r \sum_j a_i \rho_j^r = c^r \bar{N}^r, \qquad r = \overline{1,R}$$

$$h_H \sum_j a_i \rho_j^H = c^H \bar{N}^H \tag{5.7 }$$

სადაც ρ_j – არის “შენაკადების” ჯგუფის არჩევის ალბათობები მოცემული j “წყაროსათვის”.

განტოლება (7) ამონახსნებია

$$T_i = \frac{h_H}{c_H} \sum_j a_i \rho_j^H, \qquad i = I^H$$

ხოლო თუ a_j-ის განვსაზღვრავთ h_r, h_H-ის საშუალებით, გვექნება

$$T_i = \frac{h_H}{c_H} \sum_j F_j \frac{\mu_{ji}}{\sum_r h_r \rho_i^r + h_H \rho_i^H}$$

იმ შემთხვევაში, თი არ გვაქვს რაიონები ფიქსირებული მაცხოვრებლებით $(h_r = 0), T_i$ - პოტენციალებისათვის გვექნება

$$T_i = \frac{1}{c_H} \sum_j E_j \frac{\mu_{ji}}{\sum_i \mu_{ji}}$$

5.2 მომსახურების ობიექტების იერარქიული განთავსება შეზღუდვების დროს

შეიძლება იყოს ისეთი სიტუაცია, როდესაც j რაიონში A_j – ტერიტორია არასაკმარისია მომსახურების სფეროს ობიექტების განთავსებისათვის. თუ მივიღებთ მომსახურების უფრო მაღალი პრიორიტეტულობას, მაშინ ობიექტების განთავსების ამოცანა დაკავშირებულია შემდეგ საერთო შეზღუდვებთან:

$$\sum_{k=2}^{k} \frac{1}{Z_j^k} \sum_{l=1}^{L} x_{ej}^k \leq A_j, \quad j = \overline{1,n} \tag{5.8}$$

სადაც $L = (k+2)n$.

ამასთანავე მივიღოთ, რომ თუ რაიონში მოთხოვნილია მომსახურებაში განთავსდეს E_j^k სამუშაო ადგილები, რომელიც ნაკლებია რომელიღაც ε_j^k-ზე, მაშინ ამ ტიპის მომსახურებას საერთოდ უკუაგდებენ. ანუ მომსახურების k ტიპისათვის განსაზღვრულია გარკვეული რაიონები.

$$Z_3^k = \left\{j: E_j^k \geq \varepsilon_j^k\right\} \qquad k = \overline{1,K}$$

ხოლო დარჩენილ რაიონებში $j \in Z_4^k$,

$$Z_4^k = \{1, \dots, n\} \backslash Z_3^k,$$

$$E_j^k = 0$$

ანუ $x_{ej}^k = 0, \quad l = \overline{1,L}, \quad j \in Z_4^k$.

სისტემის წონასწორული-მდგრადი მდგომარეობის ცნების შესაბამისად [13] მივიღებთ, რომ ყოველი ფიქსირებული k ქვესისტემის მდგრადი მდგომარეობა განისაზღვრება სხვა ქვესისტემების ფიქსირებული მდგომარეობების დროს.

5.3 ოპტიმალური საინვესტიციო პოლიტიკის შემუშავებისათვის

ზემოთ ნაჩვენებია, რომ Q ვექტორის კოორდინატები სასურველია იყოს მცირე, წინააღმდეგ შემთხვევაში K კომპლექსის გეომეტრიული სტრუქტურა შეზღუდავს K-ს ნებისმიერი π მოდელის ცვლილების შესაძლებლობას (იხ.თავი . პარაგრაფი).

ამასთან, გეომეტრიის თვალსაზრისით იდეალურია ისეთი მიმართება X და Y სიმრავლეთა შორის, როცა ყოველი x_i λ მიმართებაშია ყოველ y_j-თან, მაგრამ

ასეთ შემთხვევაში მოსალოდნელია სისტემის ზოგიერთი მახასიათებლის გაუარესება. გარდა ამისა, ასეთი სისტემის პრაქტიკული რეალიზაცია არაა მიზანშეწონილი. ქვემოთ ნაჩვენებია აღწერილი გეომეტრიული მეთოდის ისეთი მოდიფიკაცია, რომელიც საშუალებას გვაძლევს ავირჩიოთ ორი სავარაუდო λ და λ_1 მიმართებებიდან ერთ-ერთი განსაზღვრული ω კრიტერიუმის მიხედვით.

$$\omega(x,y):\{\lambda\} \longrightarrow R$$

სადაც $\{\lambda\}$ აღნიშნავს X და Y სიმრავლეებს შორის λ მიმართებათა სიმრავლეს.

თუ გაჩნდება X და Y სიმრავლეების ელემენტებს შორის კავშირებისათვის წონების მინიჭების აუცილებლობა, ჩვენ შეგვიძლია მცირე (გარკვეულ რიცხვზე ნაკლები) წონის მქონე კავშირები უგულებელვყოთ, ე.ი. ვიგულისხმოთ, რომ განსახილველ λ მიმართებაში შესაბამის ადგილებზე ნულები წერია.

ამრიგად, ჩვენი ამოცანა შეიძლება ჩამოვაყალიბოთ შემდეგნაირად: ვიპოვოთ ისეთი λ მიმართება (მოცემული X და Y სიმრავლეებისთვის), რომელზეც ω ფუნქცია მაქსიმუმს აღწევს.

X და Y სიმრავლეებს შორის მიმართებათა $\{\lambda\}$ სიმრავლე შეგვიძლია განვიხილოთ, როგორც ნაწილობრივ დალაგებული სიმრავლე: ორი λ და λ_1 მიმართებისათვის $\lambda \le \lambda_1$ მაშინ და მხოლოდ მაშინ, როცა λ_1-ის შესაბამის ინციდენტურობის მატრიცაში 1- წერია ყველა იმ უჯრედში, სადაც 1- წერია λ–ს შესაბამის ინციდენტურობის მატრიცაში.

პრაქტიკაში ფუნქციონირებადი სისტემებისათვის ω ფუნქცია ხშირად აკმაყოფილებს შემდეგ პირობას (რომელსაც ჩვენ (*)-ით ავღნიშნავთ):

ვთქვათ, λ_1, და λ_3 ორი ისეთი მიმართებაა X და Y სიმრავლეებს შორის, რომ $\lambda_1 \le \lambda_3$ და ამასთან სრულდება $\omega(\lambda_1) \le \omega(\lambda_3)$ უტოლობა. მაშინ თუ X და Y სიმრავლეთა შორის ისეთი λ მიმართებათა სიმრავლე, რომლებიც აკმაყოფილებენ $\lambda_1 < \lambda < \lambda_3$ უტოლობებს, არაცარიელია, ის შეიცავს ისეთ λ_2 ელემენტს, რომლისთივსაც $\omega(\lambda_1) \le \omega(\lambda_2)$.

ქვემოთ მოყვანილია თეორემა, რომელიც გვაძლევს სასურველი λ–ს საპოვნ ალგორითმს, თუკი ჩვენ მოვახერხებთ სათანადო „გაფართოებული“ კომპლექსის მოძებნას. კერძოდ, X სიმრავლის თითოეული ელემენტი საჭიროა წარმოვადგინოთ გარკვეულ განზომილებიანი კომპლექსის (და არა წერტილის) სახით ევკლიდურ სივრცეში. აღსანიშნავია, რომ ამ სიმპლექსთა წვეროებს და წახნაგებს რაიმე ფიზიკური შინაარსი საზოგადოდ არ გააჩნიათ.

უფრო ზუსტად რომ ვთქვათ, $\overline{K}$ სიმპლიციალურ კომპლექსს ვუწოდებთ K სიმპლიციალური კომპლექსის გაფართოებას, თუ არსებობს ისეთი $f{:}K \longrightarrow \overline{K}$ ასახვა განსაზღვრული სიმპლექსებზე (და შეთანხმებულ სიმპლექსთა წახნაგებზე), რომ K-ს ყოველი სიმპლექსის განზომილება არ აღემატება ანასახის განზომილებას.

თეორემა% ვთქვათ X და Y სასრული სიმრავლეებია და $\omega_{(x,y)}:\{\lambda\} \longrightarrow R$ ასახვა აკმაყოფილებს (*) პირობას. ვთქვათ, λ ნებისმიერი ისეთი მიმართებაა X და Y სიმრავლეებს შორის, რომლის ინციდენტურობია მატრიცის არც ერთი სტრიქონი და არც ერთი სვეტი არ შედგება მხოლოდ ნულებისაგან. მაშინ არსებობს $K_y(X;\lambda)$ კომპლექსის ისეთი გაფართოებული $\overline{K}_y\left(\overline{X};\overline{\lambda}\right)$ კომპლექსი (მაშასადამე $f{:}K \longrightarrow \overline{K}$ ასახვა), და ალგორითმი, რომელიც საშუალებას გვაძლევს ვიპოვოთ ისეთი λ მიმართება, რომლისთივსაც

$$\omega_{(x.y)}(\lambda')=\max_{\lambda_1 \geq \lambda} \omega_{(x.\ y)}(\lambda_1)$$

ალგორითმი შემდეგია:

მოვახდინოთ K კომპლექსის Q - ანალიზი $f(y_1);..;f(y_n)$ სიმპლექსების მიმართ და ვთქვათ, K-ს სტრუქტურის ვექტორია $Q=(Q_0; Q_1;...; Q_m)$. დავუშვათ, $i \geq 0$ ისეთი უმცირესი ინდექსია, რომლისთივსაც $q_{i.}>1$. ავირჩიოთ ნებისმიერი ორი ისეთი სიმპლექსი $f(y_1); ...; f(y_n)$ სიმპლექსებიდან, რომლებიც არ არიან i-ბმულნი (ცხადია, ისინი i-1 ბმულნი არიან) და გავხადოთ ისინი i-ბმული ერთ-ერთი მათგანის შეცვლით კომპლექსით, რომელიც მიიღება მისთივს მეორე სიმპლექსის ნებისმიერი წვეროს დამატებით (რომელიც მას არ ეკუთვნოდა). განვიხილოთ ყველა შესაძლო ასეთი ცვლილების სტრუქტურის ვექტორები და რეალურად მოვახდინოთ ის ცვლილება, რომლის სტრუქტურის ვექტორს აქვს უმცირესი i-რი კოორდინატი. თუ ასეთი რამდენიმეა, მაშინ მათ შორის მივიჩნევთ იმას, რომელსაც აქვს უმცირესი i+1 კოორდინატი და ა.შ. ამგვარად, მივიღებთ ახალ $\overline{K}_1$ კომპლექსს. მოვახდინოთ ინდუქცირებული ცვლილება K კომპლექსშიც (წინასახეებზე ანუ შესაბამის y-ს დავუმატოთ შესაბამისი წვერო). მივიღებთ K_1 კომპლექსს, რომლისთივსაც $\overline{K}_1$ გაფართოებული კომპლექსია .ვთქვათ, K_1-ის შესაბამისი მიმართებაა λ_1. ამ პროცედურის გამეორება გვაძლევს მიმართებათა $\lambda=\lambda_0, \lambda_1, \lambda_2,$ მიმდევრობას. ვთქვათ, $j \geq 1$ ისეთი უმცირესი ინდექსია, რომლისთვისაც $\omega(\lambda_{j+1})<\omega(\lambda_j)$, მაშინ

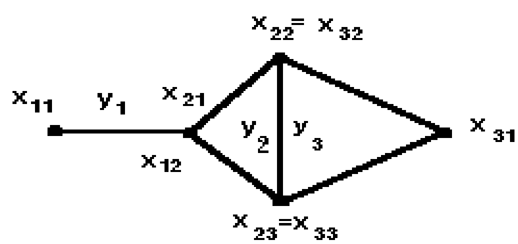
$$\omega(\lambda_j)=\max_{\lambda_i \geq \lambda} \omega_{(x.\ y)}(\lambda_i)$$

სიძნელე, რომელსაც ამ პროცედურის გამოყენებისას ვაწყდებით, იმაში მდგომარეობს, რომ რთულია სასრული გაფართოების მოძებნა. გარდა ამისა, ასეთი

გაფართოებული კომპლექსის განზომილება შეიძლება იყოს ისეთი დიდი, რომ გაჭირდეს მისი Q ანალიზი.

პარქტიკულად, უნდა დავკმაყოფილდეთ ისეთი გაფართოებული კომპლექსების განხილვით, რომელთა მიმართ ზემოთ მოყვანილი ალგორითმის გამოყენება მოგვცემს სასურველ შედეგს გარკვეული მიახლოებით.

მოყვანილი თეორემის საილუსტრაციოდ მოვიყვანოთ ასეთი მაგალითი. ვთქვათ, X შედგება სამი x_1, x_2, x_3, რაიონის, ხოლო Y სამი y_1, y_2, y_3, საწარმოსაგან, რომლებსიც უნდა განხორციელდეს ინვესტირება. ვთქვათ, დასაწყისისათვის თითოეულ რაიონს ამარაგებს მხოლოდ ერთი საწარმო (ზოგადობის შეუზღუდავად შეგვიძლია ვიგულისხმოთ, რომ x_i-ს ამარაგებს y_i) ტერიტორიული სიახლოვის მიხედვით. შესაბამისი მიმართების მატრიცას ექნება დიაგონალური სახე. ყოველ $i \neq j$ (i, $j \in \{1,2,3\}$) წყვილს შევუსაბამოთ ნამდვილი $t(i,j)$ რიცხვი, ვთქვათ მოსალოდნელი სატრანსპორტო ხარჯები (x_i,y_j) კავშირისათვის.

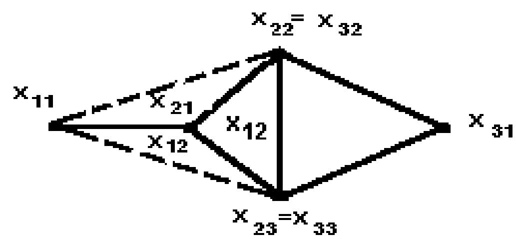


ნახ 5.1

დავუშვათ, ეს შესაბამისობა ისეთია, რომ $t(2,1)<t(2,3)<t(1,3)$ მაშინ, საწყისი გაფართოებული $\overline{K}$ კომპლექსის როლში ავიღოთ სურ. 5.1-ზე გამოსახული კომპლექსი, რომლის სტრუქტურის ვექტორია (1,2,2). მართლაც, $\overline{K}$ კომპლექსის განზომილება (რაც განისაზღვრება როგორც მასში შემევალი სიმპლექსების განზომილებათა მაქსიმუმი), ცხადია ორის ტოლია. $\overline{K}$ შეიცავს ორ-ორ განზომილებიან სიმპლექსს: y_2-ს და y_3-ს, რომლებიც ერთმანეთს არ ემთხვევიან. ამგვარად, Q_2=2. მაგრამ y_2-ს და y_3 1-ბმულნი არიან. მათ გააჩნიათ საერთო ერთგანზომილებიანი წახნაგი (მონაკვეთი). y_1-ის განზომილება 1-ის ტოლია, ამასთან, ის არ არის არც y_2-ის და არც y_3-ის წახნაგი. ამდენად, Q_1=2. ნახაზიდან აგრეთვე ცხადია, რომ $\overline{K}$ არ შეიცავს სავსებით იზოლირებულ ნაწილებს, ასე რომ Q_0=1. ამგვარად, საბოლოოდ Q=(1,2,2). ავღნიშნოთ $\overline{K}$-ს შესაბამისი მიმართება λ_0-ით.

ზემოთ განხილული ალგორითმი გვეუბნება, რომ საინვესტიციო რისკის შემცირებისათვის უნდა გადავიდეთ ახალ კომპლექსზე (სურ. 5.2) და შესაბამისად,

ახალ $\overline{\lambda}_1$ მიმართებაზე X და Y სიმრავლეებს შორის, რომლის სტრუქტურის ვექტორია (1,1,2,1). სურ.2, (რომელშიც წერტილი ხაზებით ახლად გაჩენილი ბმებია აღნიშნული) გვიჩვენებს, რომ y_1 და y_2 სიმპლექსები ერთი (სამგანზომილებიანი) სიმპლექსის წახნაგებს წარმოადგენს. ეს ბუნებრივიცაა: „არაგაფართოებულ" სიმპლექსში x_1 და x_2 რაიონების ერთმანეთთან დაკავშირება y_1 და y_2 საწარმოებს ერთ $[x_1 x_2]$ მონაკვეთად (ან მის წვეროებად) აქცევს.



ნახ 5.2

მეორე მხრივ, ის ფაქტი, რომ t(2,1) მინიმალურია, გვარწმუნებს იმაში, რომ ჩვენი ალგორითმის „არჩევანი" - x_1 და x_2 რაიონების ბმა - ოპტიმალურია. ახლა უნდა შევადაროთ $\omega(\lambda_0)$ და $\omega(\lambda_1)$. თუ $\omega(\lambda_0)>\omega(\lambda_1)$, მაშინ ყველაზე ეფექტური ყოფილა თავდაპირველად არსებული სისტემა (თითო საწარმო ამარგებს თითო რაიონს) და მისი გაუმჯობესება შეუძლებელია. თუ $\omega(\lambda_0)\leq\omega(\lambda_1)$, მაშინ ავაგოთ ახალი კომპლექსი (რაღაც λ_2 მიმართებით) ალგორითმის მიხედვით.

გეომეტრიული სურათი ცხადყოფს, რომ ამისათვის საჭიროა y_2 და y_3 სიმპლექსების საერთო წახნაგის განზომილების გაზრდა. ამასვე ადასტურებს ჩვენს მაგალითში $(i,j)\rightarrow t(i,j)$ შესაბამისობის არჩევანიც.

ახლა, ისღა დაგვრჩენია, შევადაროთ $\omega(\lambda_1)$ და $\omega(\lambda_2)$. თუ $\omega(\lambda_1)\geq\omega(\lambda_2)$, მაშინ ყველაზე ეფექტური იქნება λ_1 მიმართების შესაბამისი სისტემა, წინააღმდეგ შემთხვევაში უმჯობესია λ_2-ის შესაბამისი სისტემა.

5.4 იერარქიული სისტემების ბმულობის ანალიზი

ახლა გავარკვიოთ, თუ როგორ ხდება ურთიერმოქმედებათა გავრცელება მრავალი რგოლისაგან შემდგარ გრძელ ჯაჭვში.

ვთქვათ, მოცემული გვაქვს სამი X_1, X_2, და X_3 სიმრავლე და ორი მიმართება მათ შორის: $\lambda^{(1,2)} \subseteq X_1 \times X_2$ და $\lambda^{(2,3)} \subseteq X_2 \times X_3$. ცნობილია, რომ ასეთ სიტუაციაში შეიძლება განისაზღვროს $\lambda^{(1,2)}$ და $\lambda^{(2,3)}$ მიმართებათა კომპოზიცია, რომელიც იქნება მიმართება X_1 და X_3 სიმრავლეებს შორის. აღვნიშნოთ $\lambda^{(1,2)} \circ \lambda^{(2,3)}$ კომპოზიცია $\lambda^{(1,3)}$ სიმბოლოთი. თუ $x_1 \in X_1$ და $x_3 \in X_3$ მაშინ მიმართებათა კომპოზიციის განსაზღვრის ძალით $(x1; x3) \in \lambda^{(1,3)}$ მაშინ, თუ არსებობს ისეთი $x_2 \in X_2$, რომ $(x_1; x_2) \in \lambda^{(1,2)}$ და $(x_2, x_3) \in \lambda^{(2,3)}$.

მიმართებათა კომპოზიციის განხილვის აუცილებლობამდე პრაქტიკას მივყავართ, მართლაც, განვიხილოთ ზემოთ მოყვანილი მაგალითი: ვთქვათ, X_2 წარმოადგენს ქალაქში პურის მაღაზიების სიმრავლეს. X_0 - პურის საცხობების სიმრავლეს, ხოლო X_3 - ფქვილის საწყობების სიმრავლეს, მაშინ ბუნებრივია, რომ $\lambda^{(1,2)}$ და $\lambda^{(2,3)}$ მიმართებები განსაზღვრული იყოს შემდეგნაირად: $(x_1; x_2) \in \lambda^{(1,2)}$ მაშინ და მხოლოდ მაშინ, თუ $x_2 \in X_2$ საცხობი ამარაგებს $x_1 \in X_1$ მაღაზიას, ხოლო $(x_2; x_3) \in \lambda^{(2,3)}$ მაშინ და მხოლოდ მაშინ, თუ $x_3 \in X_3$ საწყობიდან ფქვილი მიეწოდება $x_2 \in X_2$ საცხობს. ცხადია უშუალო კავშირი პურის მაღაზიებსა და ფქვილის საწყობებს შორის არ არსებობს, მაგრამ ცხადია აგრეთვე ისიც, რომ X_3 სიმრავლის ელემენტების (ფქვილის საწყობების) შეშფოთებამ როგორღაც შეიძლება იმოქმედოს X_1 სიმრავლის ელემენტებზეც (პურის მაღაზიებზე) და პირიქით, ამდენად პრაქტიკულად საინტერესოა $\lambda^{(1,3)}$ მიმართების და მისი შესაბამისი სიმპლიციალური კომპლექსის შესწავლაც.

ზოგადად, თუ მოცემული გვაქვს n ცალი სიმრავლე $X_1,...,X_n$ და n-1 ცალი მიმართება მათ შორის% $\lambda^{(1,2)} \subseteq X_1 \times X_2,...,\lambda^{(n-1,\ n)} \subseteq X_{n-1} \times X_n$ შეგვიძლია განვიხილოთ ამ მიმართებათა $\lambda^{(1,2)} \circ ... \circ \lambda^{(n-1,\ n)} \subseteq X_1 \times X_n$ კომპოზიცია (აღვნიშნოთ ის $\lambda^{(1,n)}$ -ით), რომელიც შემდეგნაირადაა განსაზღვრული: თუ $x_1 \in X_1$ და $x_n \in X_n$, მაშინ $(x_1; x_n) \in \lambda^{(1,n)}$ მაშინ და მხოლოდ მაშინ თუ არსებობს ისეთი $x_2 \in X_2,...,x_{n-1} \in X_{n-1}$ ელემენტები, რომ $(x^1; x^2) \in \lambda^{(1,2)},...,(x^{n-1}, x^n) \in \lambda^{(n-1,\ n)}$.

პრაქტიკული მოსაზრებიდან ცხადია, რომ ზოგადად $\lambda^{(1,n)}$ მიმართების შესაბამისი სისტემა უფრო მოქნილია n-ის დიდი მნიშვნელობებისათვის. მართლაც,

როგორც აღვნიშნეთ, უშუალო კავშირი X_1 და X_n სიმრავლეების ელემენტებს შორის არ არსებობს. X_1 სიმრავლის ელემენტების შეშფოთებას სისტემამ შეიძლება გაუძლოს იმის ხარჯზე, რომ შეიძლება მოხერხდეს ამ შეშფოთების გადანაწილება $X_2;...;X_n$ სიმრავლეთა ელემენტებზე. მაგალითად, სხვადასხვა მაღაზიაში პურზე არაპროპორციულად გაზრდილი მოთხოვნილება შეიძლება დაკმაყოფილებულ იქნას როგორც პურის საცხობებიდან მაღაზიებში პურის მიწოდების გრაფიკების სათანადო კორექტირებით, ასევე თვით პურის საცხობების ფქვილით მომარაგების გაუმჯობესებითაც, სადაც ეს შესაძლებელია.

ასეთი სახის ემპირიულ დასკვნას მკაცრი მათემატიკური სურათიც ადასტურებს. ვნვხოთ, თუ რას წარმოადგენს $\lambda^{(1,n)}$ მიმართება და მისი შესაბამისი $K^{(1,n)}$ კომპლექსი. განვიხილოთ ჯერ $\lambda^{(n-1,\ n)}$ მიმართება და შესაბამისი სტრუქტურის $Q^{(n-1,\ n)}$ ვექტორი და $K^{(n-1,\ n)}$ კომპლექსი. როგორც ვიცით, სასურველია, რომ $Q^{(n-1,\ n)}$ ვექტორის კოორდინატები იყოს მცირე, რაც იმას ნიშნავს, რომ $K^{(n-1,\ n)}$-ის შემადგენელი სიმპლექსები იყვნენ @კარგად ბმული@. ვთქვათ, ორ Δ და Δ' სიმპლექსს $K^{(n-1, n)}$ -დან აქვს საერთო m განზომილებიანი წახნაგი. ეს იმას ნიშნავს, რომ თუ Δ სიმპლექსი შეესაბამება რომელიღაც $x_n \in X_n$ ელემენტს, ხოლო Δ' - $x_n' \in X_n$ ელემენტს, მაშინ არსებობს ისეთი $x_{n-1}^{(1)},...,x_{n-1}^{(m)}$ ელემენტები X_{n-1}-დან, რომ თითოეული მათგანი $\lambda^{(n-1, n)}$ მიმართებაშია როგორც x_n-თან, ასევე x_n' -თანაც. მეორეს მხრივ, თუ ვიგულისხმებთ, რომ $n>2$, X_{n-1} -ის თითოეული ელემენტი (და კერძოდ $x_{n-1}^{(1)},...,x_{n-1}^{(m)}$ ელემენტებიც) თავად წარმოადგენს სიმპლექსს, როგორიც ეს ჩანს $\lambda^{(n-2, n-1)}$ მიმართებიდან, რადგან, როგორც წესი (პრაქტიკაში), ნებისმიერი $x_{n-1} \in X_{n-1}$ -თვის არსებობს ერთი მაინც ისეთი $x_{n-2} \in X_{n-2}$, რომ $(x_{n-2}, x_{n-1}) \in \lambda_{(n-2, n-1)}$ ამდენად ცხადია, რომ x_n და x_n' ელემენტების შესაბამის სიმპლექსებს $K^{(n-2, n)}$ კომპლექსში ჩვეულებრივ აქვთ სულ მცირე m საერთო წვერო მაინც, რადგან იგივე მართებულია ნებისმიერი სიმპლექსებისათვის, ცხადია, რომ $K^{(n-2, n)}$ კომპლექსი უკეთაა ბმული, ვიდრე $K^{(n-1,\ n)}$ კომპლექსი და შესაბამისად, ზოგადად, $Q^{(n-2,\ n)}$ ვექტორის კოორდინატები უფრო მცირეა, ვიდრე $Q^{(n-1,\ n)}$ ვექტორის. ანალოგიური მსჯელობის გაგრძელებით ვასკვნით, რომ ზოგადად $Q^{(1,\ n)}$ ვექტორის კოორდინატები უფრო მცირეა, ვიდრე $Q^{(i,\ n)}$ $(i=2,...,n-1)$ ვექტორების შემაბამისი კოორდინატები (თუმცა აქ ისიც უნდა აღინიშნოს, რომ ამ ვექტორებს ზოგადად სხვადასხვა განზომილებები აქვთ). ეს კი სწორედ იმას ნიშნავს, რომ X_1 სიმრავლის ელემენტების შეშფოთებას $\lambda^{(1,\ n)}$

მიმართების შესაბამისი სისტემა ჩვეულებრივ უფრო უკეთ გაუძლებს დიდი ნატურალური n რიცხვებისათვის.

5.5 ქალაქური სისტემების ფუნქციონალური ბმულობების ცვლილების ალგორითმი საინვესტიო რისკების მინიმიზაციისათვის

ყოველი λ მიმართება x და y სასრულ სიმრავლეებს შორის წარმოადგენს $y \times x$ დეკარტული ნამრავლის ქვესიმრავლეს, ანუ $\lambda \subset y \times x$. $(y_i, x_j) \in \lambda$, თუ y_i არის λ მიმართებაში x_j-თან. ეს მიმართება შეიძლება ჩავწეროთ ინციდენტურობის მატრიცის სახით $\lambda = (\lambda_{ij})$, სადაც

$$\lambda_{ij} = 1, \text{ თუ } (y_i, x_j) \in \lambda$$

$$\lambda_{ij} = 0, \text{ თუ } (y_i, x_j) \notin \lambda$$

ყოველი λ მიმართება გვაძლევს $k_y(x, y)$ სიმპლეციალურ კომპლექსს, რომელსაც ეწოდება "λ მიმართების სტრუქტურა" და განისაზღვრება შემდეგნაირად:

1. $k_y(x, y)$ არის სიმპლექსების ერთობლიობა

$$\sigma_p : p = 0,1, \dots, N$$

2. ოველი $\sigma_p \in k$ სიმპლექსი განისაზღვრება (p+1) განსხვავებული x_i ელემენტით x სიმრავლიდან, რომლისთვისაც არსებობს ერთი მაინც ელემენტი
   $y_k \in y$ ისეთი, რომ $(y_k;\, x_i) \in \lambda$.
3. სიმპლექსი σ_0^i განსაზღვრულია $x_{i,}, i = \overline{1,n}$.

რ. ეტკინი განიხილავს ამ კომპექსის სტრუქტურულ ვექტორს Q =($Q_{0,\dots,}Q_n$). ვექტორის კოორდინატები ხასიათდება სხვადასხვა განზომილების კომპლექსების კავშირით. ამასთან, ნაჩვენებია, რომ რაც უფრო ნაკლებია სტრუქტურული ვექტორის კოორდინატები, მით უფრო სტაბილურია სისტემა.

ჩვენ მხედველობაში ვიღებთ ე.წ. "ინციდენტურობის წონიან მატრიცას", რაც საშუალებას გვაძლევს არა მარტო შევაფასოთ სისტემის სტაბილურობა, არამედ დავაკვირდეთ როგორ ხდება ურთიერთქმედებათა გავრცელება მრავალი რგოლისაგან შემდგარ გრძელ ჯაჭვში.

ვთქვათ მოცემული გვაქვს სამი X, Y, და Z სიმრავლე და ორი მიმართება მათ შორის $\lambda^{(1,2)} \subseteq X \times Y$ და $\lambda^{(2,3)} \subseteq Y \times Z$. ცნობილია, რომ ასეთ სიტუაციაში

შეიძლება განისაზღვროს $\lambda^{(1,2)}$ და $\lambda^{(2,3)}$ მიმართებათა კომპოზიცია, რომელიც იქნება მიმართება X და Z სიმრავლეებს შორის.

$$\lambda^{(1,3)} = \lambda^{(1,2)} O \lambda^{(2,3)} \subseteq X \times Z$$

სადაც $(x,z)\epsilon\lambda(1,3)$, თუ $\exists y\epsilon Y$, რომ $(x,y)\epsilon\lambda(1,2)$ და $(y,z)\epsilon\lambda(2,3)$.

$\lambda(1,2)$ მიმართებას X და Y სიმრავლეებს შორის, შევუსაბამოთ ინციდენტურობის მატრიცა

$$I(1,2) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix},$$

სადაც $a_{ij}(i = \overline{1,m},\ j = \overline{1,n})$, აღნიშნავს X სიმრავლის i-ური ელემენტის რა ნაწილი შეიცავს Y სიმრავლის j-ურ ელემენტს. ე.ი. $\sum_{j=1}^{n} a_{ij} = 1,\ i = \overline{1,m}$

ანალოგიურად $\lambda(2,3)$ მიმართებას შევუსაბამოთ ინციდენტურობის "წონიანი" მატრიცა.

$$I(2,3) = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \cdots & b_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nk} \end{pmatrix}.$$

სტოქასტიკური მატრიცის მსგავსად, I(1,3) ინციდენტურობის მატრიცა, რომელიც

შეესაბამება λ (1,3) ტოლია

$$I(1,3) = I(1,2) \times I(2,3) = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mk} \end{pmatrix}.$$

აქედან გამომდინარე, ჩვენ ვამცირებთ Q-ს კოორდინატებს, რაც აუმჯობესებს სისტემის კავშირს. იმისათვის, რომ დავახასიათოთ სისტემის კავშირი, ჩვენ განვიხილავთ H ფუნქციას, რომელიც განისაზღვრება როგორც ინფორმაციული ენტროპიის ქცევა სტოქასტიკური ინციდენტურობის მატრიციდან

$$H_i^{ll} = \sum_{j=1}^{k} - b_{ij} ln b_{ij},$$

$$H_i^{l} = \sum_{j=1}^{m} - H_j^{ll} ln a_{ij},$$

$$\omega H_i^{l} = \sum_{i=1}^{n} - H_{i\omega i}^{l} ln \omega_i.$$

აღნიშნული პირობების დროს მაქსიმუმი იქნება

$$H = \ln n \times \ln m \times \ln k.$$

ყველაზე ეფექტური ალგიროთმი იმისათვის, რომ გაიზარდოს H, მდგომარეობს შემდეგში: ვთქვათ, გვაქვს $\lambda_{ij} = \{(i,j)\}$ ისეთი წყვილების სიმრავლე, რომ a_{ij}=0 და ასახვა $\varphi : \lambda_{ij} \to N$, სადაც Nნატურალურ რიცხვთა სიმრავლეა:

$$\varphi(i,j) = \sum_{i=1}^{k} \mathrm{sgn}\, b_{jt} - \sum_{i=1}^{k} \mathrm{sgn}(b_{ji} \cdot c_{ij}).$$

ჩვენ ვირჩევთ ისეთ წყვილებს, რომლებისთვისაც φ ფუნქცია მიაღწევს მაქსიმუმს და შემდეგ ბიჯზე გამოვყოფთ i_0და j_0ინდექსებს ისე, რომ კავშირი x-ის i_0 ელემენტსა და y-ის j_0 ელემენტს შორის H ფუნქციის მაქსიმალურ ზრდას განაპირობებს.

ახლა განვიხილოთ ასეთი ამოცანა: ვთქვათ, რეგიონი დაყოფილია h რაიონად, სადაც ცხოვრობს მოსახლეობას Q ჯგუფი, ხოლო T არის ინტერესების სიმრავლე. განვიხილოთ შემდეგი წონიანი ინციდენტურობის მატრიცები:

$$I(Q,T) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1T} \\ a_{21} & a_{22} & \cdots & a_{2T} \\ \cdots & \cdots & \cdots & \cdots \\ a_{Q1} & a_{Q2} & \cdots & a_{QT} \end{pmatrix}$$

და

$$I(T,L)=\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \cdots & \cdots & \cdots & \cdots \\ b_{T2} & b_{T2} & \cdots & b_{TL} \end{pmatrix},$$

სადაც a_{qt} აღნიშნავს მოსახლეობის q-ური ჯგუფის ცხოვრებაში რა ადგილი უკავია t –ურ ინტერესს, ხოლო b_{tl} გვიჩვენებს t ინტერესი რა დონეზე კმაყოფილდება l რაიონში. ($q=\overline{1,Q};\ t=\overline{1,T},\ l=\overline{1,L}$). ცხადია,

$$\sum_{t=1}^{T} a_{qt} = \sum_{l=1}^{L} d_{tl} = 1.$$

თუ განვიხილავთ მატრიცას:

$$I(Q,L) = I(Q,T)\times I(T,L) \ = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1L} \\ c_{21} & c_{22} & \cdots & c_{2L} \\ \cdots & \cdots & \cdots & \cdots \\ c_{Q1} & c_{Q1} & \cdots & a_{QL} \end{pmatrix},$$

მივიღებთ რომ c_{ql} აღნიშნავს l რაიონში რა ხარისხით დაკმაყოფილდება q ჯგუფის მოსახლეობის საერთო ინეტრესები, $q=\overline{1,Q}$რადგან$\sum_{k=1}^{L} c_{qk} = 1,$.ჩვენ შეგვიძლია გამოვიყენოთ კომუნიკაციურობის მატრიცი, რომელიც ემყარება ენტროპიის მაქსიმიზაციის პრინციპს, რათა მივიღოთ რეგიონში მოსახლეობის გავრცელების სურათი.

ვთქვათx_{ij}აღნიშნავს ($i=\overline{1,Q};\ j=\overline{1,h}$) მოსახლეობის რაოდენობას, რომლებიც ირჩევენ საცხოვრებლად j რეგიონს. მივიღებთ მიზნობრივ ფუნქციას:

$$\phi\ (X) = \sum_{i=1}^{Q}\sum_{j=1}^{h} x_{ij} \ln \frac{c_{ij}}{x_{ij}}$$

და შეზღუდვები

$$x_{ij} \geq 0\ \left(i=\overline{1,Q}\ ;\ j=\overline{1,h}\right),$$

$$p_i = \sum_{j=1}^{L} x_{ij}, \quad R_j = \sum_{i=1}^{Q} x_{ij},$$

სადაც p არის i ჯგუფის მოსახლეობის რაოდენობა, ხოლო Q_j- jრეგიონის.ჩვენი მიზანია ვიპოვოთ ისეთი x_{ij}, რომლის მნიშვნელობებისთვისაც, შეზღუდულობის პირობებში $\Phi(X)$ მიიღებს მაქსიმალურ მნიშვნელობას.გამოთვლები შესაძლებელია პირობითი ექსტრემუმის პოვნის კლასიკური მეთოდის გამოყენებით.

თავი 6. რეგიონში მიმდინარე საინვესტიციო პროცესების სისტემური იმიტაციური მოდელები. დინამიკური განსაკუთრებულებები.

ვთქვათ, განსახილველ რეგიონში ფუნქციონირებს M ურთიერთდაკავშირებული ქვესისტემა (საწარმო). I_m(t)-თი აღვნიშნოთ ინვეsტიციების სიდიდე, რომელიც დროის t მომენტში იდება m საწარმოში. α_m-ით აღვნიშნოთ m საწარმოს მოგების კოეფიციენტი. μ აღნიშნავს რეგიონის მახასიათებელ პარამეტრს, $o \leq \mu \leq 1$. μ გვიჩვენებს თუ რამდენად კარგად არიან დაკავშირებული საწარმოები ერთმანეთთან. თუ ყოველი საწარმო არის იზოლირებული (ანუ არც ერთი საწარმო არაა კავშირში სხვა რომელიმე საწარმოსთან - მაგალითად, არა ჰყავთ საერთო მომწოდებელი) მაშინ $\mu = 0$, ხოლო თუ ყოველ საწარმოს კავშირი აქვს ყოველ მომწოდებელთან, მაშინ $\mu = 1$. სხვა შემთხვევაში $o < \mu < 1$. μ-ს გამოთვლის ერთ-ერთი ვარიანტი მოცემულია დანართი 1-ში.

განვიხილავთ შემთხვევას, როცა რეგიონი ღიაა ინვესტიციების მიმართ (შესაძლებელია რეგიონის გარედან ინვესტიციების შემოდინება და ინვესტიციების გადინება რეგიონის გარეთ). რეგიონის გარე არე ჩავთვალოთ 0-ვანი ნომრის მქონე საწარმოდ. თუ $x_{sm}(t)$ თი აღვნიშნოთ ინვესტიციის სიდიდე, რომელიც დროის t მომენტში s საწარმოდან იდება m საწარმოში, მაშინ თითოეულ საწარმოში ინვესტიციის სიდიდის გამოსათვლელად გვექნება დიფერენციალურ განტოლებათა შემდეგი სისტემა:

$$\frac{dI_m(t)}{dt} = \alpha_m \cdot I_m(t) + \mu \cdot \left(\sum_{s=0}^{M} x_{sm}(t) - \sum_{s=0}^{M} x_{ms}(t)\right), \quad m \in \overline{[1;M]} \qquad (1)$$

განვიხილავთ შემთხვევას, როცა m საწარმოდან სხვა საწარმოებში განხორციელებული ჯამური ინვესტიცია პროპორციულია I_m(t)-სი:

$$\sum_{s=0}^{M} x_{ms}(t) = h_m \cdot I_m(t), \quad m \in \overline{[1;M]} \qquad (2)$$

(2)-ის გათვალისწინებით (1) დიფერენციალურ განტოლებათა სისტემა მიიღებს სახეს:

$$\frac{dI_m(t)}{dt} = (\alpha_m - \mu \cdot h_m) \cdot I_m(t) + \mu \sum_{s=0}^{M} x_{sm}(t), \quad m \in \overline{[1;M]} \qquad (3)$$

თუ $\nu_{sm}(t)$ აღნიშნავს s საწარმოსთვის m საწარმოს მიზიდვის ფუნქციას, მაშინ ენტროპიის მაქსიმიზაციის პრინციპის თანახმად უალბათესია x_{sm}(t) სიდიდეების ის მნიშვნელობები, რომლისთვისაც სრულძება (2) პირობა და რომლისთვისაც ფუნქცია

$$S(t) = \sum_{s=0}^{M} \sum_{m=0}^{M} x_{sm}(t) \cdot ln \frac{x_{sm}(t)}{\nu_{sm}(t)} \qquad (4)$$

აღწევს მაქსიმუმს.

x_{sm}(t) სიდიდეების ის მნიშვნელობები, რომლებიც აკმაყოფილებენ (2) პირობას და რომელზედაც S(t) ფუნქცია აღწევს მაქსიმუმს, მოიცემა ფორმულით (იხ. დანართი 2)

$$x_{sm}(t) = \frac{\nu_{sm}(t) \cdot h_s}{\sum_{k=0}^{M} \nu_{sk}(t)} \cdot I_s(t) \qquad (5)$$

თუ x_{sm}(t) სიდიდეების მიღებულ მნიშვნელობებს შევიტანთ დიფერენციალური განტოლებების (3) სისტემაში, მივიღებთ:

$$\frac{dI_m(t)}{dt} = (\alpha_m - \mu \cdot h_m) \cdot I_m(t) + \mu \sum_{s=0}^{M} \frac{\nu_{sm}(t)}{\nu_{sk}(t)} \cdot h_s \cdot I_s(t), \quad m \in \overline{[1;M]} \qquad (6)$$

განვიხილავთ შემთხვევას, როცა $\nu_{sm}(t)$ ფუნქცია არის ლოგისტიკური ფუნქცია:

$$\nu_{sm}(t) = \delta_{sm} \cdot I_m(t) \cdot \left(V_m - I_m(t)\right), \quad s, m \in \overline{[1;M]} \qquad (7)$$

სადაც δ_{sm} არის s საწარმოსთვის m საწარმოს მიმზიდველობის მახასიათებელი პარამეტრი (δ_{sm} პარამეტრის გამოთვლის ერთ-ერთი ვარიანტი მოცემულია დანართი 3-ში).

თუ დიფერენციალური განტოლებების (6) სისტემაში $\nu_{sm}(t)$ -ს ნაცვლად შევიტანთ მის (7) მნიშვნელობას, მივიღებთ დიფერენციალურ განტოლებათა ცხად სისტემას I_m(t) ცვლადების მიმართ:

$$\frac{dI_m(t)}{dt} = (\alpha_m - \mu \cdot h_m) \cdot I_m(t) + \mu \sum_{s=0}^{M} \frac{\delta_{sm} \cdot I_m(t) \cdot (V_m - I_m(t))}{\sum_{s=0}^{M} \delta_{sk} \cdot I_k(t) \cdot (V_k - I_k(t))} \cdot h_s \cdot I_s(t), \quad m \in \overline{[1;M]} \qquad (8)$$

მიღებული დიფერენციალური განტოლების ამონახსნი (საზოგადოდ, ჩვენ არ შეგვიძლია ამ ამონახსნის ჩაწერა ცხადი სახით, ჩვენ ვპოულობთ მის მიახლოებით მნიშვნელობას) საშუალებას გვაძლევს ვიცოდეთ ყოველ საწარმოში ჩადებული ინვესტიციების უალბათესი მნიშვნელობა დროის t მომენტისათვის.

თუ (8) ტოლობების მარჯვენა მხარეებს გავუტოლებთ 0-ს და ამოვხსნით მიღებულ განტოლებათა სისტემას, ამით ვიპოვით რეგიონში არსებულ წონასწორობის მდგომარეობების შესაძლო მნიშვნელობებს. საზოგადოდ მიღებულ განტოლებათა სისტემას ექნება რამდენიმე (ერთი, ან მეტი) არაუარყოფითი ამონახსნი. შესაბამისად, რეგიონში გვექნება რამდენიმე წონასწორობის მდგომარეობა.

ყოველივე ზემოთქმულის თვალსაჩინოდ ჩვენებისათვის ჩვენ განვიხილავთ შემთხვევას, როცა M=2 (გვაქვს 2 საწარმო), δ_{sm} არის დამოკიდებული მხოლოდ m −ზე და რეგიონის გარე არის მიზიდვის ფუნქცია $\nu_o(t) = V^2$ (თუ გვაქვს არა 2 არამედ M საწარმო, მაშინ შეიძლება მოვიქცეთ შემდეგნაირად: ვთქვათ გვინდა შევისწავლოთ i-ური საწარმო. დანარჩენი საწარმო განვიხილოთ როგორც №2 საწარმო. №2 საწარმოს საწყისი პარამეტრები, მაგალითისათვის, შეიძლება გამოვითვალოთ შემდეგნაირად:

$$\alpha_2^* = \frac{\alpha_1+\alpha_2+ \ldots+\alpha_{i-1}+\alpha_{i+1}+ \ldots+ \alpha_m}{M-1};$$

$$V_2^* = V_1 + V_2 + \ldots + V_{i-1} + V_{i+1} + \ldots + V_m$$

$$\delta_2^* = \frac{\delta_1+\delta_2+ \ldots+ \delta_{i-1}+ \delta_{i+1}+ \ldots+ \delta_m}{M-1}$$

ეს საშუალებას მოგვცემს i-ური საწარმოს შესასწავლად გამოვიყენოთ ის შედეგები, რომლებიც მიიღება იმ შემთხვევაში, როცა რეგიონში იყო სულ 2 საწარმო).

ორი საწარმო შემთხვევაში დიფერენციალურ განტოლებათა (8) სისტემა ასე გადაიწერება:

$$\begin{cases} \frac{dI_1(t)}{dt} = (\alpha_1 - \mu \cdot h_1) \cdot I_1(t) + \frac{\mu\delta_1 \cdot I_1(t) \cdot (V_1 - I_1(t))}{V^2 + \delta_1 \cdot I_1(t) \cdot (V_1 - I_1(t)) + \delta_2 \cdot I_2(t) \cdot (V_2 - I_2(t))} \left(h_1 \cdot I_1(t) + h_2 \cdot I_2(t)\right) \\ \frac{dI_2(t)}{dt} = (\alpha_2 - \mu \cdot h_2) \cdot I_2(t) + \frac{\mu\delta_2 \cdot I_2(t) \cdot (V_2 - I_2(t))}{V^2 + \delta_1 \cdot I_1(t) \cdot (V_1 - I_1(t)) + \delta_2 \cdot I_2(t) \cdot (V_2 - I_2(t))} \left(h_1 \cdot I_1(t) + h_2 \cdot I_2(t)\right) \end{cases}$$
(9)

შევნიშნოთ, რომ თუ რომელიმე m-თვის (m=1, ან m=2) $\alpha_m \geq \mu h_m$, მაშინ (9)-სისტემას m-ის შესაბამისი განტოლების მარჯვენა მხარე არის დადებითი, ამ შემთხვევაში m-ურ

საწარმოში ინვესტიცია უსასრულოდ იზრდება, ამიტომ ჩვენთვის საინტერესოა შემთხვევა, როცა $\alpha_m < \mu h_m$.

დიფერენციალური განტოლებების (9) სისტემას ზღვრული ციკლები არ გააჩნია და (9)-ის ამონახსნები მიისწრაფვიან მუდმივი ამონახსნებისაკენ - წონასწორობის მდგომარეობისაკენ. ჩვენი მიზანია $(\delta_1 ; \delta_2)$ საკოორდინაციო სიბრტყის I მეოთხედში ვიპოვოთ არაუარყოფითი წონასწორობის მდგომარეობები. ამისათვის საჭიროა ვიპოვოთ შემდეგი განტოლებათა სისტემის ყველა არაუარყოფითი ამონახსნი:

$$\begin{cases} \dfrac{\delta_1 I_1 (V_1 - I_1)(\mu h_1 I_1 + \mu h_2 I_2)}{V^2 + \delta_1 I_1 (V_1 - I_1) + \delta_2 I_2 (V_2 - I_2)} = (\mu h_1 - \alpha_1) I_1 \\ \dfrac{\delta_2 I_2 (V_2 - I_2)(\mu h_1 I_1 + \mu h_2 I_2)}{V^2 + \delta_1 I_1 (V_1 - I_1) + \delta_2 I_2 (V_2 - I_2)} = (\mu h_2 - \alpha_2) I_2 \end{cases} \tag{10}$$

I) $I_1 = 0;\ \ I_2 = 0$ - სისტემას ტრივიალური ამონახსნი გააჩნია δ_1 და δ_2 პარამეტრების ნებისმიერი მნიშვნელობისათვის.

II) $I_1 = 0;\ \ I_2 \neq 0$. (10)-ის მეორე განტოლებიდან:

$$\frac{\delta_2 \mu h_2 I_2 (V_2 - I_2)}{V^2 + \delta_2 I_2 (V_2 - I_2)} = \mu h_2 - \alpha_2$$

თუ შემოვიღებთ აღნიშვნას $\delta_2 \cdot I_2 (V_2 - I_2) \equiv U_2$, მაშინმიღებული განტოლება ასე ჩაიწერება:

$$\mu h_2 U_2 = (\mu h_2 - \alpha_2)(V^2 + U_2)$$

$$\mu h_2 U_2 = (\mu h_2 - \alpha_2) V^2 + \mu h_2 U_2 - \alpha_2 U_2$$

$$U_2 = \frac{(\mu h_2 - \alpha_2) V^2}{\alpha_2}$$

დავუბრუნდეთ აღნიშვნას:

$$\delta_2 I_2 (V_2 - I_2) = \frac{(\mu h_2 - \alpha_2) V^2}{\alpha_2}$$

$$(I_2)^2 - V_2 I_2 + \frac{(\mu h_2 - \alpha_2) V^2}{\alpha_2 \delta_2} = 0 \tag{11}$$

რადგან ჩვენ განვიხილავთ შემთხვევას, როცა $\mu h_2 - \alpha_2 > 0$, ამიტომ იმისათვის, რომ (11) განტოლებას გააჩნდეს დადებითი ამონახსნი, აუცილებელი და საკმარისია, რომ დისკრიმინანტი იყოს არაუარყოფითი:

$$D_2 = V_2^2 - \frac{4(\mu h_2 - \alpha_2)V^2}{\alpha_2 \delta_2} \geq 0 \tag{12}$$

(12)-დან ვღებულობთ, რომ რეგიონში არსებობს $(0; I_2)$ სახისწონსაწორობის მდგომარეობა მაშინ და მხოლოდ მაშინ, როცა

$$\delta_2 \geq \frac{4(\mu h_2 - \alpha_2)V^2}{\alpha_2 V_2^2} \tag{13}$$

თუ შემოვიღებთ აღნიშვნებს:

$$K_i = \alpha_i(\mu h_i - \alpha_i), \quad i = 1; 2, \tag{14}$$

$$e_i = \frac{V_i}{V}, \ \ i = 1; 2, \tag{15}$$

მაშინ (13) პირობა ასე ჩაიწერება:

$$\delta_2 \geq \frac{4K_2}{(\alpha_2)^2 e_2^2} \tag{16}$$

(წირს, რომლის წერტილები აკმაყოფილებენ პირობებს: $\begin{cases} \delta_1 > 0 \\ \delta_2 = \frac{4K_2}{(\alpha_2)^2 e_2^2}, \end{cases}$ დავარქვათ l_1 წირი)

მაშასადამე, თუ სრულდება (16) პირობა, მაშინ რეგიონში გვაქვს$\left(0; \overset{\pm}{I}_2\right)$ სახის წონასწორობის მდგომარეობა, სადაც

$$\overset{\pm}{I}_2 = \frac{V_2}{2} \pm \frac{V}{2\alpha_2}\sqrt{(\alpha_2)^2 e_2^2 - \frac{4K_2}{\delta_2}}$$

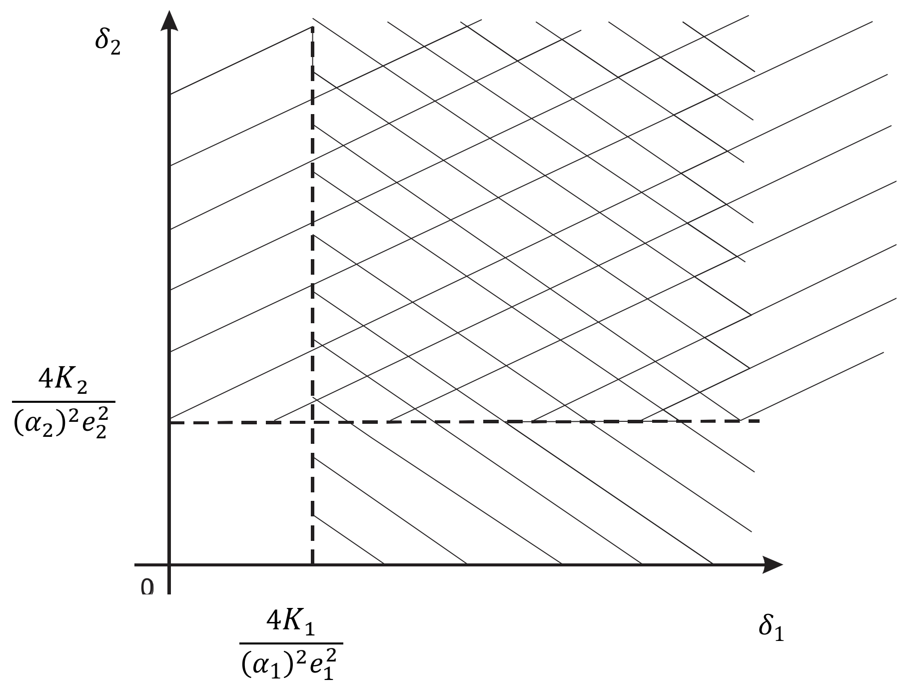
ანალოგიურად მივიღებთ, რომ თუ სრულდება პირობა

$$d_1 \geq \frac{4K_1}{(\alpha_1)^2 e_1^2} \tag{17}$$

(წირს, რომლის წერტილები აკმაყოფილებენ პირობებს: $\begin{cases} \delta_2 > 0 \\ \delta_1 = \frac{4K_2}{(\alpha_2)^2 e_2^2}, \end{cases}$ დავარქვათ l_2 წირი)

მაშინ რეგიონში გვაქვს $\left(\overset{\pm}{I}_1;\ 0\right)$ სახის წონასწორობის მდგომარეობა, სადაც

$$\overset{\pm}{I}_1 = \frac{V_1}{2} \pm \frac{V}{2\alpha_1}\sqrt{(\alpha_1)^2 e_1^2 - \frac{4K_1}{\delta_1}}$$



ნახ. 1

როგორც ზემოთ აღვნიშნეთ, თუ დისკრიმინანტი არის არაუარყოფითი, მაშინ არსებობს და ამასთან აუცილებლად $\overset{\pm}{I}_1 > 0$, ამიტომ დამატებით იმის გარკვევა, თუ კიდევ რა პირობებს უნდა აკმაყოფილებდეს δ_i პარამეტრები იმისათვის, რომ $\overset{\pm}{I}_i$ იყოს დადებითი, აღარაა საჭირო.

III) ვეძებთ რეგიონში დადებითი წონასწორობის მდგომარეობებს: რა პირობები უნდა დააკმაყოფილოს δ_1 და δ_2 პარამეტრებმა, რომ რეგიონში არსებობდეს $I_1 > 0, I_2 > 0$ წონასწორობის მდგომარეობა. ასეთ შემთხვევაში (9)-დან მივიღებთ:

$$\begin{cases} \dfrac{\delta_1(V_1 - I_1)(\mu h_1 I_1 + \mu h_2 I_2)}{V^2 + \delta_1 I_1(V_1 - I_1) + \delta_2 I_2(V_2 - I_2)} = \mu h_1 - \alpha_1 \\ \dfrac{\delta_2(V_2 - I_2)(\mu h_1 I_1 + \mu h_2 I_2)}{V^2 + \delta_1 I_1(V_1 - I_1) + \delta_2 I_2(V_2 - I_2)} = \mu h_2 - \alpha_2 \end{cases} \tag{18}$$

(18) სისტემის პირველი განტოლება ასე გადავწეროთ:

$$\frac{\frac{\delta_1(V_1-I_1)}{V(\mu h_1-\alpha_1)}\Big(\frac{\mu h_1 V_1+\mu h_2 V_2}{V}-\frac{\mu h_1(\mu h_1-\alpha_1)}{\delta_1}\cdot\frac{\delta_1(V_1-I_1)}{V(\mu h_1-\alpha_1)}-\frac{\mu h_2(\mu h_2-\alpha_2)}{\delta_2}\cdot\frac{\delta_2(V_2-I_2)}{V(\mu h_2-\alpha_2)}}{1+(\mu h_1-\alpha_1)\frac{\mu h_1(V_1-I_1)}{V(\mu h_1-\alpha_1)}\Big(\frac{V_1}{V}-\frac{(\mu h_1-\alpha_1)}{\delta_1}\cdot\frac{\delta_1(V_1-I_1)}{V(\mu h_1-\alpha_1)}\Big)+(\mu h_2-\alpha_2)\frac{\delta_2(V_2-I_2)}{V(\mu h_2-\alpha_2)}\Big(\frac{V_2}{V}-\frac{\mu h_2-\alpha_2}{\delta_2}}$$

როგორც (18)-დან ჩანს, ადგილი აქვს ტოლობას:

$$\frac{\delta_1(V_1-I_1)}{\mu h_1-\alpha_1}=\frac{\delta_2(V_2-I_2)}{\mu h_2-\alpha_2}$$

ამიტომ, თუ შემოვიღებთ აღნიშვნას

$$\frac{\delta_1(V_1-I_1)}{\mu h_1-\alpha_1}\equiv z$$

მაშინ (19)-დან მივიღებთ:

$$\frac{z\left(\frac{\mu h_1 V_1+\mu h_2 V_2}{V}-\frac{\mu h_1(\mu h_1-\alpha_1)}{\lambda_1^p}\cdot z-\frac{\mu h_2(\mu h_2-\alpha_2)}{\lambda_2^p}\cdot Z_p\right)}{1+(\mu h_1-\alpha_1)Z_p\left(\frac{V_1}{V}-\frac{(\mu h_1-\alpha_1)}{\delta_1}\cdot Z_p\right)+(\mu h_1-\alpha_2)z\left(\frac{V_2}{V}-\frac{\mu h_2-\alpha_2}{\lambda_2^p}\cdot z\right)}=1\Rightarrow$$

$$\Rightarrow-\left(\frac{\mu h_1(\mu h_1-\alpha_1)}{\delta_1}+\frac{\mu h_2(\mu h_2-\alpha_2)}{\delta_2}\right)z+\frac{\mu h_1 V_1+\mu h_2 V_2}{V}z=$$

$$=-\left(\frac{\mu h_1(\mu h_1-\alpha_1)}{\delta_1}+\frac{\mu h_2(\mu h_2-\alpha_2)}{\delta_2}\right)z^2+\left(\frac{\left(\mu h_1-d_1^p\right)V_1}{V}+\frac{\left(\mu h_2-d_2^p\right)V_2}{V}\right)z+1\Rightarrow$$

$$\Rightarrow\left(\frac{\mu h_1(\mu h_1-\alpha_1)}{\delta_1}+\frac{\mu h_2(\mu h_2-\alpha_2)}{\delta_2}\right)z^2-\left(\frac{\alpha_1 V_1}{V}+\frac{\alpha_2 V_2}{V}\right)z+1=0 \qquad (20)$$

თუ გავიხსენებთ (14) და (15) აღნიშვნებს და შემოვიტანთ კიდევ ერთ აღნიშვნას:

$$\alpha_1 e_1+\alpha_2 e_2\equiv A$$

მაშინ (20) გადაიწერება შემდეგნაირად:

$$\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)z^2-A\,z+1=0 \qquad (21)$$

იმისათვის, რომ (21) განტოლებას ჰქონდეს ამონახსნი, საჭიროა მისი დისკრიმინანტი იყოს არაუარყოფითი:

$$A^{\,2}-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)\geq 0 \qquad (22)$$

რადგან ჩვენ ვიხილავთ შემთხვევას, როცა $\delta_1 > 0$ და $\delta_2 > 0$, ამიტომ (22) უტოლობა ექვივალენტურია შემდეგი უტოლობის:

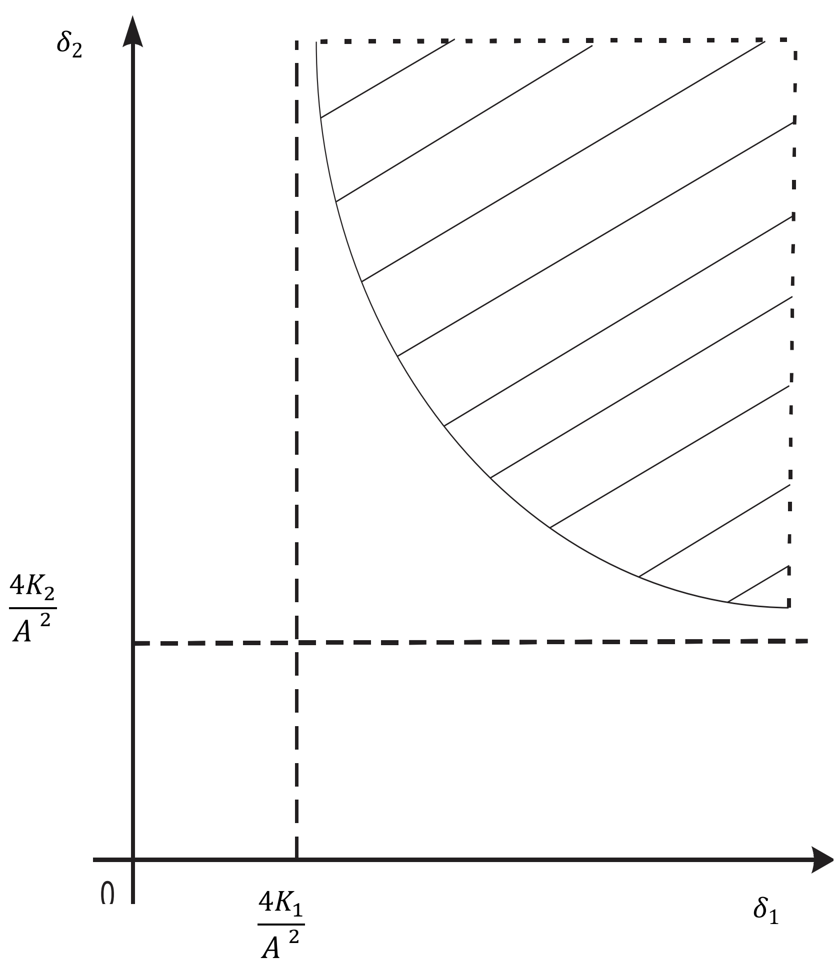
$$\delta_2 \geq \frac{4K_2}{A^{\,2}}\left(1+\frac{\frac{4K_1}{A^{\,2}}}{\delta_1-\frac{4K_1}{A^{\,2}}}\right)$$

ესაა არე, რომელიც მდებარეობს

$$\delta_2 = \frac{4K_2}{A^{\,2}}\left(1+\frac{\frac{4K_1}{A^{\,2}}}{\delta_1-\frac{4K_1}{A^{\,2}}}\right) \tag{23}$$

ტოლობით მოცემულია ჰიპერბოლის ზევით. აღნიშნული ჰიპერბოლის ასიმპტოტებია:

$$\delta_1 = \frac{4K_1}{A^{\,2}}; \quad \delta_2 = \frac{4K_2}{A^{\,2}}$$



ნახ. 2

თუ სრულდება (22) პირობა, მაშინ (21) განტოლების ამონახსნი მოიცემა ტოლობით:

$$z^{\pm}=\frac{A\ \pm\sqrt{A^{\,2}-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}}{2\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}$$

თუ გავიხსენებთ, რომ

$$\frac{\delta_1(V_1-I_1)}{\mu h_1-\alpha_1}=\frac{\delta_2(V_2-I_2)}{\mu h_2-\alpha_2}=z$$

მაშინ მივიღებთ:

$$I_i^{\pm}=V_i-V(\mu h_i-\alpha_i)\frac{A\pm\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}}{2\delta_i\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)} \qquad (24)$$

შევნიშნოთ, რომ (22) პირობა არის აუცილებელი, მაგრამ არა საკმარისი იმისათვის, რომ რეგიონში არსებობდეს დადებითი წონასწორობის მდგომარეობა: $I_1>0$, $I_2>0$. ვნახოთ დამატებით კიდევ რა პირობები უნდა დააკმაყოფილოს δ_1, δ_2 პარამეტრებმა იმისათვის, რომ $I_1^+>0, I_1^+>0$; $I_1^->0, I_2^->0$.

განვიხილოთ შემთხვევა, როცა $\alpha_1 e_1<\alpha_2 e_2$. ვნახოთ ამ შემთხვევაში გარდა (22) პირობისა კიდევ რა პირობას უნდა აკმაყოფილებდნენ δ_1 და δ_2 პარამეტრები იმისათვის, რომ I_1^+ იყოს დადებითი, ანუ შესრულდეს პირობა:

$$V_1-V(\mu h_1-\alpha_1)\frac{A+\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}}{2\delta_1\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}>0\Leftrightarrow$$

$$\Leftrightarrow\frac{2V_1}{V(\mu h_1-\alpha_1)}(K_1\delta_2+K_2\delta_2)-\delta_1 A>\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}\Leftrightarrow$$

$$\Leftrightarrow\left(\frac{2V_1K_1}{V(\mu h_1-\alpha_1)}-A\right)\delta_1+\frac{2V_1K_2}{V(\mu h_1-\alpha_1)}\delta_1>\delta_2\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}$$

$$\Leftrightarrow \left(\frac{2e_1K_1}{(\mu h_1-\alpha_1)}-A\right)\delta_2+\frac{2e_1K_2}{(\mu h_1-\alpha_1)}\delta_1>\delta_2\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}\Leftrightarrow$$

$$\Leftrightarrow (\alpha_1e_1-\alpha_2e_2)\delta_2+\frac{2\alpha_1e_1K_2}{K_1}\delta_1>\delta_2\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}$$

თუ შემოვიღებთ აღნიშვნას

$$(A)^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)\equiv D$$

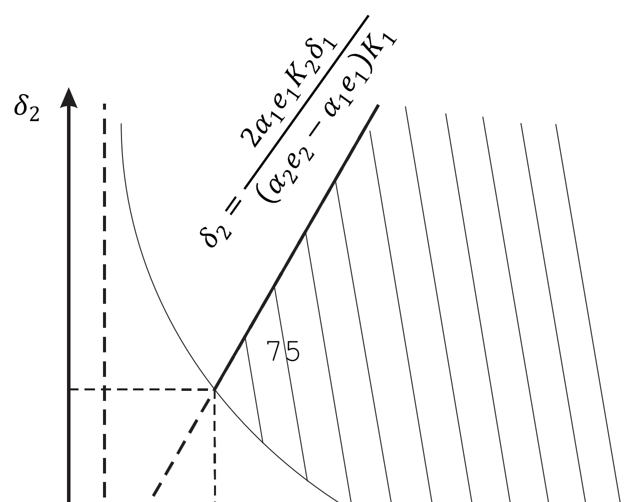
მაშინ უკანასკნელი უტოლობა შეიძლება ჩავწეროთ შემდეგი უტოლობათა სისტემის სახით:

$$\begin{cases}(\alpha_1e_1-\alpha_2e_2)\delta_1+\dfrac{2\alpha_1e_1K_2}{K_1}\delta_1\geq 0 & (25)\\ (\alpha_1e_1-\alpha_2e_2)^2(\delta_2)^2+\dfrac{4\alpha_1\alpha_1e_1(\alpha_1e_1-\alpha_2e_2)K_2^p}{K_1}\delta_1\delta_2+\dfrac{4(\alpha_1)^2{e_1}^2(K_2)^2}{(K_1)^2}(\delta_1)^2>(\delta_2)^2D & (26)\end{cases}$$

რადგან $\alpha_1e_1-\alpha_2e_2<0$, ამიტომ (25) უტოლობა ტოლფასია უტოლობის:

$$\delta_1\leq\frac{2\alpha_1e_1K_2}{(\alpha_2e_2-\alpha_1e_1)K_1}\delta_1 \qquad (27)$$

მაშასადამე, როცა $\alpha_1e_1<\alpha_2e_2$, მაშინ I_1^+-ის დადებითობისათვის აუცილებელია რომ $(\delta_1;\delta_2)$ ეკუთვნოდეს (ნახ. 3)-ზე დასტრიხულ არეს:



$$\frac{4K_1}{A(\alpha_1 e_2 - \alpha_2 e_1)}$$

$$(A)^2 = 4\left(\frac{K_1}{\delta_1} + \frac{K_2}{\delta_2}\right)$$

$$\frac{4K_1}{(A)^2}$$

$$\frac{4K_1}{(A)^2} \qquad \frac{2K_1^p}{\alpha_1 e_1 A} \qquad \delta_1$$

ვნახოთ კიდევ რა პირობა უნდა დააკმაყოფილონ $(\delta_1;\delta_2)$ პარამეტრებმა იმისათვის, რომ შესრულდეს (26) პირობა, ანუ იმისათვის რომ I_1^+ იყოს დადებითი.

(26)-დან ვღებულობთ:

$$(\alpha_1)^2(e_1)^2(\delta_2)^2 - 2\alpha_1 e_1 \alpha_2 e_2 (\delta_2)^2 + (\alpha_2)^2(e_2)^2(\delta_2)^2 + \frac{4(\alpha_1)^2(e_1)^2(K_2)^2}{K_1}\delta_1\delta_2$$
$$- \frac{4\alpha_1 e_1 \alpha_2 e_2 K_2}{K_1}\delta_1\delta_2 + \frac{4(\alpha_1)^2(e_1)^2(K_2)^2}{K_1}(\delta_1)^2 > (\delta_2)^2 A^2 - \frac{4K_1(\delta_2)^2}{\delta_1} - 4K_2\delta_2$$

$$[(\delta_1)^2(e_1)^2\delta_1 - 2\alpha_1 e_1 \alpha_2 e_2 \delta_1 + (\alpha_2)^2(e_2)^2\delta_1 - A^2\delta_1 + 4K_1](\delta_2)^2 +$$
$$+\left(\frac{4(\alpha_1)^2(e_1)^2(K_2)^2}{K_1}(\delta_1)^2 - \frac{4\alpha_1 e_1 \alpha_2 e_2 K_2}{K_1}(\delta_2)^2 + 4K_2\delta_1\right)\delta_2 + \frac{4(\alpha_1)^2(e_1)^2(K_2)^2}{(K_1)^2}(\delta_1)^3 > 0 \Leftrightarrow$$

$$\Leftrightarrow (-4\alpha_1 e_1 \alpha_2 e_2 \delta_1 + 4K_1)(\delta_2)^2 + \frac{4K_2\delta_1}{K_1}[((\delta_1)^2(e_1)^2 - \alpha_1 e_1 \alpha_2 e_2)\alpha_1 + K_1]\alpha_2 +$$
$$+\frac{4(\alpha_1)^2(e_1)^2(K_2)^2}{(K_1)^2}(\delta_1)^3 > 0 \qquad (28)$$

რადგან ჩვენ განვიხილავთ δ_1-ს იმ მნიშვნელობებს, რომლებიც მეტია ან ტოლი $\frac{2K_1}{\alpha_1 e_1 A}$-ზე, ამიტომ

$$-4\alpha_1 e_1 \alpha_2 e_2 \delta_1 + 4K_1 \le -4\alpha_1 e_1 \frac{2K_1}{\alpha_1 e_1 A} + 4K_1 =$$
$$= 4K_1\left(-\frac{2\alpha_2 e_2}{A} + 1\right) = 4K_1 \frac{\alpha_1 e_1 - \alpha_2 e_2}{A} < 0 \qquad (29)$$

ამიტომ (28) არის კვადრატული უტოლობა δ_2-ს მიმართ. თუ ამოვხსნით (28) უტოლობის შესაბამის კვადრატულ განტოლებას, ვნახავთ, რომ მისი ფესვებია

$$\overset{+}{\delta}_2 = -\frac{K_2}{\delta_1}; \quad \overset{-}{\delta}_2 = \frac{\alpha_1^2 e_1^2 K_2 (\delta_1)^2}{K_1 \left(\alpha_1^p e_1 \alpha_2^p e_2 - K_1^p\right)}$$

ამიტომ (28) უტოლობა ტოლფასია შემდეგი უტოლობის:

$$(-4\alpha_1 e_1 \alpha_2 e_2 \delta_1 + 4K_1)\left(\delta_2 + \frac{K_2 \delta_1}{K_1}\right)\left(\delta_2 - \frac{(\alpha_1)^2 (e_1)^2 (K_2)^2 (\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)}\right) > 0 \tag{30}$$

რადგან (29)-ის თანახმად $-4\alpha_1 e_1 \alpha_2 e_2 \delta_1 + 4K_1 < 0$ და $\delta_2 + \frac{K_2 \delta_1}{K_1} > 0$, ამიტომ (30) ტოლფასია უტოლობის:

$$\delta_1 < \frac{(\alpha_1)^2 (e_1)^2 (K_2)^2 (\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)} \tag{31}$$

შევისწავლოთ წირი:

$$\delta_2 = \frac{(\alpha_1)^2 (e_1)^2 K_2 (\delta_1)^2}{K_1(\alpha_1 \alpha_2 e_1 e_2 \delta_1 - K_1)} \tag{32}$$

მისი ვერტიკალური ასიმპტოტია

$$\delta_1 = \frac{K_1}{\alpha_1 \alpha_1 e_1 e_2}$$

ვიპოვოთ დახრილი ასიმპტოტი

$$k = \lim_{\delta_1 \to \infty} \frac{\delta_2}{\delta_1} = \lim_{\delta_1 \to \infty} \frac{(\alpha_1)^2 (e_1)^2 K_2 \delta_1}{K_1(\alpha_1 \alpha_2 e_1 e_2 \delta_1 - K_1)} = \frac{\alpha_1 e_1 K_2}{\alpha_2 e_2 K_1}$$

$$b = \lim_{\delta_1 \to \infty} \left(\frac{(\alpha_1)^2 (e_1)^2 K_2 \delta_1}{K_1(\alpha_1 \alpha_2 e_1 e_2 \delta_1 - K_1)} - \frac{\alpha_1 e_1 K_2}{\alpha_2 e_2 K_1}\right) =$$

$$= \frac{\alpha_1 e_1 K_2}{K_1} \lim_{\delta_2 \to \infty} \frac{\alpha_1 \alpha_2 e_1 e_2 (\delta_1)^2 - \alpha_1 \alpha_2 e_1 e_2 (\delta_1)^2 + K_1 \delta_1}{\alpha_2 e_2 (\alpha_1 \alpha_2 e_1 e_2 \delta_1 - K_1)} =$$

$$= \frac{\alpha_1 e_1 K_2}{K_1} \cdot \frac{K_1}{(\alpha_2)^2 (e_2)^2 \alpha_1 e_1} = \frac{K_2}{(\alpha_2)^2 (e_2)^2} \tag{33}$$

ადვილი შესამოწმებელია, რომ თუ $\delta_1 > \frac{K_1}{\alpha_1 \alpha_2 e_1 e_2}$, მაშინ (32) წირი მდებარეობს (33) ასიმპტოტის ზემოთ.

ვიპოვოთ (32) წირის ზრდადობისა და კლებადობის შუალედები და ექსტრემუმი:

$$(\alpha_1)' = \frac{\alpha_1 e_1 K_2}{K_1} \cdot \frac{2\delta_1(\alpha_1 \alpha_2 e_1 e_2 \delta_1 - K_1) - \alpha_1 \alpha_2 e_1 e_2 (\delta_1)^2}{(\alpha_1 \alpha_2 e_1 e_2 \delta_1 - K_1)^2} =$$

$$= \frac{\alpha_1 e_1 K_2}{K_1} \cdot \frac{(\alpha_1\alpha_2 e_1 e_2 \delta_1 - 2K_2)\delta_1}{(\alpha_1\alpha_2 e_1 e_2 \delta_1 - K_1)^2}$$

რადგან $\delta_1 > 0$, ამიტომ $\delta_2 = \delta_2(\delta_1)$ ფუნქცია ზრდადია, როცა $\delta_1 > \frac{2K_2}{\alpha_1\alpha_1 e_1 e_2}$; $\delta_2 = \delta_2(\delta_1$ ფუნქცია კლებადია, როცა $\delta_1 < \frac{K_1}{\alpha_1\alpha_2 e_1 e_2}$; $\frac{2K_2}{\alpha_1\alpha_2 e_1 e_2}$არის ფუნქციის მინიმუმის წერტილი.

$$\delta_2\left(\frac{2K_2}{\alpha_1\alpha_2 e_1 e_2}\right) = \frac{(\alpha_1)^2(e_1)^2 K_2}{K_1} \cdot \frac{4K_1}{(\alpha_1)^2(\alpha_2)^2(e_1)^2(e_2)^2} \cdot \frac{1}{\alpha_1\alpha_2 e_1 e_2 \cdot \frac{2K_2}{\alpha_1\alpha_2 e_1 e_2} - K_1} = \frac{4K_2}{(\alpha_2)^2(e_2)^2}$$

ეს ნიშნავს, რომ (32) წირი ეხება $\delta_2 = \frac{4K_2}{(\alpha_2)^2(e_2)^2}$ წრფეს – (16) ტოლობით მოცემულ წრფეს.

ვნახოთ კვეთს თუ არა ერთმანეთს (23') და (32) წირები $(\delta_1;\delta_2)$ საკოორდინატო სიბრტყის პირველ მეოთხედში. (23)-დან

$$\delta_2 = \frac{4K_2\delta_1}{A^2\delta_1 - 4K_1}$$

$$\frac{4K_2\delta_1}{(A^p)^2\delta_1 - 4K_1} = \frac{(\alpha_1)^2(e_1)^2 K_2(\delta_1)^2}{K_1(\alpha_1\alpha_2 e_1 e_2\delta_1 - K_1)}$$

$$\frac{4}{A^2\delta_1 - 4K_1} = \frac{(\alpha_1)^2(e_1)^2\delta_1}{K_1(\alpha_1\alpha_2 e_1 e_2\delta_1 - K_1)}$$

მარტივი გარდაქმნების შემდეგ მივიღებთ:

$$A^2(\alpha_1)^2(e_1)^2(\delta_1)^2 - 4\alpha_1 e_1 K_1(\alpha_1 e_1 + \alpha_2 e_2)\delta_2 + 4(K_1)^2 = 0 \qquad (34)$$

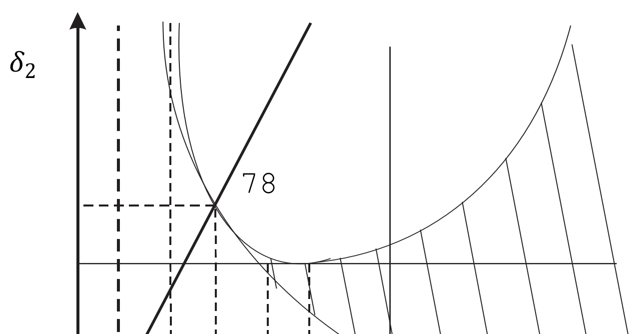
ანუ $(A\alpha_1 e_1\delta_1 - 2K_1)^2 = 0$.

ეს კი ნიშნავს, რომ (23) და (32) წირები ერთმანეთს ეხებიან წერტილში

$$\left(\frac{2K_1}{\alpha_1 e_1 A}; \frac{4K_1}{A(\alpha_2 e_2 - \alpha_1 e_1)}\right).$$

თუ გავიხსენებთ (ნახ. 3)-ს დავინახავთ, რომ (23) და (32) წირები ერთმანეთს ეხებიან წერტილში, რომელშიც $\delta_2 = \frac{2\alpha_1 e_1 K_2}{K_1(\alpha_2 e_2 - \alpha_1 e_1)}\delta_1$წრფე კვეთს (23) წირს.

ყოველივე ზემოთქმულის გათვალისწინებით მივიღებთ, რომ როცა $\alpha_1 e_1 < \alpha_2 e_2$, I_1^+ დადებითია (ნახ. 4)-ზე დაშტრიხულ სიმრავლეზე.



$\frac{4K_2}{(A)^2}$

$\frac{4K_1}{(A)^2}$ $\frac{2K_1}{\alpha_1 e_1 A}$ $\frac{2K_1}{\alpha_1 e_1 \alpha_2 e_2}$ δ_1

$\frac{4K_1}{\alpha_1 e_1 \alpha_2 e_2}$ $\frac{4K_1}{(\alpha_1)^2 {e_1}^2}$

$\frac{4K_1}{(\alpha_1)^2 {e_1}^2 - 2\,\alpha_1 e_1 \alpha_2 e_2}$

ახლა ვნახოთ, რა პირობებს უნდა აკმაყოფილებდეს δ_1 და δ_2 პარამეტრებს იმისათვის, რომ I_2^+ იყოს დადებითი (იგულისხმება, რომ $\alpha_1 e_1 < \alpha_2 e_2$):

$$I_2^+ = V_2 - V(\mu h_2 - \alpha_2)\frac{A+\sqrt{A^2-4\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)}}{2\delta_2\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)} \tag{35}$$

თუ ჩავატარებთ იმის ანალოგიურ გარდაქმნებს, რაც ჩავატარეთ I_1^+ −ის დადებითობის არის დასადგენად მივიღებთ, რომ (35) ტოლფასია უტოლობის:

$$(\alpha_2 e_2 - \alpha_1 e_1)^2(\delta_1)^2 + \frac{2\alpha_2 e_2 K_1}{K_2}\delta_2 > \delta_1\sqrt{D} \tag{36}$$

რადგან ჩვენ განვიხილავთ შემდხვევას, როცა $\alpha_1 e_1 < \alpha_2 e_2$, ამიტომ (36) უტოლდება მარცხენა მხარე $(\delta_1; \delta_2)$ საკოორდინატო სიბრტყის პირველ მეოთხედში ღებულობს მხოლოდ დადებით მნიშვნელობებს და (36) ტოლფასია უტოლობის:

$$(\alpha_2 e_2 - \alpha_1 e_1)^2(\delta_1)^2 + \frac{4\alpha_2 e_2(\alpha_2 e_2 - \alpha_1 e_1)K_1}{K_2}\delta_1\delta_2 + \frac{4(\alpha_2)^2 {e_2}^2 {K_1}^2}{(K_2)^2}(\delta_2)^2 > {\delta_1}^2 \cdot D \Leftrightarrow \tag{37}$$

$$\Leftrightarrow (-4\alpha_1 e_1 \alpha_2 e_2 \delta_2 + 4K_2)(\delta_1)^2 + \frac{4K_1\delta_2}{K_2}[((\alpha_2)^2 {e_2}^2 - \alpha_1 e_1 \alpha_2 e_2)\delta_2 + K_2]\delta_1 +$$

$$+\frac{4(\alpha_2)^2 {e_2}^2 (K_1)^2}{(K_2)^2}{\delta_2}^3 > 0 \tag{38}$$

თუ $(\delta_1)^2$-ის კოეფიციენტი არის 0-ის ტოლი, ანუ თუ

$$\delta_2 = \frac{4K_2}{\alpha_1 e_1 \alpha_2 e_2},$$

მაშინ ასეთი δ_2-თვის:

$$((\alpha_2)^2 {e_2}^2 - \alpha_1 e_1 \alpha_2 e_2)\delta_2 + K_2 = ((\alpha_2)^2 {e_2}^2 - \alpha_1 e_1 \alpha_2 e_2) \cdot \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2} + K_2 = \frac{\alpha_2 e_2}{\alpha_1 e_1} K_2 > 0$$

ამიტომ (38) უტოლობა სრულდება ყოველი დადებითი δ_1-თვის ეს ნიშნავს, რომ თუ $\delta_2 = \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}$, მაშინ I_2^+ დადებითია (23) პირობით განსაზღვრული არის ყოველ შესაბამის წერტილზე.

თუ ჩავატარებთ იმის ანალოგიურ გარდაქმნას, რაც ჩავატარეთ (28) უტოლობის ამოხსნისას, მივიღებთ, რომ (38) უტოლობა ტოლფასია შემდეგი უტოლობის:

$$(-4\alpha_1 e_1 \alpha_2 e_2 \delta_2 + 4K_2)\left(\delta_1 - \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)}\right) > 0$$

$$\Leftrightarrow \left[\begin{array}{l} \begin{cases} \delta_2 < \dfrac{K_2}{\alpha_1 e_1 \alpha_2 e_2} \\ \delta_1 > \dfrac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)} \end{cases} \\ \begin{cases} \delta_2 > \dfrac{K_2}{\alpha_1 e_1 \alpha_2 e_2} \\ \delta_1 < \dfrac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)} \end{cases} \end{array}\right. \tag{39}$$

შევისწავლოთ წირი:

$$\delta_1 = \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)} \tag{40}$$

(40) წირის ასიმპტოტებია

$$\delta_2 = \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2} \text{და} \delta_1 = \frac{\alpha_2 e_2 K_1}{\alpha_1 e_1 K_2} \delta_2 + \frac{K_1}{(\alpha_1)^2 {e_2}^2}$$

$(\delta_1; \delta_2)$საკოორდინატო სიბრტყის პირველ მეოთხედში (40) ფუნქციის მინიმუმი სწერტილია:

$$\left(\frac{4K_1}{(\alpha_1)^2 {e_1}^2}; \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}\right)$$

ვნახოთ აქვსთ თუ არა (23) და (40) წირებს საერთო წერტილი $(\delta_1;\delta_2)$საკოორდინატო სიბრტყის პირველ მეოთხედში:

(23)-დან:

$$\delta_1=\frac{4K_1}{(A^p)^2\delta_2-4K_2}$$

$$\frac{4K_1}{(A^p)^2\delta_2-4K_2}=\frac{(\alpha_2)^2{e_2}^2K_1(\delta_2)^2}{K_2(\alpha_1e_1\alpha_2e_2\delta_2-K_2)}$$

მარტივი გარდაქმნის შემდეგ მივიღებთ:

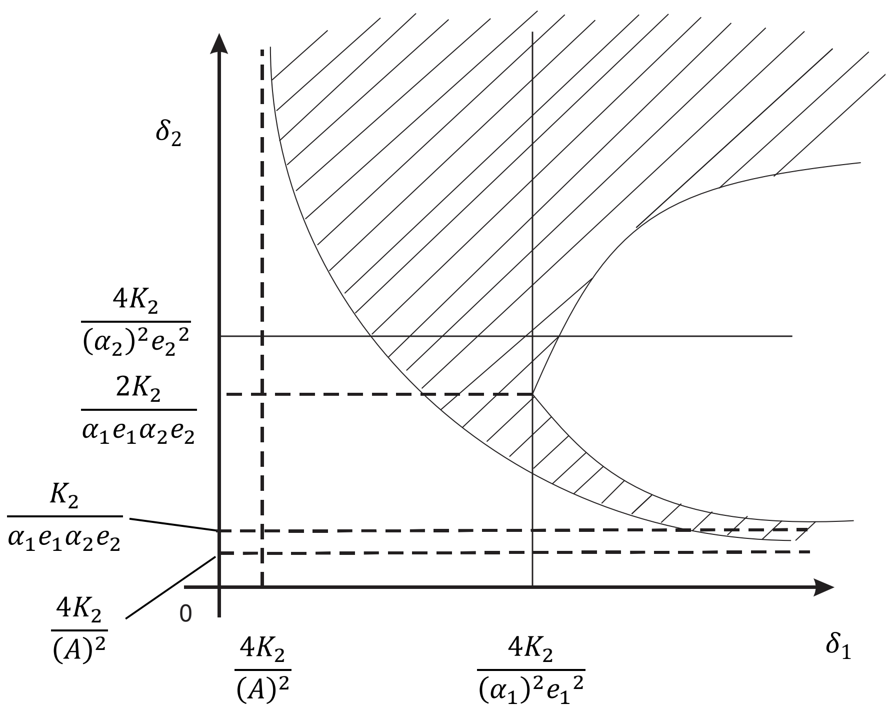
$$A^p\alpha_2e_2\delta_2-2K_2=0$$

ეს ნიშნავს, რომ (23) და (40) წირების საერთო წერტილია

$$\left(\frac{4K_1}{A(\alpha_1e_1-\alpha_2e_2)};\frac{2K_2}{\alpha_2e_2A}\right)$$

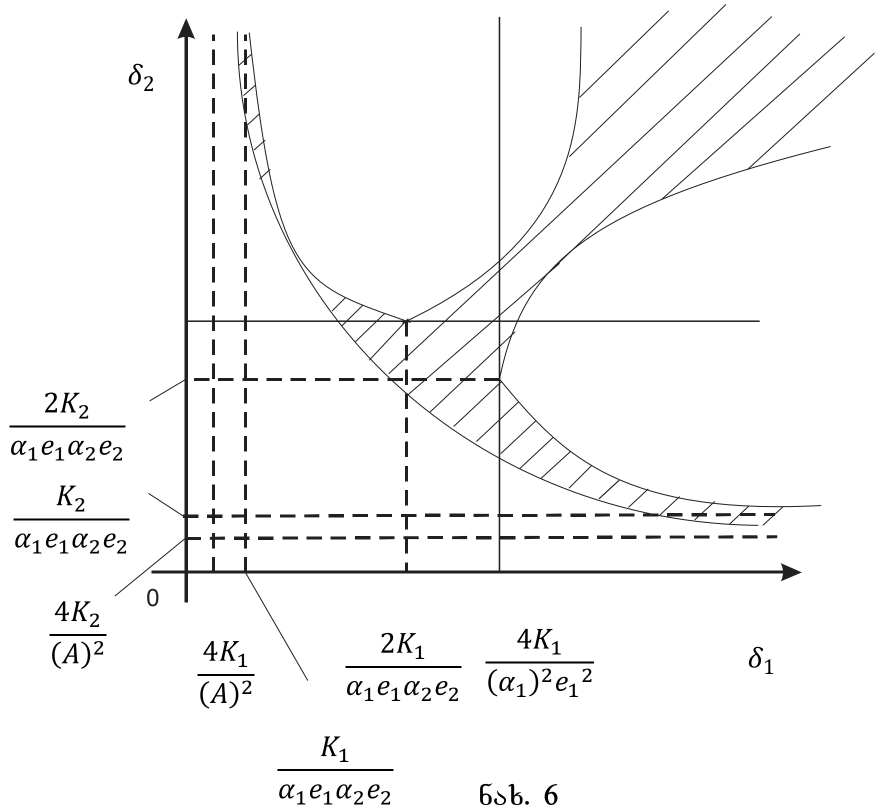
რადგან ჩვენ განვიხილავთ შემთხვევას, როცა $\alpha_1e_1<\alpha_2e_2$, ამიტომ (23') და (40) წირებს $(\delta_1;\delta_2)$საკოორდინატო სიბრტყის პირველ მეოთხედში საერთო წერტილი არ აქვთ.

ყოველივე ზემოთქმულის გათვალისწინებით მივიღებთ, რომ I_2^+ დადებითია $(\delta_1;\delta_2)$საკოორდინატო სიბრტყის პირველ მეოთხედში იმ წერტილებზე, რომლებიც დაშტრიხულია ნახ. 5-ზე.



ნახ. 5

მაშასადამე, I_1^+ და I_2^+ სიდიდეები ორივე ერთად უარყოფითია ნახ. 6-ზე დაშტრიხულ არეში.



ნახ. 6

ახლა ვნახოთ რა პირობებს უნდა აკმაყოფილებდნენ $(\delta_1; \delta_2)$პარამეტრები იმისათვის, რომ $\bar{I_1}$ იყოს დადებითი (იგულისხმება, რომ განვიხილავთ შემთხვევას, როცა $\alpha_1 e_1 < \alpha_2 e_2$).

თუ გავიხსენებთ იმ აღნიშვნებსა და გარდაქმნებს, რომლებიც გამოვიყენეთ I_1^+-ს დადებითობის პირობის მისაღებად.მივიღებთ, რომ I_1^- დადებითია, როცა სრულდება პირობა:

$$(\alpha_1 e_1 - \alpha_2 e_2)\delta_1 + \frac{2\alpha_1 e_1 K_2}{K_1}\delta_1 > \delta_2\sqrt{D} \Leftrightarrow$$

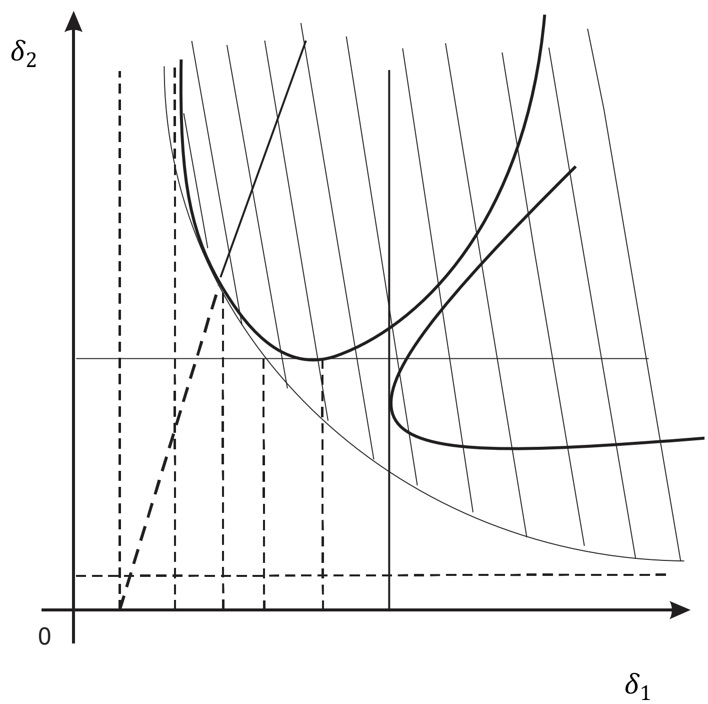
$$\Leftrightarrow \left[\begin{array}{l} \begin{cases} \delta_2 < \dfrac{2\alpha_1 e_1 K_2}{(\alpha_1 e_1 - \alpha_2 e_2)}\delta_1 \\ \delta_1 > 0, \quad \delta_2 > 0 \end{cases} \\ \begin{cases} \delta_2 > \dfrac{2\alpha_1 e_1 K_2}{K_1(\alpha_2 e_2 - \alpha_1\ e_1)}\delta_1 \\ \delta_1 > \dfrac{(\alpha_1)^2 {e_1}^2 K_2 (\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)} \end{cases} \end{array}\right.$$

რაც შეეხება I_2^--ს, ის დადებითია, როცა სრულდება პირობა:

$$(\alpha_2 e_2 - \alpha_1 e_1)\delta_1 + \frac{2\alpha_2 e_2 K_1}{K_2^p}\delta_1 > -\delta_1\sqrt{D} \tag{41}$$

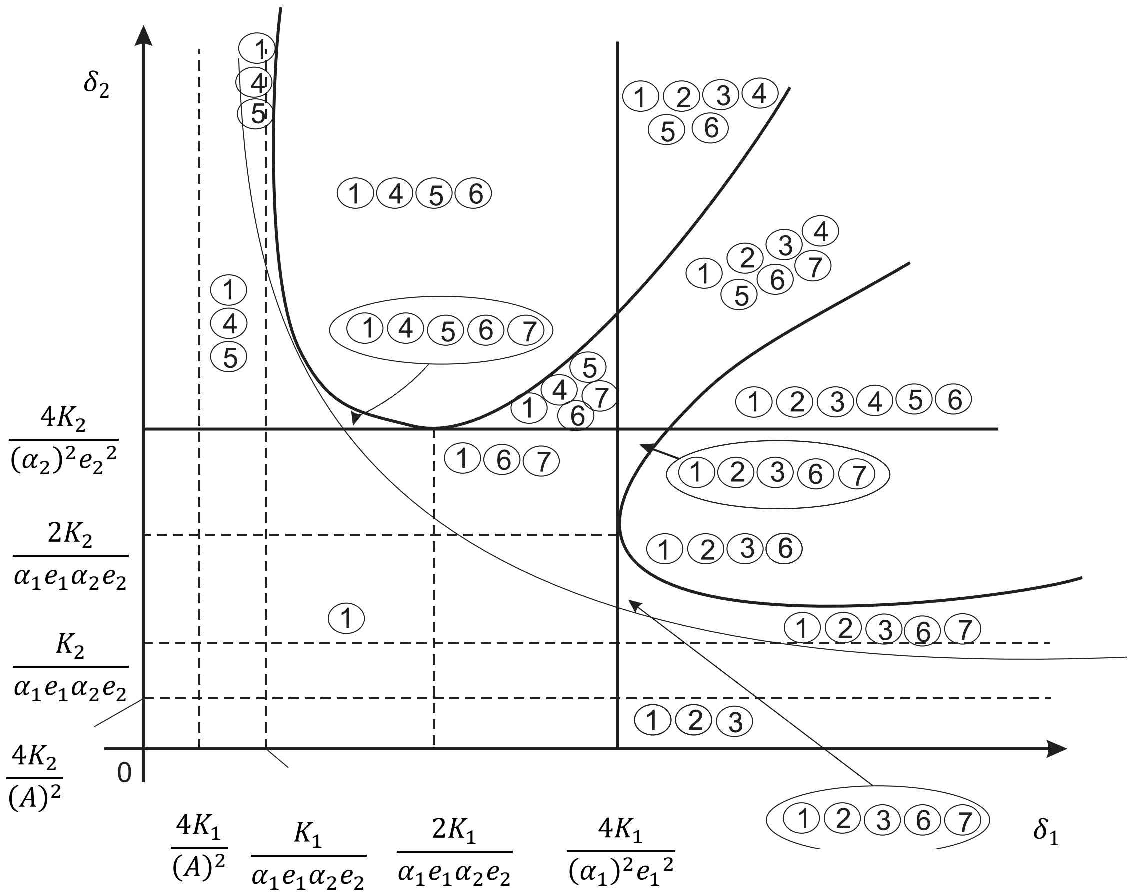
რადგან $\alpha_2 e_2 - \alpha_1 e_1 > 0$, ამიტომ (41) უტოლობა სრულდება $(\delta_1; \delta_2)$საკოორდინატო სიბრტყის პირველი მეოთხედის ყოველ წერტილზე.

მაშასადამე, თუ $\alpha_2 e_2 > \alpha_1 e_1$, მაშინ I_1^- და I_2^- სიდიდეები ორივე ერთად არაუარყოფითია ნახ. 7-ზე დაშტრიხულ არეში:



ნახ. 7

თუ ჩვენ 1-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის (0;0) მდგომარეობა, 2-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის $\left(\overset{-}{\widetilde{I}}_1; 0\right)$მდგომარეობა, 3-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის $\left(\overset{+}{\widetilde{I}}_1; 0\right)$მდგომარეობა, 4-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის $\left(0; \overset{-}{\widetilde{I}}_2\right)$ მდგომარეობა, 5-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის $\left(0; \overset{+}{\widetilde{I}}_2\right)$ მდგომარეობა, 6-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის $(I_1^-; I_2^-)$ მდგომარეობა, 7-ს ჩავწერთ იმ არეში, რომელშიც არსებობს წონასწორობის $(I_1^+; I_2^+)$ მდგომარეობა, მაშინ, თუ $\alpha_2 e_2 > \alpha_1 e_1$, მივიღებთ ნახ. 8-ზე მოცემულ სურათს:



ნახ. 8

ახლა განვიხილოთ შემთხვევა, როცა $\alpha_2 e_2 < \alpha_1 e_1$. იმისათვის, რომ I_1^+ იყოს დადებითი, უნდა შესრულდეს პირობა:

$$V_1 - V(\mu h_1 - \alpha_1)\frac{A+\sqrt{D}}{2\delta_1\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)} > 0 \;\Leftrightarrow\; (\alpha_1 e_1 - \alpha_2 e_2)\delta_1 + \frac{2\alpha_1 e_1 K_2}{K_1}\delta_1 > \delta_2 \qquad (42)$$

რადგან $\alpha_1 e_1 - \alpha_2 e_2 > 0$, ამიტომ (42) უტოლობის მარცხება მხარეში მდგარი გამოსახულება $(\delta_1;\delta_2)$საკოორდინატო სიბრტყის პირველ მეოთხედში ღებულობს დადებით მნიშვნელობას და (42) უტოლობა ტოლფასია უტოლობის:

$$(\alpha_1 e_1 - \alpha_2 e_2)^2(\delta_2)^2 + \frac{4\alpha_1 e_1 K_2(\alpha_1 e_1 - \alpha_2 e_2)}{K_1}\delta_1\delta_2 + \frac{4(\alpha_1)^2{e_1}^2(K_2)^2}{(K_1)^2}(\delta_1)^2 > (\delta_2)^2\cdot D \Leftrightarrow$$

$$\Leftrightarrow (-4\alpha_1 e_1 \alpha_2 e_2 \delta_1 + 4K_1)(\delta_2)^2 + \frac{4K_2\delta_1}{K_1}[((\alpha_1)^2{e_1}^2 - \alpha_1 e_1 \alpha_2 e_2)\delta_1 + K_1]\delta_2 +$$

$$+\frac{4(\alpha_1)^2{e_1}^2(K_2)^2}{(K_1)^2}(\delta_1)^3 > 0 \qquad (43)$$

თუ $(\delta_2)^2$-ის კოეფიციენტი არის 0-ის ტოლი, ანუ თუ

$$\delta_1 = \frac{K_1}{\alpha_1 e_1 \alpha_2 e_2}$$

მაშინ ასეთი δ_1-თვის:

$$((\alpha_1)^2{e_1}^2 - \alpha_1 e_1 \alpha_2 e_2)\delta_1 + K_1 = \alpha_1 e_1(\alpha_1 e_1 - \alpha_2 e_2)\cdot\frac{K_1}{\alpha_1 e_1 \alpha_2 e_2} + K_1 = \frac{\alpha_1 e_1 K_1}{\alpha_2 e_2} > 0$$

ამიტომ (43) უტოლობა სრულდება ყოველი დადებითი δ_2-თვის. ეს ნიშნავს, რომ თუ

$$\delta_1 = \frac{K_1}{\alpha_1 e_1 \alpha_2 e_2}$$

მაშინ I_1^+ დადებითია (23) პირობით განსაზღვრული არის ყოველ შესაბამის წერტილზე.

თუ

$$\delta_1 \neq \frac{K_1}{\alpha_1 e_1 \alpha_2 e_2}$$

და ჩავატარებთ იმის ანალოგიურ გარდაქმნებს, რაც ჩავატარეთ (28) უტოლობის ამოხსნისას, მივიღებთ, რომ (43) უტოლობა ტოლფასია უტოლობის:

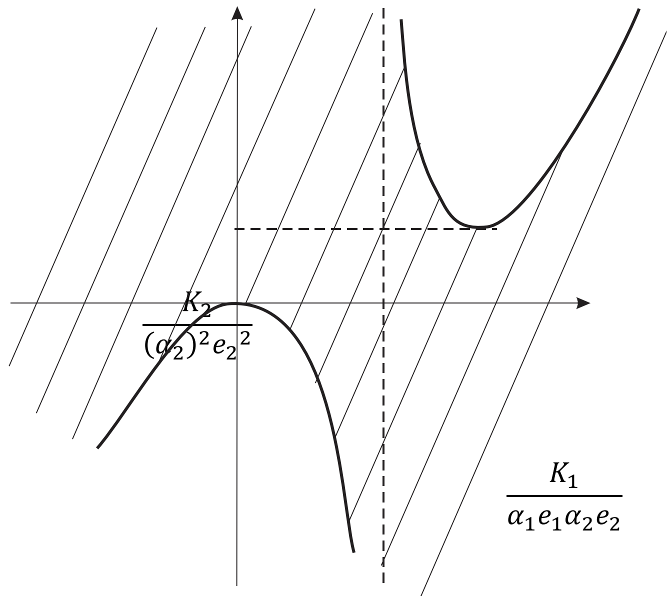
$$(-4\alpha_1 e_1 \alpha_2 e_2 \delta_1 + 4K_1)\left(\delta_2 - \frac{4(\alpha_1)^2(e_1)^2 K_2(\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)}\right) > 0 \Leftrightarrow$$

$$\Leftrightarrow \left[\begin{array}{l} \begin{cases} \delta_1 < \dfrac{K_1}{\alpha_1 e_1 \alpha_2 e_2} \\ \delta_2 > \dfrac{(\alpha_1)^2{e_1}^2 K_2(\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)} \end{cases} \\ \begin{cases} \delta_1 > \dfrac{K_1}{\alpha_1 e_1 \alpha_2 e_2} \\ \delta_2 < \dfrac{(\alpha_1)^2{e_1}^2 K_2(\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)} \end{cases} \end{array}\right. \qquad (44)$$

თუ გავიხსენებთ

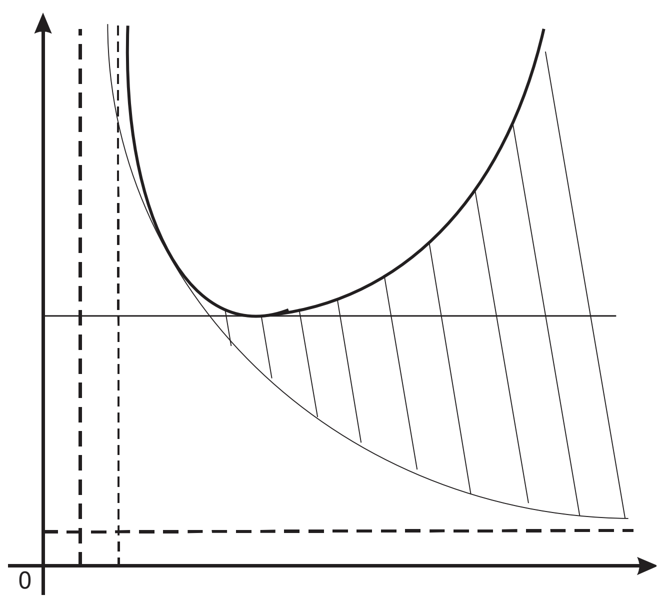
$$\delta_2 = \frac{(\alpha_1)^2{e_1}^2 K_2(\delta_1)^2}{K_1(\alpha_1 e_1 \alpha_2 e_2 \delta_1 - K_1)}$$

წირის გრაფიკს, მივიღებთ, რომ (44) სრულდება ნახ. 9-ზე მოცემულ დაშტრიხულ არეში:



ნახ. 9

თუ გავიხსენებთ, რომ I_1^+-ის დადებითობისათვის აუცილებელია, რომ შესრულდეს (23) პირობა და ამასთან $\delta_1 > 0$, $\delta_2 > 0$, მივიღებთ, რომ თუ $\alpha_1 e_1 > \alpha_2 e_2$, მაშინ $I_1^+ > 0$, როცა $(\delta_1; \delta_2)$ ეკუთვნის ნახ. 10-ზე დაშტრიხულ არეს (როცა $\alpha_1 e_1 > \alpha_2 e_2$, მაშინ (23) და (32) წირები $(\delta_1; \delta_2)$ საკოორდინატო სიბრტყის პირველ მეოთხედში ერთმანეთს არ ეხებიან).



ნახ. 10

ახლა ვნახოთ, თუ რა სიმრავლეზეა დადებითი I_2^+ იმ შემთხვევაში, როცა $\alpha_1 e_1 > \alpha_2 e_2$— ამოვხსნათ (36) უტოლობა იმ შემთხვევაში, როცა $\alpha_1 e_1 > \alpha_2 e_2$. ამ შემთხვევაში (36) უტოლობა ტოლფასია უტოლობათა სისტემის:

$$\begin{cases} (\alpha_2 e_2 - \alpha_1 e_1)\delta_1 + \dfrac{2\alpha_2 e_2 K_1}{K_2}\delta_2 & (45.1) \\ ((\alpha_2 e_2 - \alpha_1 e_1))^2 (\delta_1)^2 + \dfrac{4\alpha_2 e_2 K_2((\alpha_2 e_2 - \alpha_1 e_1))}{K_2}\delta_1\delta_2 + \dfrac{4(\alpha_2)^2 {e_2}^2 (K_1)^2}{(K_2)^2}(\delta_2)^2 > (\delta_1)^2 \cdot D & (45.2) \end{cases}$$

(45.1) ტოლფასია უტოლობის:

$$\delta_1 \le \frac{2\alpha_2 e_2 K_1}{(\alpha_1 e_1 - \alpha_2 e_2)K_2}\delta_2$$

ვიპოვოთ $\delta_1 = \frac{2\alpha_2 e_2 K_1}{(\alpha_2 e_2 - \alpha_1 e_1)\delta_1}\delta_2$ წირისა და $A^2 - 4\left(\frac{K_1}{\delta_1} + \frac{K_2}{\delta_2}\right)$ წირების თანაკვეთის წერტილი

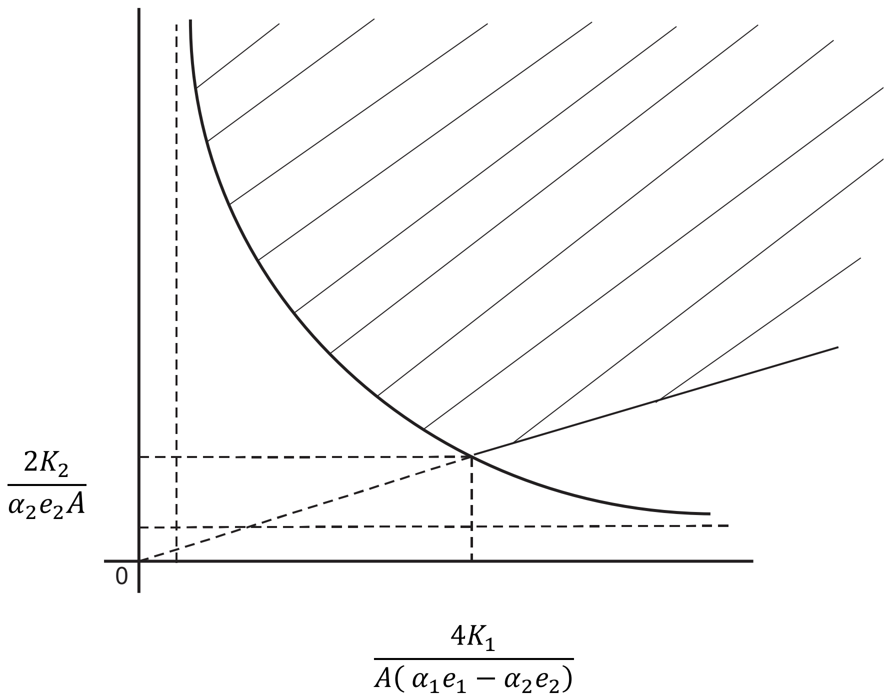
$$A^2 - 4\left(\frac{(\alpha_1 e_1 - \alpha_2 e_2)K_2}{2\alpha_2 e_2 \delta_2} + \frac{K_2}{\delta_2}\right) = 0$$

$$A^2 - 4 \cdot \frac{K_2(\alpha_1 e_1 - \alpha_2 e_2 + 2\alpha_2 e_2)}{2\alpha_2 e_2 \delta_2} = 0$$

$$\delta_2 = \frac{2K_2}{\alpha_2 e_2 A}$$

$$\delta_1 = \frac{2\alpha_2 e_2 K_1}{(\alpha_1 e_1 - \alpha_2 e_2)K_2}; \qquad \delta_2 = \frac{2\alpha_2 e_2 K_1}{(\alpha_1 e_1 - \alpha_2 e_2)K_2} \cdot \frac{2K_2}{\alpha_2 e_2 A} = \frac{4K_1}{A(\alpha_1 e_1 - \alpha_2 e_2)}$$

მაშასადამე, იმისათვის, რომ I_2^+ იყოს დადებითი, აუცილებელია $(\delta_1; \delta_2)$ ეკუთვნოდეს ნახ. 11-ზე დაშტრიხულ არეს.



ნახ. 11

ახლა ვნახოთ, რა პირობები უნდა დააკმაყოფილოს δ_1 და δ_2 პარამეტრებმა იმისათვის, რომ შესრულდეს (45.2) პირობა. თუ ჩავატარებთ იგივე გარდაქმნებს, რომელთა საშუალებითაც მივიღეთ, რომ (36) უტოლობა ტოლფასია (38)-ის, გვექნება, რომ (45.2) ტოლფასია უტოლობის:

$$(-4\alpha_1 e_1 \alpha_2 e_2 \delta_2 + 4K_2)(\delta_1)^2 + \frac{4K_1\delta_2}{K_2^p}[((\alpha_2)^2 {e_2}^2 - \alpha_1 e_1 \alpha_2 e_2)\delta_1 + K_2]\delta_1 +$$

$$+\frac{4(\alpha_2)^2 {e_2}^2 (K_1)^2}{(K_2)^2}(\delta_2)^3 > 0 \qquad (46)$$

რადგან ჩვენ განვიხილავთ δ_2 −ის იმ მნიშვნელობებს, რომლებიც მეტია ან ტოლი $\frac{2K_2}{\alpha_2 e_2 A}$, ამიტომ:

$$-4\alpha_1 e_1 \alpha_2 e_2 \delta_2 + 4K_2 \le -4\alpha_1 e_1 \alpha_2 e_2 \cdot \frac{2K_2}{\alpha_2 e_2 A} + 4K_2 = 4K_2\left(-\frac{2\alpha_1 e_1}{A} + 1\right) = 4K_2 \frac{\alpha_2 e_2 - \alpha_1 e_1}{A} < 0$$

ამიტომ (46) არის კვადრატული უტოლობა δ_1-ს მიმართ. შესაბამისი კვადრატული განტოლების ფესვებია

$$\delta_1^+ = -\frac{K_!\delta_2}{K_2}; \quad \delta_1^- = \frac{(\alpha_2)^2(e_2)^2 K_1(\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)}$$

ამიტომ (46) ტოლფასია უტოლობის:

$$(-4\alpha_1 e_1 \alpha_2 e_2 \delta_2 + 4K_2)\left(\delta_1 + \frac{4K_1\delta_2}{K_2}\right)\left(\delta_1 - \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)}\right) > 0 \qquad (47)$$

რადგან $-4\alpha_1 e_1 \alpha_2 e_2 \delta_2 + 4K_2 < 0$, $\delta_1 + \frac{4K_1\delta_2}{K_2} > 0$, ამიტომ (47) უტოლობა ტოლფასია უტოლობის:

$$\delta_1 - \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)} < 0 \qquad (48)$$

წირი$\delta_1 = \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)}$ იგივეა, რაც (40) ტოლობით მოცემული წირი, ამიტომ როგორც ვნახეთ, მისი ასიმპტოტებია

$$\delta_2 = \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}; \quad \delta_1 = \frac{\alpha_2 e_2 K_1}{\alpha_1 e_1 K_2}\delta_2 + \frac{K_1}{(\alpha_1)^2 {e_1}^2}$$

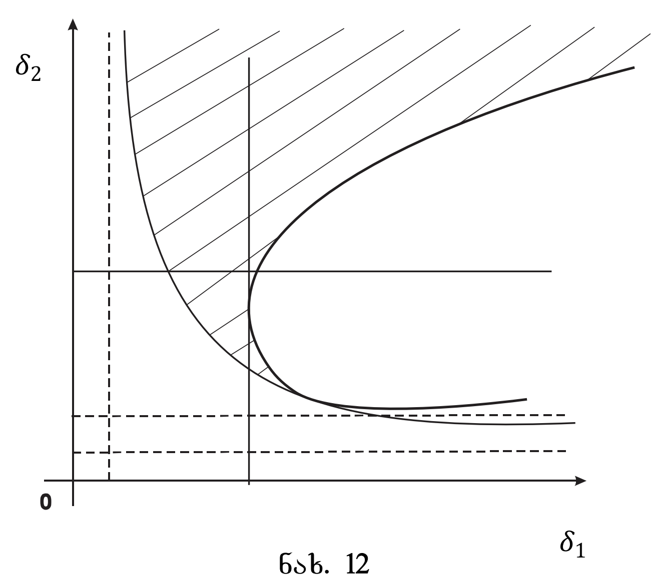
$(\delta_1;\delta_2)$საკოორდინატო სიბრტყის პირველ მეოთხედში მისი მინიმუმის წერტილია:

$$\left(\frac{4K_1}{(\alpha_1)^2 {e_1}^2};\frac{2K_2}{\alpha_1 e_1 \alpha_2 e_2}\right)$$

ამ წირს (23) წირთან შეხების წერტილია:

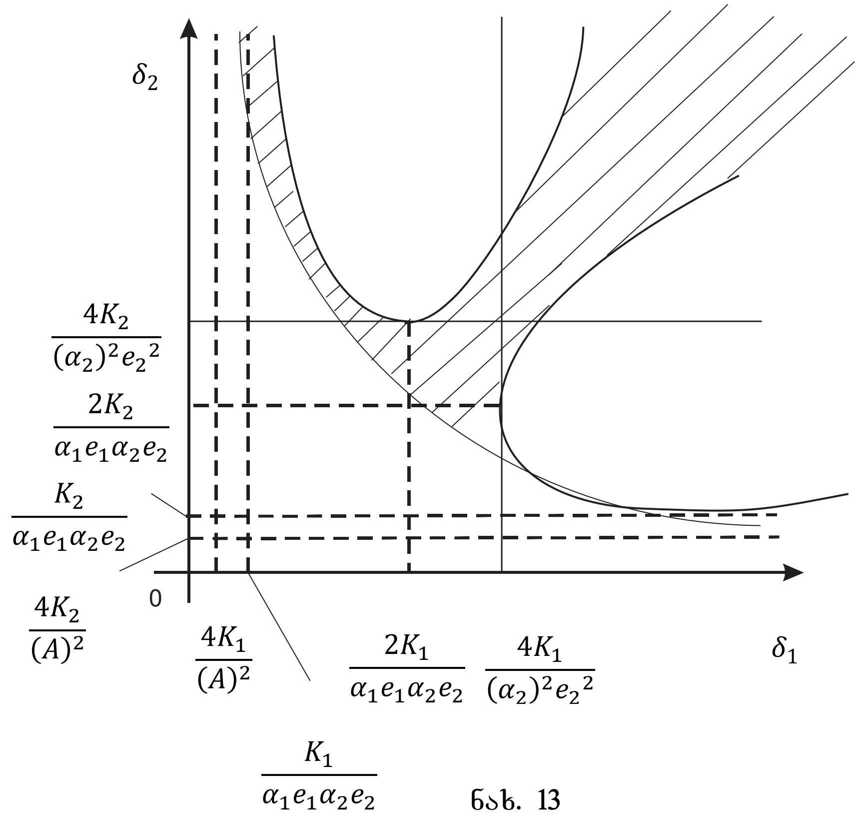
$$\left(\frac{4K_1}{A(\alpha_1 e_1 - \alpha_2 e_2)};\frac{2K_2}{\alpha_2 e_2 A}\right)$$

ყოველივე ზემოთქმულს თუ გავითვალისწინებთ მივიღებთ, რომ I_2^+ დადებითია $(\delta_1;\delta_2)$ საკოორდინატო სიბრტყის პირველ მეოთხედში იმ წერტილებზე, რომლებიც დაშტრიხულია ნახ. 12-ზე.



ნახ. 12

მაშასადამე, თუ $\alpha_1 e_1 > \alpha_2 e_2$, მაშინ I_1^+ და I_2^+ სიდიდეები ორივე ერთად არაუარყოფითია ნახ. 13-ზე დაშტრიხულ არეში:



ნახ. 13

ახლა ვნახო, რა პირობები უნდა დააკმაყოფილონ δ_1 და δ_2 პარამეტრებმა იმისათვის, რომ I_1^- იყოს დადებითი (იგულისხმება, რომ $\alpha_1 e_1 > \alpha_2 e_2$):

$$(\alpha_1 e_1 - \alpha_2 e_2)\delta_2 + \frac{2\alpha_1 e_1 K_2}{K_1}\delta_1 > -\delta_2\sqrt{D} \tag{49}$$

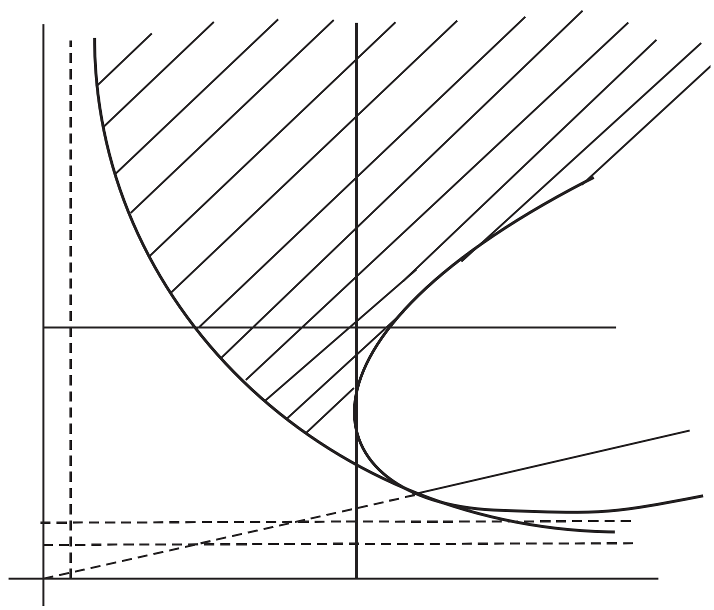
რადგან $\alpha_1 e_1 > \alpha_2 e_2$, ამიტომ (49) პირობა სრულდება $(\delta_1; \delta_2)$ საკოორდინატო სიბრტყის პირველი მეოთხედის ყოველ წერტილზე.

ახლა ვნახოთ I_2^--ს დადებითობის პირობა:

$$(\alpha_2 e_2 - \alpha_1 e_1)\delta_1 + \frac{2\alpha_2 e_2 K_1}{K_2^p}\delta_2 > -\delta_1\sqrt{D} \Leftrightarrow$$

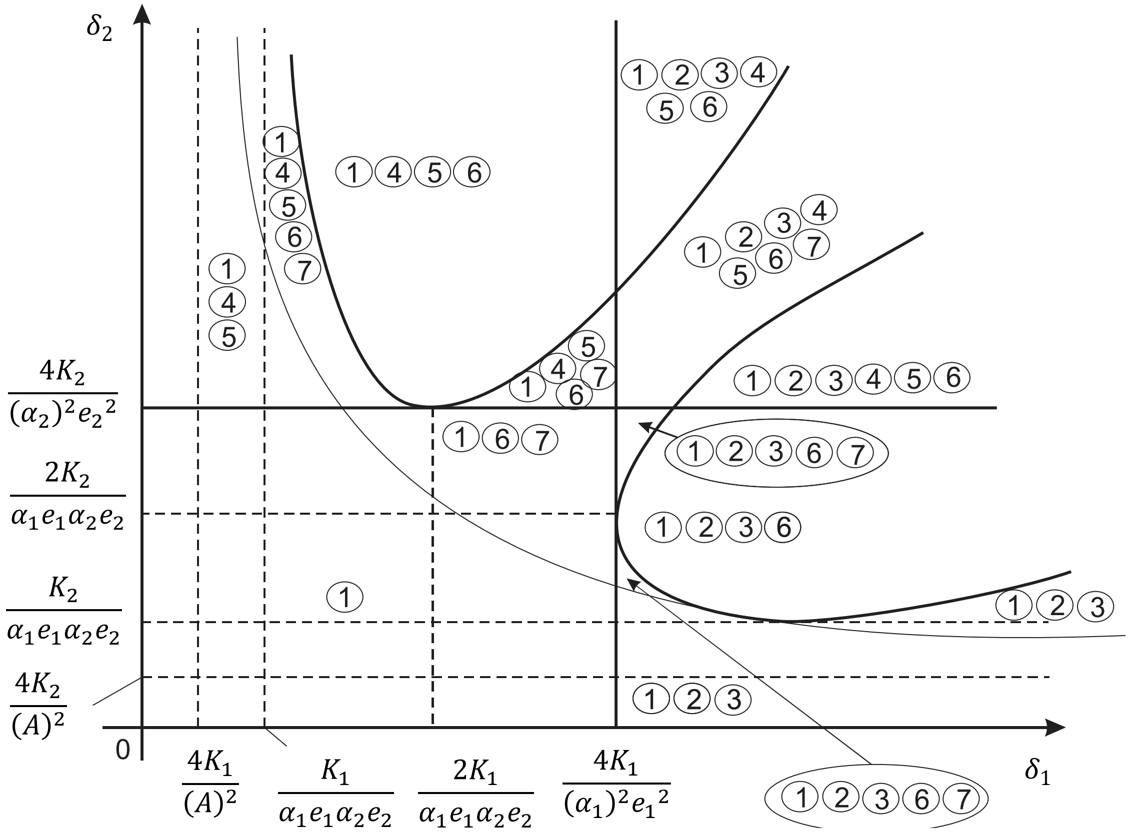
$$\Leftrightarrow \left[\begin{array}{l} \begin{cases} \delta_1 < \dfrac{2\alpha_2 e_2 K_1}{K_2(\alpha_1 e_1 - \alpha_2 e_2)} \delta_2 \\ \delta_1 > 0, \quad \delta_2 > 0 \end{cases} \\ \begin{cases} \delta_1 > \dfrac{2\alpha_2 e_2 K_1}{K_2(\alpha_1 e_1 - \alpha_2 e_2)} \delta_2 \\ \delta_1 > \dfrac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)} \end{cases} \end{array} \right.$$

მაშასადამე, თუ $\alpha_1 e_1 > \alpha_2 e_2$, მაშინ I_1^- და I_2^- სიდიდეები ორივე ერთად დადებითია ნახ. 14-ზე დაშტრიხულ არეზე.



ნახ. 14

თუ გამოვიყენებთ იმავე აღნიშვნებს, რომლებიც გამოვიყენეთ $(\delta_1; \delta_2)$ საკოორდინატო სიბრტყის პირველი მეოთხედის არეებში წონასწორობის მდგომარეობის აღსანიშნავად იმ შემთხვევაში, როცა $\alpha_1 e_1 < \alpha_2 e_2$, მაშინ $\alpha_1 e_1 > \alpha_2 e_2$ შემთხვევაში გვექნება ნახ. 15-ზე მოცემული სურათი:



ნახ. 15

განვიხილოთ შემთხვევა, როცა $\alpha_1 e_1 = \alpha_2 e_2$. ამოვხსნათ$I_1^+ > 0$ უტოლობა ამ შემთხვევაში.

$$V_1 - V(\mu h_1 - \alpha_1)\frac{A+\sqrt{D}}{2\delta_1\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)} > 0$$

$$\frac{2\alpha_1 e_1}{K_1}(K_1\delta_1 + K_2\delta_2) - A\delta_2 > \delta_2\sqrt{D} \Leftrightarrow$$

$$\Leftrightarrow (2\alpha_1 e_1 - A^p)\delta_2 + \frac{2\alpha_1 e_1 K_2}{K_1}\delta_1 > \delta_2\sqrt{D} \Leftrightarrow$$

$$\Leftrightarrow \frac{2\alpha_1 e_1 K_2}{K_1}\delta_2 > \delta_2\sqrt{D} \Leftrightarrow$$

$$\Leftrightarrow \frac{4(\alpha_1)^2 e_1{}^2 (K_2)^2}{(K_1)^2}(\delta_1)^2 > (\delta_2)^2\left((A^p)^2 - \frac{4K_1\delta_1 + 4K_2\delta_2}{\delta_1\delta_2}\right) \Leftrightarrow$$

$$\Leftrightarrow \frac{4(\alpha_1)^2 e_1{}^2 (K_2)^2}{(K_1)^2}(\delta_1)^3 > A^2(\delta_2)^2\delta_1 - 4K_1(\delta_2)^2 - 4K_2\delta_1\delta_2 \Leftrightarrow$$

$$\Leftrightarrow (-A^2\delta_1 + 4K_1)(\delta_2)^2 + 4K_2\delta_1\delta_2 + \frac{4(\alpha_1)^2 {e_1}^2 (K_2)^2}{(K_1)^2}(\delta_1)^3 > 0 \tag{50}$$

(22) პირობის ძალით, იმისათვის, რომ არსებობდეს I_1^+ აუცილებელია რომ $\delta_1 > \frac{4K_1}{A^2}$ (იხ. ნახ. 2), ამიტომ $-A^2\delta_1 + 4K_1 < 0$ და (50) არის კვადრატული უტოლობა δ_2-ს მიმართ. (50) უტოლობის შესაბამისი კვადრატული განტოლების ფესვებია:

$\delta_2^+ = -\frac{K_2\delta_1}{K_1}$; $\delta_2^- = \frac{(\alpha_1)^2(e_1)^2K_2(\delta_1)^2}{K_1((\alpha_1)^2(e_1)^2\delta_1 - K_1)}$

ამიტომ (50) უტოლობა ტოლფასია უტოლობის:

$$(-4(\alpha_1)^2{e_1}^2 + 4K_1)\left(\delta_2 + \frac{K_2\delta_1}{K_1}\right)\left(\delta_2 - \frac{(\alpha_1)^2{e_1}^2K_2(\delta_1)^2}{K_1((\alpha_1)^2{e_1}^2\delta_1 - K_1)}\right) > 0 \Leftrightarrow$$

$$\Leftrightarrow \delta_2 < \frac{(\alpha_1)^2{e_1}^2K_2(\delta_1)^2}{K_1((\alpha_1)^2{e_1}^2\delta_1 - K_1)} \tag{51}$$

შევისწავლოთ წირი

$$\delta_2 = \frac{(\alpha_1)^2{e_1}^2K_2(\delta_1)^2}{K_1((\alpha_1)^2{e_1}^2\delta_1 - K_1)} \tag{52}$$

მისი ვერტიკალური ასიმპტოტია: $\delta_1 = \frac{K_1}{(\alpha_1)^2{e_1}^2}$.

როგორც ვხედავთ, როცა $\alpha_1 e_1 = \alpha_2 e_2$, მაშინ (52) წირის ვერტიკალური ასიმპტოტი იგივეა, რაც (23) წირის ვერტიკალური ასიმპტოტი.

ვიპოვოთ დახრილი ასიმპტოტი:

$$k = \lim_{\delta_1\to\infty}\frac{\delta_2}{\delta_1} = \lim_{\delta_1\to\infty}\frac{(\alpha_1)^2{e_1}^2K_2\delta_1}{K_1((\alpha_1)^2{e_1}^2\delta_1 - K_1)} = \frac{K_2}{K_1}$$

$$b = \lim_{\delta_1\to\infty}(\delta_2 - k\delta_1) = \lim_{\delta_1\to\infty}\left(\frac{(\alpha_1)^2{e_1}^2K_2\delta_1}{K_1((\alpha_1)^2{e_1}^2\delta_1 - K_1)} - \frac{K_2}{K_1}\delta_1\right) =$$

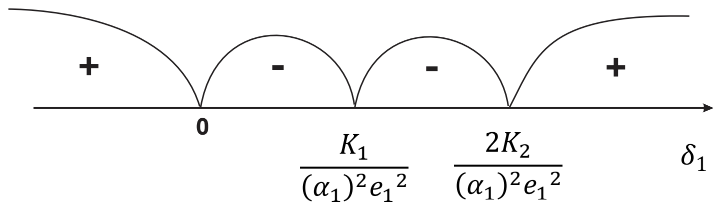
$$= \frac{K_2}{K_1}\lim_{\delta_1\to\infty}\left(\frac{(\alpha_1)^2{e_1}^2K_2(\delta_1)^2 - (\alpha_1)^2{e_1}^2K_2(\delta_1)^2 + \delta_1K_1}{(\alpha_1)^2{e_1}^2\delta_1 - K_1}\right) = \frac{K_2}{(\alpha_1)^2{e_1}^2}$$

მაშასადამე, (52) წირის დახრილი ასიმპტოტია:

$$\delta_2 = \frac{K_2}{K_1}\delta_1 + \frac{K_2}{(\alpha_2)^2(e_2)^2}$$

ვიპოვოთ (52) წირის ზრდადობისა და კლებადობის შუალედები და მაქსიმუმისა და მინიმუმის წერტილებია:

$(\delta_2)' = \frac{(\alpha_1)^2 {e_1}^2 K_2}{K_1} \cdot \frac{((\alpha_1)^2 {e_1}^2 \delta_1 - 2K_1)\delta_1}{((\alpha_1)^2 {e_1}^2 \delta_1 - 2K_1)^2}$



0 არის მაქსიმუმის წერტილი. თუ $\delta_1 = 0$, მაშინ $\delta_2 = 0$.

$\delta_1 = \frac{2K_1}{(\alpha_1)^2 {e_1}^2}$ არის მინიმუმის წერტილი. ამ დროს

$$\delta_2 = \frac{(\alpha_1)^2 {e_1}^2 K_2}{K_1} \cdot \frac{4(K_1)^2}{(\alpha_1)^4 {e_1}^4} \cdot \frac{1}{(\alpha_1)^2 {e_1}^2 \frac{2K_1}{(\alpha_1)^2 {e_1}^2} - K_1} = \frac{4K_2}{(\alpha_1)^2 {e_2}^2} = \frac{4K_2}{(\alpha_2)^2 {e_2}^2}$$

ეს ნიშნავს, რომ (52) წირი ეხება (16) ტოლობით მოცემულ წრფეს.

ვნახოთ, როცა $\alpha_1 e_1 = \alpha_2 e_2$, მაშინ აქვთ თუ არა საერთო წერტილი $(\delta_1; \delta_2)$ საკოორდინატო სიბრტყის პირველ მეოთხედში (23) და (52) წირებს. (23)-დან:

$$\delta_2 = \frac{4K_2\delta_1}{A^2\delta_1 - 4K_1} = \frac{K_2\delta_1}{(\alpha_1)^2 {e_1}^2 \delta_1 - K_1} \Longrightarrow$$

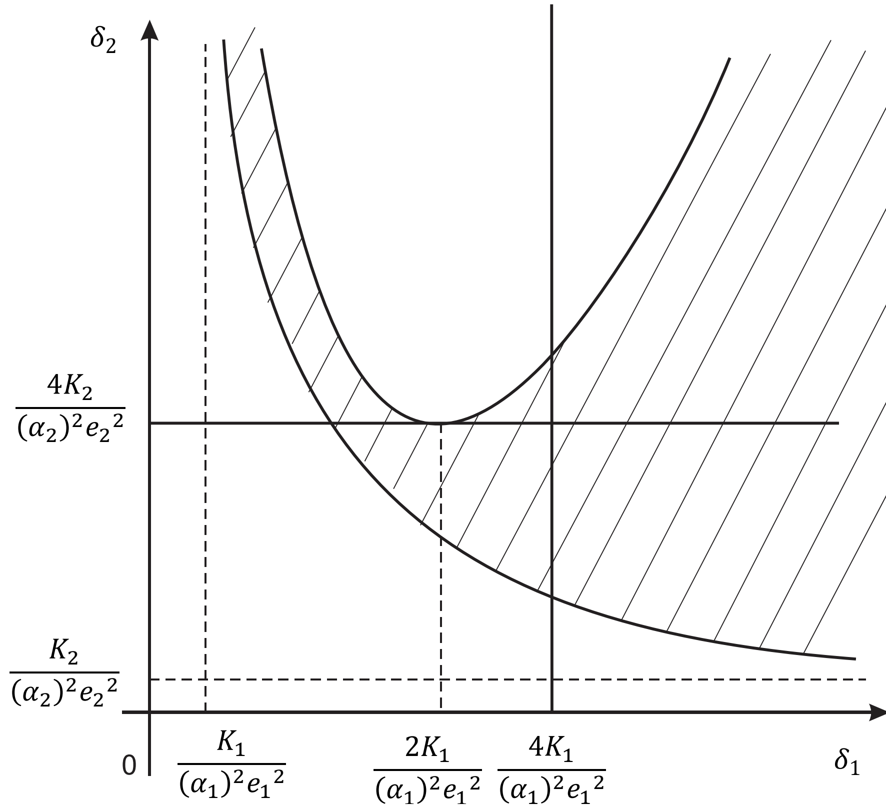
$$\Longrightarrow \frac{K_2\delta_1}{({\alpha_1}^2)(e_1)^2\delta_1 - K_1} = \frac{(\alpha_1)^2 {e_1}^2 K_2 (\delta_1)^2}{K_1((\alpha_1)^2 {e_1}^2 \delta_1 - K_1)} \Longrightarrow$$

რადგან ჩვენ განვიხილავთ შემთხვევებს, როცა $\delta_1 > 0$, $\delta_2 > 0$, ამიტომ

$$1 = \frac{(\alpha_1)^2 {e_1}^2 \delta_1}{K_1} \Longrightarrow \; \delta_1 = \frac{K_1}{(\alpha_1)^2 {e_1}^2}$$

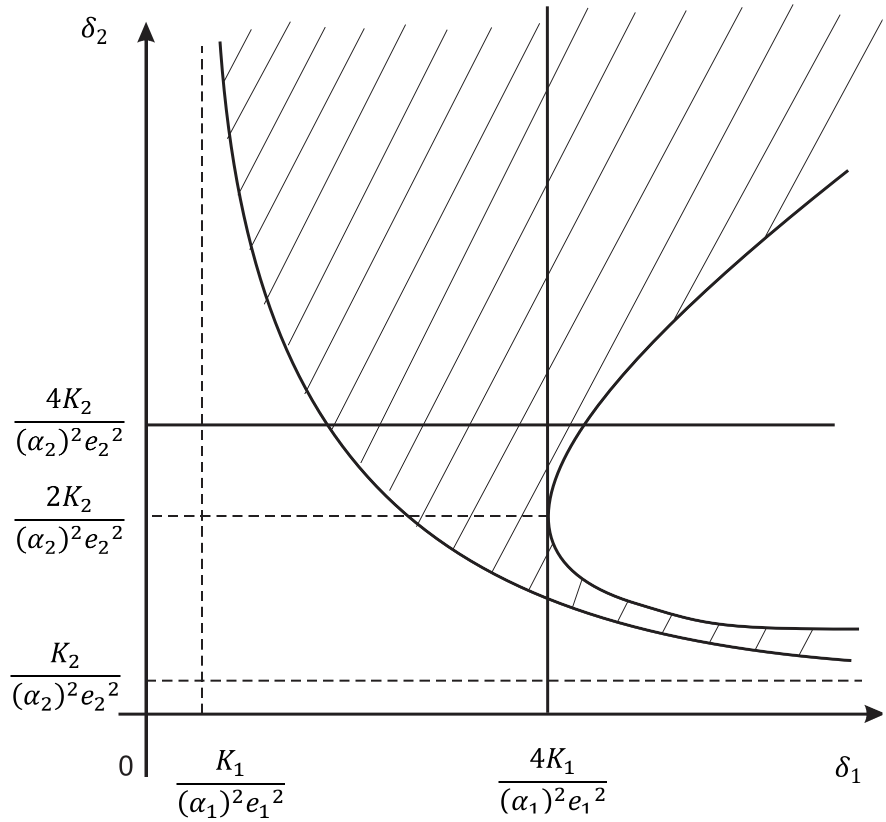
$\delta_1 = \frac{K_1}{(\alpha_1)^2 {e_1}^2}$ არის (52) წირის ვერტიკალური ასიმპტოტი, ამიტომ (23) და (52) წირებს საერთო წერტილები არ გააჩნიათ.

ყოველივე ზემოთქმულის გათვალისწინებით მივიღებთ, რომ როცა $\alpha_1 e_1 = \alpha_2 e_2$, მაშინ I_1^+ დადებითია ნახ. 16-ზე დაშტრიხულ სიმრავლეზე



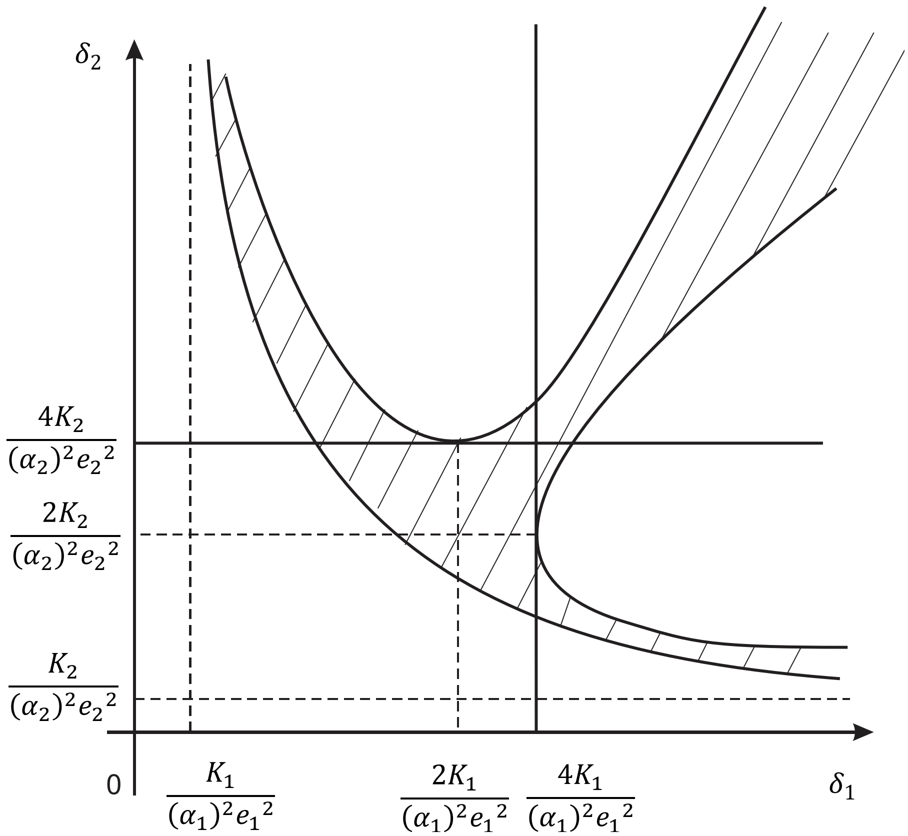
ნახ. 16

ზუსტად ისევე, როგორც დავადგინეთ I_1^+-ს დადებითობის არე, შეგვიძლია დავადგინოთ, რომ I_2^+-ის დადებითობის არეა ნახ. 17-ზე მოცემული დაშტრიხული სიმრავლე.



ნახ. 17

მაშასადამე, თუ $\alpha_1 e_1 = \alpha_2 e_2$, მაშინ I_1^+ და I_2^+ სიდიდეები ორივე ერთად დადებითია ნახ. 18-ზე დაშტრიხულ სიმრავლეზე:



ნახ. 18

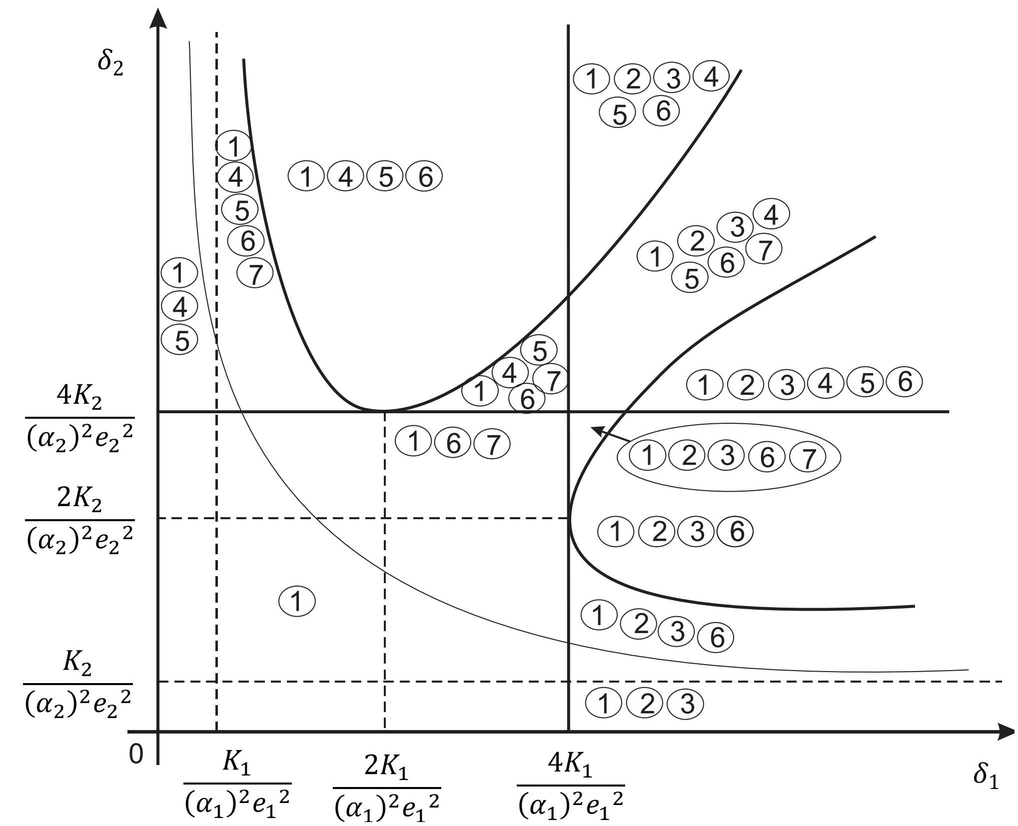
თუ $\alpha_1 e_1 = \alpha_2 e_2$, მაშინ $I_1^+ > 0$ უტოლობა ტოლფასია უტოლობის:

$$\frac{2\alpha_i e_i \alpha K_J}{K_i}\delta_i > -\delta_j\sqrt{D}$$

(აქ თუ $i = 1$, მაშინ $j = 2$; თუ $i = 2$, მაშინ $j = 1$)

ეს უკანასკნელი პირობა სრულდება $(\delta_1;\delta_2)$ საკოორდინატო სიბრტყის პირველი მეოთხედის ყოველ წერტილზე, ანუ თუ $(\delta_1;\delta_2)$ წერტილი ეკუთვნის (23) პირობით განსაზღვრულ არეს, მაშინ აუცილებლად $I_1^- > 0$, $I_2^- > 0$.

მაშასადამე, როცა $\alpha_1 e_1 = \alpha_2 e_2$, მაშინ გვაქვს ნახ. 19-ზე მოცემული სურათი:


δ_2
δ_1
0
$\frac{K_1}{(\alpha_1)^2 e_1{}^2}$
$\frac{2K_1}{(\alpha_1)^2 e_1{}^2}$
$\frac{4K_1}{(\alpha_1)^2 e_1{}^2}$
$\frac{K_2}{(\alpha_2)^2 e_2{}^2}$
$\frac{2K_2}{(\alpha_2)^2 e_2{}^2}$
$\frac{4K_2}{(\alpha_2)^2 e_2{}^2}$
1 4 5
1 4 5 6 7
1 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6 7
1 4 5 6 7
1 2 3 4 5 6
1 6 7
1 2 3 6 7
1 2 3 6
1
1 2 3 6
1 2 3

ნახ. 19

$I_i = 0; I_i = \overset{+}{I}_i(\delta_1;\delta_2), \; I_i = \overset{-}{I}_i(\delta_1;\delta_2), \; I_i = I_i^{+}(\delta_1;\delta_2), \; I_i = I_i^{-}(\delta_1;\delta_2)$ ფუნქციების გრაფიკები აღვნიშნოთ შესაბამისასდ $0_i, \overset{+}{I}_i, \overset{-}{I}_i, I_i^{+}, I_i^{-}$. თითოეული ფუნქცია განსაზღვრულია $(\delta_1;\delta_2)$ საკოორდინატო სიბრტყის პირველი მეოთხედის გარკვეულ არეში. $\pi^p(A)$-თი აღვნიშნოთ A სიმრავლის პროექცია $(\delta_1;\delta_2)$ სბრტყეზე. ჩვენი მიზანია ვიპოვოთ $\pi\left(0_i \cap \overset{+}{I}_i\right), \pi\left(0_i \cap \overset{-}{I}_i\right), \; \pi\left(0_i \cap I_i^{+}\right), \; \pi\left(0_i \cap I_i^{-}\right),$

$, \pi\left(\overset{+}{I}_i \cap I_i^{+}\right), \pi\left(\overset{+}{I}_i \cap I_i^{-}\right), \pi\left(\overset{-}{I}_i \cap I_i^{+}\right), \pi\left(\overset{-}{I}_i \cap I_i^{-}\right),$

$\pi\left(\overset{-}{I}_i \cap \overset{-}{I}_i\right), \; \pi\left(\overset{+}{I}_i \cap \overset{-}{I}_i\right).$

სხვა სიტყვებით რომ ვთქვათ, ჩვენი მიზანია $(\delta_1;\delta_2)$ საკოორდინატო სიბრტყის პირველ მეოთხედში მოვძებნოთ $(\delta_1;\delta_2)$ პარამეტრების ის მნიშვნელობები, რომლებისთვისაც სრულდება ერთ-ერთი მაინც შემდეგი ტოლობებიდან:

$\overset{+}{I}_i = 0, \; \overset{-}{I}_i = 0, \; I_i^{+} = 0, \; I_i^{-} = 0, \; \overset{+}{I}_i = \overset{-}{I}_i, \; \overset{+}{I}_i = I_i^{+}, \; \overset{+}{I}_i = I_i^{-}, \; \overset{-}{I}_i = I_i^{+}, \; \overset{-}{I}_i = I_i^{-}, \; I_i^{+} = I_i^{-}$

$$1. \; \overset{+}{I}_i = 0 \Leftrightarrow \frac{V_i}{2} + \frac{V}{2\alpha_i}\sqrt{(\alpha_i)^2 e^2 - \frac{4K_i}{\delta_i}} = 0 \tag{53}$$

მარცხენა მხარეში მდგომი პირველი შესაკრები დადებითია, მეორე კი არაუარყოფითი, ამიტომ (53) განტოლებას ამონახსნი არა აქვს:

$$\pi\left(0_i \cap \overset{+}{I}_i\right) = \varnothing$$

$$2. \; \overset{-}{I}_i = 0 \Leftrightarrow \frac{V_i}{2} - \frac{V}{2\alpha_i}\sqrt{(\alpha_i)^2 e_i^2 - \frac{4K_i}{\delta_i}} = 0 \qquad)$$

$$\frac{V_i}{2} - \frac{V_i}{2\alpha_i}\sqrt{(\alpha_i)^2 e^2 - \frac{K_i}{\delta_i}} = \frac{V_i}{2} - \frac{V\alpha_i e_i}{2\alpha_i}\sqrt{1 - \frac{4K_i}{(\alpha_i)^2 e_i^2 \delta_i}} = \frac{V_i}{2} - \frac{V_i}{2}\sqrt{1 - \frac{4K_i}{(\alpha_i)^2 e_i^2 \delta_i}} > 0$$

ამიტომ (54) განტოლებას ამონახსნი არ აქვს:

$$\pi\left(0_i \cap \overset{-}{I}_i\right) = \varnothing$$

3. $$I_i^+ = 0 \Leftrightarrow V_i - V(\mu h_i - \alpha_i)\frac{A\sqrt{D}}{2\delta_i\left(\frac{K_1}{\delta_1}+\frac{K_2}{\delta_2}\right)} = 0 \qquad (55)$$

$$\Leftrightarrow (\alpha_i e_i - \alpha_i e_j)\delta_j + \frac{2\alpha_i e_i K_j}{K_i^p}\delta_j = \delta_j\sqrt{D}$$

თუ გავყვებით $I_i^+ > 0$ უტოლობის ამოხსნის გზას, ვნახავთ, რომ თუ $\alpha_i e_i \geq \alpha_j e_j$, მაშინ ეს ტოლობა სრულდება წირზე:

$$\delta_j = \frac{K_i(\alpha_i)^2 e_i^2(\delta_i)^2}{K_j(\alpha_1 e_1 \alpha_2 e_2 \delta_i - K_j)} \qquad (56)$$

ხოლო თუ $\alpha_i e_i < \alpha_j e_j$, მაშინ ტოლობა სრულდება (56) წირის იმ წერტილებისათვის, რომლებიც აკმაყოფილებენ არეს:

$$\alpha_i \geq \frac{2K_i}{\alpha_i e_i A}$$

4. $$I_i^- = 0 \Leftrightarrow (\alpha_i e_i - \alpha_j e_j)\delta_j + \frac{2\alpha_i e_i K_j}{K_i}\delta_i = -\delta_j\sqrt{D}$$

(57)

თუ გავყვებით $I_i^- > 0$ უტოლობის ამოხსნის გზას, ვნახავთ, რომ როცა $\alpha_i e_i \geq \alpha_j e_j$, მაშინ (57) ტოლობა არ სრულდება $(\delta_1;\delta_2)$ საკოორდინატო სიბრტყის პირველი მეოთხედის არც ერთ წერტილზე, ხოლო თუ $\alpha_i e_i < \alpha_j e_j$, მაშინ (57) ტოლობას ადგილი აქვს (56) წირის იმ წერტილებისათვის, რომლებისთვისაც

$$\delta_i \leq \frac{2K_i}{\alpha_i e_i A}$$

5. $$\overset{+}{I_i} = \overset{-}{I_i} \Leftrightarrow \frac{V_i}{2} + \frac{V}{2\alpha_i}\sqrt{(\alpha_i)^2 e_i^{\,2} - \frac{4K_i}{\delta_i}} = \frac{V_i}{2} - \frac{V}{2\alpha_i}\sqrt{(\alpha_i)^2 e_i^{\,2} - \frac{4K_i}{\delta_i}}$$

ეს უტოლობა სრულდება მაშინ და მხოლოდ მაშინ, როცა $\sqrt{(\alpha_i)^2 e_i^{\,2} - \frac{4K_i}{\delta_i}} = 0$, ანუ როცა

$$\delta_i = \frac{4K_i}{(\alpha_i)^2 {e_i}^2}.$$

6. $\overset{+}{I}_i = I_i^+$

ამოვხსნათ ეს განტოლება იმ შემთხვევაში, როცა $i = 1$ (ანალოგიურად იქნება $i = 2$-თვის):

$$\frac{V_1}{2} - \frac{V}{2\alpha_i}\sqrt{(\alpha_1)^2 {e_1}^2 - \frac{4K_1}{\delta_1}} = \frac{V_1}{2} - V(\mu h_1 - \alpha_1)\frac{A + \sqrt{D}}{2\delta_1\left(\frac{K_1}{\delta_1} + \frac{K_2}{\delta_2}\right)} \Leftrightarrow$$

$$\Leftrightarrow \sqrt{(\alpha_1)^2 {e_1}^2 - \frac{4K_1}{\delta_1}} = \alpha_1 e_1 - \frac{A^p + \sqrt{D}}{\delta_1\left(\frac{K_1}{\delta_1} + \frac{K_2}{\delta_2}\right)} \Leftrightarrow$$

$$\Leftrightarrow \sqrt{(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1\delta_1} = \alpha_1 e_1 \delta_1 - \frac{K_1\delta_1\delta_2\left(A + \sqrt{D}\right)}{K_1\delta_2 + K_2\delta_1} \Leftrightarrow$$

$$\Leftrightarrow \alpha_1 e_1 \delta_1 - \frac{K_1\delta_1\delta_2 A^p}{K_1\delta_2 + K_2\delta_1} = \sqrt{(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1\delta_1} + \frac{K_1\delta_1\delta_2\sqrt{D}}{K_1\delta_2 + K_2\delta_1}$$

$$\Leftrightarrow \frac{(\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2)\delta_1}{K_1\delta_2 + K_2\delta_1}$$

$$= \frac{(K_1\delta_2 + K_2\delta_1)\sqrt{(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1\delta_1} + K_1\delta_1\delta_2\sqrt{D}}{K_1\delta_2 + K_2\delta_1} \qquad (58)$$

ავიყვანოთ (58) ტოლობის ორივე მხარე კვადრატში:

$$(\alpha_1)^2 {e_1}^2 (K_2)^2 (\delta_1)^4 - 2\alpha_1 e_1 \alpha_2 K_1 K_2 (\delta_1)^3 \delta_2 + (\alpha_2)^2 {e_2}^2 (K_1)^2 (\delta_1)^2 (\delta_2)^2 =$$

$$= [(K_1)^2(\delta_2)^2 + 2K_1K_2\delta_1\delta_2 + (K_2)^2(\delta_1)^2][(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1\delta_1] +$$

$$+2(K_1\delta_2 + K_2\delta_1)K_1\delta_1\delta_2\sqrt{\delta_1((\alpha_1)^2 {e_1}^2 \delta_1 - 4K_1\delta_1) \cdot \frac{((A^p)^2\delta_1\delta_2 - 4K_1\delta_2 - 4K_2\delta_1)}{\delta_1\delta_2}} +$$

$$+(K_1)^2(\delta_1)^2(\delta_2)^2\left((\alpha_1)^2 {e_1}^2 + 2\alpha_1 e_1 \alpha_2 e_2 + (\alpha_2)^2 {e_2}^2 - \frac{4K_1}{\delta_1} - \frac{4K_2}{\delta_2}\right) \Leftrightarrow$$

$$\Leftrightarrow (\alpha_1)^2 {e_1}^2 (K_2)^2 (\delta_1)^4 - 2\alpha_1 e_1 \alpha_2 e_2 K_1 K_2 (\delta_1)^3 \delta_2 + (\alpha_1)^2 {e_2}^2 (K_1)^2 (\delta_1)^2 (\delta_2)^2 =$$

$$= (\alpha_1)^2 {e_1}^2 (K_1)^2 (\delta_1)^2 (\delta_2)^2 + 2(\alpha_1)^2 {e_1}^2 K_1 K_2 (\delta_1)^3 \delta_2 + (\alpha_1)^2 {e_1}^2 (K_2)^2 (\delta_1)^4 -$$

$$-4(K_1)^3\delta_1(\delta_2)^2 - 8(K_1)^2K_2(\delta_1)^2\delta_2 - 4K_1(K_2)^2(\delta_1)^3 +$$

$$+(\alpha_1)^2 {e_1}^2 (K_1)^2 (\delta_1)^2 (\delta_2)^2 + 2\alpha_1 e_1 \alpha_2 e_2 (K_1)^2 (\delta_1)^2 (\delta_2)^2 +$$

$$+(\alpha_2)^2 {e_2}^2 (K_1)^2 (\delta_1)^2 (\delta_2)^2 - 4(K_1)^3 \delta_1 (\delta_2)^2 - 4(K_1)^2 K_2 (\delta_1)^2 \delta_2 +$$

$$+2(K_1\delta_2 + K_2\delta_2) K_1 \delta_1 \sqrt{\delta_2((\alpha_1)^2 {e_1}^2 \delta_1 - 4K_1)((A^p)^2 \delta_1 \delta_2 - 4K_1\delta_2 - 4K_2\delta_1)} \Leftrightarrow$$

$$\Leftrightarrow 2\alpha_1 e_1 \alpha_2 e_2 K_1 (\delta_1)^2 \delta_2 (K_1\delta_2 + K_2\delta_1) + 2(\alpha_1)^2 {e_1}^2 K_1 \delta_1 (K_1\delta_2 + K_2\delta_1) -$$

$$-8(K_1)^2 \delta_1 \delta_2 (K_1\delta_2 + K_2\delta_1) - 4K_1K_2(\delta_2)^2 (K_1\delta_2 + K_2\delta_1) =$$

$$= -2(K_1\delta_2 + K_2\delta_1) K_1 \delta_1 \sqrt{\delta_2((\alpha_1)^2 {e_1}^2 \delta_1 - 4K_1)((A^p)^2 \delta_1 \delta_2 - 4K_1\delta_2 - 4K_2\delta_1)} \Leftrightarrow$$

$$\Leftrightarrow -\alpha_1 e_1 A^p \delta_1 \delta_2 + 4K_1\delta_2 + 2K_2\delta_1 =$$

$$= \sqrt{\delta_2((\alpha_1)^2 {e_1}^2 \delta_1 - 4K_1)((A^p)^2 \delta_1 \delta_2 - 4K_1\delta_2 - 4K_2\delta_1)} \qquad (59)$$

ავიყვანოთ (59) ტოლობის ორივე მხარე კვადრატში:

$$(\alpha_1)^2 {e_1}^2 A^2 (\delta_1)^2 (\delta_2)^2 + 16(K_1)^2 (\delta_2)^2 + 4(K_2)^2 (\delta_1)^2 - 8\alpha_1 e_1 K_1 A \delta_1 (\delta_2)^2 -$$

$$-4\alpha_1 e_1 A (\delta_1)^2 \delta_2 + 16K_1K_2\delta_1\delta_2 =$$

$$= (\alpha_1)^2 {e_1}^2 A^2 (\delta_1)^2 (\delta_2)^2 - 4(\alpha_1)^2 {e_1}^2 K_1 \delta_1 (\delta_2)^2 - 4(\alpha_1)^2 {e_1}^2 K_2 (\delta_1)^2 \delta_2 -$$

$$-4K_1 (A^p)^2 \delta_1 (\delta_2)^2 + 16(K_1)^2 (\delta_2)^2 + 16K_1K_2\delta_1\delta_2 \Leftrightarrow$$

$$\Leftrightarrow (K_2)^2 \delta_1 - 2\alpha_1 e_1 K_1 A (\delta_2)^2 - \alpha_1 e_1 K_2 A \delta_1 \delta_2 + (\alpha_1)^2 e_1 K_1 (\delta_2)^2 +$$

$$+(\alpha_1)^2 e_1 K_2 \delta_1 \delta_2 + K_1 A^2 (\delta_2)^2 = 0 \Leftrightarrow$$

$$\Leftrightarrow (\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2 - \alpha_1 e_1 \alpha_2 e_2 K_2 \delta_1 \delta_2 + (K_2)^2 \delta_1 = 0 \Leftrightarrow$$

$$\Leftrightarrow \delta_1 = \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)}$$

ეს კი არის l_5 წირის განტოლება.

l_5წირის განტოლება ჩვენ მივიღეთ (58) ტოლობიდან. ვიპოვოთ l_5 წირის ის წერტილები, რომლებიც არაა გარეშე ფესვი (58) განტოლებისათვის. ამისათვის საკმარისია ვიპოვოთ l_5 წირის ის წერტილები, რომლებისთვისაც (58) და (59) განტოლების მარცხენა მხარე არის არაუარყოფითი. ჯერ ვაჩვენოთ, რომ l_5 წირით (58) ტოლობის მარცხენა მხარე არის არაუარყოფითი:

$$\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2 = \alpha_1 e_1 K_2 \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)} - \alpha_2 e_2 K_1 \delta_2 =$$

$$= \alpha_2 e_2 K_1 \delta_2 \frac{\alpha_1 e_1 \alpha_2 e_2 \delta_2 - \alpha_1 e_1 \alpha_2 e_2 \delta_2 + K_2}{\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2} = \frac{\alpha_2 e_2 K_1 K_2 \delta_2}{\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2}$$

ვინაიდან $\delta_2 = \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}$ წრფე წარმოადგენს l_5 წირის ასიმპტოტებს, ამიტომ l_5 წირის ყოველი წერტილისათვის $\delta_2 > \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}$და ამიტომ $\frac{\alpha_2 e_2 K_1 K_2 \delta_2}{\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2} > 0$ ეს კი ნიშნავს, რომ l_5 წირის წერტილებზე (58) უტოლობის მარცხენა მხარე არის დადებითი და (58)⇔(59).

ახლა ვიპოვოთ l_5 წირის ის წერტილები, რომელ წერტილებზეც (59) უტოლობის მარცხენა მხარე არის არაუარყოფითი:

$$-\alpha_1 e_1 A\delta_1\delta_2 + 4K_1\delta_2 + 2K_2\delta_1 = -(\alpha_1 e_1 A\delta_2 - 2K_2)\delta_1 + 4K_1\delta_2 =$$

$$= -(\alpha_1 e_1 A\delta_2 - 2K_2)\cdot\frac{(\alpha_2)^2 e_2{}^2 K_1(\delta_2)^2}{K_2(\alpha_1 e_1\alpha_2 e_2\delta_2 - K_2)} + 4K_1\delta_2 =$$

$$= -K_1\delta_2\cdot\frac{\alpha_1 e_1 A(\alpha_2)^2 e_2{}^2(\delta_2)^2 - 2(\alpha_2)^2 e_2{}^2 K_2\delta_2 - 4\alpha_1 e_1\alpha_2 e_2 K_2 + 4\delta_2(K_2)^2}{K_2(\alpha_1 e_1\alpha_2 e_2\delta_2 - K_2)} =$$

$$= -K_1\delta_2$$
$$\cdot\frac{(\alpha_1)^2 e_1{}^2(\alpha_2)^2 e_2{}^2(\delta_2)^2 - 4\alpha_1 e_1\alpha_2 e_2 K_2\delta_2 + 4(K_2)^2 + (\alpha_2)^2 e_2{}^2\delta_2(\alpha_1 e_1\alpha_2 e_2\delta_2 - 2K_2)}{K_2(\alpha_1 e_1\alpha_2 e_2\delta_2 - K_2)} =$$

$$= \frac{-K_1\delta_2}{K_2(\alpha_1 e_1\alpha_2 e_2\delta_2 - K_2)}\left[(\alpha_1 e_1\alpha_2 e_2\delta_2 - 2K_2)^2 + (\alpha_2)^2 e_2{}^2\delta_2(\alpha_1 e_1\alpha_2 e_2\delta_2 - 2K_2)\right]$$

როგორც ზემოთ ვნახეთ, წერტილი $Q_1 = \left(\frac{4K_1}{(\alpha_1)^2 e_1{}^2}; \frac{2K_2}{\alpha_1 e_1\alpha_2 e_2}\right)$ არის l_2 და l_5 წირის შეხების წერტილი, რომლებიც მდებარეობენ Q_1 წერტილის ზემოთ, მათთვის $\delta_2 > \frac{2K_2}{\alpha_1 e_1\alpha_2 e_2}$, ამიტომ კვადრატულ ფრჩხილებში მოთავსებული გამოსახულება არის უარყოფითი.თუ

$$\frac{K_2}{\alpha_1 e_1\alpha_2 e_2} < \delta_2 < \frac{2K_2}{\alpha_1 e_1\alpha_2 e_2}$$

(გავიხსენოთ, რომ $\delta_2 = \frac{K_2}{\alpha_1 e_1\alpha_2 e_2}$არის l_5 წირის მხები წრფე)

მაშინ კვადრატულ ფრჩხილებში მოთავსებულ გამოსახულებას ასე ჩავწერთ:

$$(\alpha_1 e_1\alpha_2 e_2\delta_2 - 2K_2)(\alpha_2 e_2 A\delta_2 - 2\delta_2)(60)$$

თუ $\alpha_1 e_1 \le \alpha_2 e_2$, მაშინ

$$\alpha_2 e_2 A\delta_2 - 2K_2 > \alpha_2 e_2 A\frac{K_2}{\alpha_1 e_1\alpha_2 e_2} - 2K_2 =$$

$$= 2K_2\left(\frac{A}{\alpha_1 e_1} - 2\right) \ge 2K_2\left(\frac{2\alpha_1 e_1}{\alpha_1 e_1} - 2\right) = 0$$

თუ $\alpha_1 e_1 > \alpha_2 e_2$, მაშინ ჩვენ გვაინტერესებს l_5 წირის ის წერტილები, რომლებისთვისაც $\delta_2 = \frac{2K_2}{\alpha_2 e_2 A^p}$(როცა $\delta_2 = \frac{2K_2}{\alpha_2 e_2 A^p}$, მაშინ l_5 წირი ეხება l_3 წირს). ასეთი δ_2-თვის (60)-ის მეორე თანამანრავლი არის დადებითი.

მაშასადამე, (59) განტოლების ამონახსნია l_5 წირის ის წერტილები, რომელებისთვისაც $I_1^+ > 0$, $I_2^+ > 0$ და რომლებიც მდებარეობენ $\delta_2 = \frac{2K_2}{\alpha_2 e_2 A}$ წრფის ქვემოთ.

ანალოგიურად ამოიხსნება განტოლება $\overset{+}{I_2} = I_2^+$ და მიიღება, რომ ან განტოლებას აკმაყოფილებენ l_4 წირის ის წერტილები, რომლებისთვისაც $I_2^+ > 0$ და რომლებიც მდებარეობენ $\delta_1 = \frac{2K_1}{\alpha_1 e_1 \alpha_2 e_2}$ წრფის მარცხნივ.

7. $\overset{+}{I_i} = I_i^-$

ამოვხსნათ ეს განტოლება $i = 1$-თვის:

$$\frac{V_1}{2} - \frac{V}{2\alpha_1}\sqrt{(\alpha_1)^2 {e_1}^2 - \frac{4K_1}{\delta_1}} = V_1 - V(\mu h_1 - \alpha_1)\frac{A + \sqrt{D}}{2\delta_1\left(\frac{K_1}{\delta_1} + \frac{K_2}{\delta_2}\right)} \tag{61}$$

თუ ჩავატარებთ იმის ანალოგიურ გარდაქმნებს, რაც ჩავატარეთ $\overset{+}{I_i} = I_i^+$ განტოლების ამოხსნისას, მივიღებთ, რომ (61) ტოლფასია შემდეგი განტოლების:

$$\left(\alpha_1 e_1 K_2 \delta_1 - \alpha_2\, e_2 K_1 \delta_2\right)\delta_1 = (K_1\delta_2 + K_2\delta_1)\sqrt{(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1\delta_1} - K_1\delta_1\delta_2\sqrt{D} \tag{62}$$

ტოლობის ორივე მხარის კვადრატში აყვანით და შემდგომი გამარტივებით (ისევე, როგორც (58) განტოლების შემთხვევაში) მივიღებთ:

$$\alpha_1 e_1 A\delta_1\delta_2 - 4K_1\delta_2 - 2K_2\delta_1 =$$

$$= \sqrt{\delta_2((\alpha_1)^2 {e_1}^2 \delta_1 - 4K_1)((A)^2\delta_1\delta_2 - 4K_1\delta_2 - 4K_2\delta_1)} \tag{63}$$

(63) ტოლობის ორივე მხარის კვადრატში აყვანით და შემდგომი გამარტივებით (ისევე, როგორც (59) განტოლების შემთხვევაში) მივიღებთ, რომ (63) განტოლების ამონახსნია l_5 წირის წერტილები:

$$\delta_1 = \frac{(\alpha_2)^2 {e_2}^2 K_1 (\delta_2)^2}{K_2(\alpha_1 e_1 \alpha_2 e_2 \delta_2 - K_2)}$$

ვიპოვოთ l_5 წირის ის წერტილები, რომლებიც არაა გარეშე ფესვი (62) განტოლებისათვის. ამისათვის საკმარისია, რომ ვიპოვოთ l_5 წირის ის წერტილები, რომლებისთვისაც სრულდება შემდეგი ორი პირობა:

(I) $(\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2)\delta_1 + K_1 \delta_1 \delta_2 D \geq 0$

(II) $(K_1\delta_2 + K_2\delta_1)^2((\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1\delta_1) - (\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2)^2 (\delta_1)^2 - (K_1)^2 (\delta_1)^2 (\delta_2)^2 D =$

$$= 2(\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2) K_1 (\delta_1)^2 \delta_2 \sqrt{D}$$

ტოლობის ორივე მხარეში იდგეს ერთნაირი ნიშნის გამოსახულებები.

როგორც ზემოთ ვნახეთ (როდესაც ვაჩვენეთ, რომ (58) ტოლობის მარცხენა მხარე არის დადებითი), l_5 წირის წერტილებისთვის $\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2 > 0$, ამიტომ (I) პირობას აკმაყოფილებს l_5 წირის ყოველი წერტილი, ხოლო (II) ტოლობის მარჯვენა მხარეში მდგომი გამოსახულება არის დადებითი. დაგვრჩა ვიპოვოთ l_5 წირის ის წერტილები, რომლებისთვისაც (II) ტოლობის მარცხენა მხარე არის დადებითი:

$$\delta_2 = \frac{2K_2}{\alpha_2 e_2 A} \tag{64}$$

თუ $\alpha_1 e_1 \leq \alpha_2 e_2$, მაშინ:

$$\frac{2K_2}{\alpha_2 e_2 A} = \frac{2K_2}{\alpha_2 e_2 (\alpha_1 e_1 + \alpha_2 e_2)} \leq \frac{2K_2}{\alpha_2 e_2 \cdot 2\alpha_1 e_1} = \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}$$

მაგრამ წრფე $\delta_2 = \frac{K_2}{\alpha_1 e_1 \alpha_2 e_2}$, არის l_5 წირის ჰორიზონტალური ასიმპტოტი. ეს ნიშნავს, რომ თუ $\alpha_1 e_1 \leq \alpha_2 e_2$, მაშინ l_5 წირის არც ერთი წერტილი არ აკმაყოფილებს (64) პირობას და ამიტომ ტოლობას $\overset{+}{I_1} = I_1^-$ ადგილი არ ექნება l_5 წირის არც ერთ წერტილზე. ამ შემთხვევაში

$$\pi\left(\overset{+}{I_1} \cap I_1^-\right) = \varnothing$$

თუ $\alpha_1 e_1 > \alpha_2 e_2$, მაშინ $\delta_1 = \frac{2K_2}{\alpha_2 e_2 A}$-თვის l_5 წირი ეხება l_3 წირს. ამიტომ (64) პირობა შესრულდება l_5 წირის იმ ნაწილზე, რომელიც მდებარეობს l_5დაl_3 წირის შეხების წერტილის ქვემოთ.

ანალოგიური მსჯელობით მივიღებთ, რომ როცა $\alpha_1 e_1 \geq \alpha_2 e_2$, მაშინ $\pi\left(\overset{+}{I_2} \cap I_2^-\right) = \varnothing$, ხოლო როცა $\alpha_1 e_1 < \alpha_2 e_2$, მაშინ ტოლობას $\overset{+}{I_2} = I_2^-$ ადგილი აქვს l_4 წირის იმ ნაწილზე, რომელიც მდებარეობს l_4 და l_3 წირების შეხების წერტილის მარცხნივ.

8. $\overline{I_i} = I_i^+$

ამოვხსნათ ეს განტოლება იმ შემთხვევაში, როცა $i = 1$

$$\frac{V_1}{2} - \frac{V}{2\alpha_1}\sqrt{(\alpha_1)^2 {e_1}^2 - \frac{4K_1}{\delta_1}} = V_1 - V(\mu h_1 - \alpha_1)\frac{A + \sqrt{D}}{2\delta_1\left(\frac{K_1}{\delta_1} + \frac{K_2}{\delta_2}\right)} \Leftrightarrow$$

$$\Leftrightarrow (\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2)\delta_1 = K_1 \delta_1 \delta_2 \sqrt{D} - (K_1 \delta_2 + K_2 \delta_2)\sqrt{(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1 \delta_1} \qquad (65)$$

ისევე, როგორც წინა შემთხვევაში, ამ ტოლობის კვადრატში აყვანით და მიღებული ტოლობის გამარტივების შემდეგ კიდევ ერთხელ კვადრატში აყვანით მივიღებთ, რომ (65) ტოლობას ადგილი შეიძლება ჰქონდეს l_5 წირზე.

9. $\overline{I_i} = I_i^-$

ამოვხსნათ განტოლება, როცა $i = 1$.

$$(\alpha_1 e_1 K_2 \delta_1 - \alpha_2 e_2 K_1 \delta_2)\delta_1 = -K_1 \delta_1 \delta_2 \sqrt{D} - (K_1 \delta_2 + K_2 \delta_1)\sqrt{(\alpha_1)^2 {e_1}^2 (\delta_1)^2 - 4K_1 \delta_1} \qquad (66)$$

თუ მიღებულ ტოლობას ავიყვანთ კვადრატში, მივიღებთ, რომ (64) ტოლობას შეიძლება აკმაყოფილებდნენ l_5 წირის წერტილები. მაგრამ როგორც ვნახეთ l_5 წირის წერტილებისათვის (66) ტოლობის მარცხენა მხარე არის დადებითი, ხოლო მაჯვენა მხარე კი უარყოფითი, ამიტომ $\overline{I_1} = I_1^-$ განტოლების ამონახსნთა სიმრავლე არის ცარიელი სიმრავლე.

ანალოგიურად შეიძლება ვაჩვენოთ, რომ $\overline{I_2} = I_2^-$ განტოლების ამონახსნთა სიმრავლე ცარიელი სიმრავლეა. მაშასადამე:

$$\pi\left(\overline{I_i} = I_i^-\right) = \varnothing$$

10. $I_i^+ = I_i^-$

ამ ტოლობას ადგილი აქვს მაშინ, როცა $D = 0$, ანუ ტოლობას ადგილი აქვს l_3 წირზე.

დანართი 1

რეგიონის მახასიათებელი μ პარამეტრის გამოთვლა.

ვთქვათ $\Omega = (a_{ij})_{M\times N}$ არის საწარმოებისა და ნედლეულის მომწოდებლებს შორის კავშირის შესაბამისი ინციდენტურონის მატრიცი ($a_{ij} = 1$, თუ i-ურ საწარმოს ამარაგებს j-ური მომწოდებელი, სხვა შემთხვევაში $a_{ij} = 0$). Ω^T-თი აღვნიშნოთ Ω მატრიცის ტრანსპონირებული მატრიცი. ვთქვათ $C = \Omega \cdot \Omega^T$. $C = (c_{ij})_{M\times M}$ მატრიცის c_{ij} ელემენტი გვიჩვენებს თუ რამდენი საერთო მომწოდებელი ჰყავს i -ურ და j -ურ საწარმოებს. შემოვიღოთ აღნიშვნა: $d_{ij} = max\{\sum_{k=1}^{N} a_{ik}; \sum_{k=1}^{N} a_{jk}\}$ (d_{ij} აღნიშნავს i-ური და j -ური საწარმოების მომწოდებელთა რიცხვს შორის უდიდესს), $p_{ij} = \frac{c_{ij}}{d_{ij}}$.

μ-ს გამოსათვლელად ვისარგებლოთ ფორმულით:

$$\mu = \frac{2\sum_{i=1}^{M}\sum_{j=i+1}^{M} p_{ij}}{M(M-1)} \qquad (1).$$

ვაჩვენოთ, რომ (1) გამოსახულება აკმაყოფილებს μ-სადმი წაყენებულ მოთხოვნებს, კერძოდ: თუ ყოველი საწარმო არის იზოლირებული (ანუ არც ერთი საწარმო არაა კავშირში სხვა რომელიმე საწარმოსთან - არა ჰყავთ საერთო მომწოდებელი) მაშინ $\mu = 0$, ხოლო თუ ყოველ საწარმოს კავშირი აქვს ყოველ მომწოდებელთან, მაშინ $\mu = 1$. სხვა შემთხვევაში $0 < \mu < 1$.

შევნიშნოთ, რომ, თუ ყოველი საწარმო არის იზოლირებული, მაშინ Ω მატრიცის ყოველ სვეტში მხოლოდ ერთი ელემენტია 1-ის ტოლი, ხოლო ყველა დანარჩენი არის 0. ამ შემთხვევაში $c_{ij} = 0$ ყოველი $i \neq j$ -თვის. და შესაბამისად $p_{ij} = \frac{c_{ij}}{d_{ij}} = 0$, ამიტომ (1)-დან მივიღებთ, რომ $\mu = 0$.

თუ ყოველ საწარმოს კავშირი აქვს ყოველ მომწოდებელთან, მაშინ ყოველი $i \neq j$ -თვის (1)-დან მივიღებთ, რომ $c_{ij} = d_{ij} = N$. (1) ტოლობის მარჯვენა მხარის მრიცხველში სულ გვაქვს $\frac{M(M-1)}{2}$ შესაკრები, ამასთან თითოეული მათგანი ტოლია 1-ის, ამიტომ ამ შემთხვევაში გვექნება, რომ $\mu = 1$.

განხილული ორი შემთხვევის გარდა სხვა შემთხვევებში არსებობს i-სა და j-ს ერთი მაინც ისეთი მნიშვნელობა, რომლისთვისაც $c_{ij} > 0$ (არსებობს ორი ერთი მაინც წყვილი საწარმოებისა, რომლებთაც აქვთ ერთი მაინც საერთო ინტერესი) და არსებობს k და l-ის ერთი მაინც ისეთი მნიშვნელობა, რომლისთვისაც $c_{kl} < 1$ (არსებობს ერთი მაინც წყვილი საწარმოებისა, რომლებთაც აქვთ ერთი მაინც განსხვავებული ინტერესი).ამ შემთხვევაში

(1) ტოლობის მრიცხველში ერთი მაინც შესაკრები იქნება დადებითი და ერთი მაინც შესაკრები იქნება 1-ზე ნაკლები, ამიტომ μ-თვის მივიღებთ პირობას: $0 < \mu < 1$.

მაშასადამე, (1) ტოლობით განსაზღვრული μ აკმაყოფილებს მისდამი წაყენებულ მოთხოვნებს.

დანართი 2

ვიპოვოთ

$$S= -\sum_{\substack{s=1\\m=0}}^{M} x_{sm} \ln \frac{x_{sm}}{\nu_{sm}} \quad (1)$$

ფუნქციის უდიდესი მნიშვნელობა იმ X_{sm} -ებისთვის, რომლებიც აკმაყოფილებენ პირობას:

$$\sum_{s=0}^{M} x_{sm} = h_m I_m \quad (2)$$

$$x_m \geq 0 \quad (3)$$

(რადგან $\lim_{p \to 0}(p \cdot \ln p) = 0$, ამიტომ თუ რომელიმე $x_m = 0$, მაშინ შეგვიძლია ჩავთვალოთ, რომ შესაბამისი შესაკრები S-ში ტოლია 0-ის).

<u>თეორემა. </u> არსებობს x_{sm} ცვლადების ერთ-ერთი მნიშვნელობა, რომლებიც აკმაყოფილებენ (2), (3) პირობებს და რომლებზედაც S ფუნქცია აღწევს თავის მაქსიმალურ მნიშვნელობას.

<u>დამტკიცება. </u>ვთქვათ Q აღნიშნავს იმ x_{sm} -ების სიმრავლეს, რომლებიც აკმაყოფილებენ (2), (3) პირობებს. ცხადია, რომ Q არის ჩაკეტილი, შემოსაზღვრული სიმრავლე. ამიტომ ვაიერშტრასის თეორემის თანახმად Q სიმრავლეზე უწყვეტი S ფუნქცია აღწევს თავის უდიდეს და უმცირეს მნიშვნელობას.

ვაჩვენოთ, რომ არსებობს x_{sm} ცვლადების ერთადერთი მნიშვნელობა, რომლებზედაც S ფუნქცია აღწევს მაქსიმუმს. ამისათვის განვიხილოთ ფუნქცია

$$f(x)=\sum_{i=1}^{p} x_i l_n \frac{x_i}{a_i}, \quad a_i > 0;\ x_i > 0$$

ვთქვათ y= $(y_1, y_2, \dots, y_p) \neq 0$.

$$f(x+y)-f(x)-(\nabla f(x), y) = \sum_{i=1}^{P}\left[(x_i + y_i)\ln\frac{x_i + y_i}{x_i} - y_i\right] \quad (4)$$

რადგან $y \neq 0$, ამიტომ არსებობს $i \in [1; p]$ ისეთი, რომ $y_i \neq 0$. ვაჩვენოთ, რომ ასეთი i -ს შესაბამისი შესაკრებები (4) ტოლობის მარჯვენა მხარეში მეტია 0-ზე.

განვიხილოთ ფუნქცია $g(x) = x \cdot \ln\frac{x}{a}, \quad x > 0,\ a > 0$.

$$g'(x) = \ln\frac{x}{a} + 1.$$

თუ $x_1 > x_2$, მაშინ $g'(x_1) > g'(x_2)$. ეს კი ნიშნავს, რომ $g(x)$ ფუნქცია არის მკაცრად ამოზნექილი, ამიტომ (4)-დან გვექნება, რომ თუ $y_1 \neq 0$, მაშინ $(x_i + y_i)\ln\frac{x_i + y_i}{x_i} - y_i > 0$, ანუ ყოველი $y \neq 0$-თვის:

$$f(x+y)-f(x) - (\nabla f(x), y) > 0.$$

ე.ი. f(x) ფუნქცია არის მკაცრად ამოზნექილი. რადგან ამოზნექილ სიმრავლეზე (Q არის ამოზნექილი სიმრავლე) მკაცრად ამოზნექილ ფუნქციას შეიძლება გააჩნდეს არა უმეტეს ერთისა მაქსიმუმის წერტილი, ამიტომ (1)-(3) ამოცანას აქვს ერთადერთი ამონახსნი - თვისება დამტკიცებულია.

ვიპოვოთ (1) – (3) ამოცანის ამონახსნი.

განვიხილოთ ლაგრანჟიანი.

$$L = -\sum_{\substack{m=0 \\ s=1}}^{M} x_{sm} \ln\frac{x_{sm}}{\nu_{sm}} + \sum_{s=1}^{M}(h_s I_s - \sum_{m=o}^{M} x_{sm}) c_s$$

$$\frac{\partial L}{\partial x_{sm}} = -\ln\frac{x_{sm}}{\nu_{sm}} - 1 - c_s,\ \ s \in [1; M],\ \ m \in [0; M]$$

$$\begin{cases} -\ln\dfrac{x_{sm}}{\nu_{sm}} - 1 - c_s, \;\; s \in [1;M], \;\; m \in [0;M] \\ \sum\limits_{m=0}^{M} x_{sm} = h_s I_s, \;\; s \in [1;M] \end{cases} \qquad (5)$$

(5) სისტემის პირველი ტოლობიდან :

$$x_{sm} = \nu_{sm} \cdot e^{-1-c_s} \qquad (6)$$

თუ x_{sm} -ების ამ მნიშნელობებს შევიტანთ (5)-ის მეორე განტოლებაში მივიღებთ:

$$e^{-1-c_s} \sum_{m=o}^{M} \nu_{sm} = h_s x_s$$

აქედან:

$$e^{-1-c_s} = \frac{h_s x_s}{\sum\limits_{m=o}^{M} \nu_{sm}}$$

ამიტომ (6)-დან მივიღებთ:

$$x_{sm} = \frac{\nu_{sm}}{\sum\limits_{m=o}^{M} \nu_{sm}} h_s x_s \qquad (7)$$

ცხადია, რომ x_{sm} -ების მიღებული მნიშვნელობები მეტია ან ტოლია 0-ზე.

რადგან S ფუნქცია არის მკაცრად ამოზნწქილი ფუნქცია და (7) არის მისი კრიტიკული წერტილი, ამიტომ (7) ტოლობით მოცემული x_{sm} -ები არიან ის მნიშვნელობები, რომლებზედაც S ფუნქცია აღწევს მაქსიმუმს და რომლებიც აკმაყოფილებენ (2), (3) პირობებს.

დანართი 3

ვთქვათ $\lambda_m^{(P)}$აღნიშნავს m-ურ საწარმოს შესაბამისი რაიონის მოსახლეობის P ჯგუფის მახასიათებელ პარამეტრს (ის, თუ როგორ გამოითვლება $\lambda_m^{(P)}$ პარამეტრი, მოცემული მოსახლეობის დინამიკის მოდელში);$\eta^{(p)}$ აღნიშნავს მოსახლეობის P ჯგუფისათვის რაიონის მახასიათებელი პარამეტრის ($\lambda_m^{(P)}$ პარამეტრები) წონას; β_m-ით აღვნიშნოთ m-ური საწარმოს გადაიარაღების კოეფიციენტი, ხოლო γ-ით კი ამ კოეფიციენტის წონა. $\alpha_m^{(i)}$ -ით აღვნიშნოთ მოგების წილი m-ურ საწარმოში განსახილველი პერიოდიდან i წლის წინ ($\alpha_m^{(0)} = \alpha_m$) , ხოლო $\omega^{(i)}$-თი კი შესაბამისი წონა. აღნიშნული წონები უნდა აკმაყოფილებდნენ ნორმირების პირობას:

$$\sum_{p=1}^{P}\eta^{(p)} + \gamma + \sum_{i=1}^{I}\omega^{(i)}=1.$$

(თუ აღნიშნული პირობა არ სრულდება, მაშინ უნდა მოვახდინოთ ამ წონათა ნორმირება).

თუ δ_{sm}-ით აღვნიშნავთ q-ს უდიდეს მნიშვნელობას, რომლისთვისაც s და m საწარმოები არიან q ბმული, ხოლო dimR_m -ით m-ურ საწარმოს (სიმპლექსის) განზომილებას, მაშინ σ_{sm} კოეფიციენტი შეიძლება გამოვითვალოთ ფორმულით:

$$\sigma_{sm}=(\sum_{p=1}^{P}\lambda_m^{(p)}\eta^{(p)} + \beta_m\gamma + \sum_{i=0}^{I}\alpha_m^{(i)}\omega^{(i)})\frac{\delta_{sm}}{\dim R_m}$$

ლიტერატურა

1. Ландау Л. Д. Лифшиц Е.М. Статистическая физика. М. Наука. 1961.
2. Попков Ю.С. Системный анализ и проблемы развития городов. М. Наука. 1983.
3. R. H. Atkin. Int. J. Man-Manchine studies, 4, 1972.
4. R. H. Atkin. Int. J. Man-Manchine studies, 6, 1974.
5. R. H. Atkin. Mathematical structure in human affairs. Heinemann Educational Books LTD. London, 1974.
6. Математическое моделирование/Под ред. Дж. Эндрюс, Р. Мак-Лоун. М. Мир. 1979.

7. Попков Ю.С. Основы теории динамических систем с энтропиинным оператором и её приложения. 2006. №6. с. 75-105.
8. Попков Ю.С. Теория макросистем. М. 1999.
9. ახობაძ მ. მაკროსისტემების მათემატიკური მოდელირებისა და მართვის საკითხები. მონოგრაფია. თბილისი 1997.
10. ახობაძე მ. კურცხალია. ე. დემოგრაფიული პროცესების მართვის იმიტაციური მოდელი. საქ. მეცნიერებათა აკადემიის მოამბე. ტ. 145. №2.
11. Постон Т. Стюарт И. Теория катастроф. М. Мир. 1980.
12. Макаров И. М. Ахрем А. А. Рахманкулов В. З. Об основных понятях математической теории решетчатых множеств. Труды ИСА РАН . 2009. С. 220-227.
13. ახობაძე მ. მუშკუდიანი ზ. რეგიონალურ სისტემათა, როგორც გომეტრიული სტრუქტურების ფუნქციონირების ანალიზი. საქ, მეცნიერებათა აკადემიის მოამბე. ტ. 150. №1, 1994.

პროგრამის აღწერა

რეგიონი დაყოფილია 2 რაიონად. რეგიონში ფუნქციონირებს 5 საწარმო, რომელთაც ჰყავთ 7 ნედლეულის მომწოდებელი. მომწოდებლებსა და საწარმოებს შორის ინციდენტურობის მატრიცას აქვს სახე:

$$\Omega = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

რეგიონის მახასიათებელი μ პარამეტრის გამოსათვლელად განვიხილოთ მატრიცი

$$C = \Omega \cdot \Omega^{\mathrm{T}} = \begin{pmatrix} 4 & 2 & 2 & 2 & 2 \\ 2 & 4 & 2 & 2 & 3 \\ 2 & 2 & 4 & 2 & 3 \\ 2 & 2 & 2 & 3 & 2 \\ 2 & 3 & 3 & 2 & 5 \end{pmatrix}$$

C მატრიცის C_{sm} ელემენტი $(s \neq m)$ გვიჩვენებს რამდენი საერთო მომწოდებელი ჰყავს s და m საწარმოს. $d_{sm} =$ აღნიშნავს s და m საწარმოების მომწოდებელთა რიცხვს შორის უდიდეს (მაგალითად $d_{15} = max\{4;5\} = 5$; $d_{23} = max\{4;4\} = 4$. $p_{sm} = \frac{C_{sm}}{d_{sm}}$.
გამოვითვალოთ რეგიონის მახასიათებელი პარამეტრი:

$$\mu = \frac{2\sum_{S=1}^{M}\sum_{m=i+1}^{M} P_{sm}}{M(M-1)} = \frac{2(\frac{2}{4}+\frac{2}{4}+\frac{2}{4}+\frac{2}{5}+\frac{2}{4}+\frac{2}{4}+\frac{3}{5}+\frac{2}{4}+\frac{3}{5}+\frac{2}{5})}{5 \cdot 4}$$

$$\mu = 0{,}5$$

საწარმოების მოგების კოეფიციენტები წლების მიხედვით მოცემულია შემდეგ ცხრილში:

<table>
<tr><td rowspan="3">საწარმოს დასახელება</td><td colspan="4">მოგების კოეფიციენტი წინა წლის</td><td colspan="4">მოგების კოეფიციენტი 2 წლის წინ</td></tr>
<tr><td colspan="2">მნიშვნელობა</td><td colspan="2">წონა</td><td colspan="2">მნიშვნელობა</td><td colspan="2">წონა</td></tr>
<tr><td>აღნიშვნა</td><td>სიდიდე</td><td>აღნიშვნა</td><td>სიდიდე</td><td>აღნიშვნა</td><td>სიდიდე</td><td>აღნიშვნა</td><td>სიდიდე</td></tr>
<tr><td>შ.პ.ს. „მუხა“</td><td>$\alpha_1^{(1)}$</td><td>0,164</td><td>$\omega_1^{(1)}$</td><td>0,22</td><td>$\alpha_1^{(2)}$</td><td>0,171</td><td>$\omega_1^{(2)}$</td><td>0,21</td></tr>
<tr><td>შ.პ.ს. „ალფა“</td><td>$\alpha_2^{(1)}$</td><td>0,183</td><td>$\omega_2^{(1)}$</td><td>0,21</td><td>$\alpha_2^{(2)}$</td><td>0,173</td><td>$\omega_2^{(2)}$</td><td>0,21</td></tr>
<tr><td>ი.ს. „ნადიბაიძე“</td><td>$\alpha_3^{(1)}$</td><td>0,175</td><td>$\omega_3^{(1)}$</td><td>0,21</td><td>$\alpha_3^{(2)}$</td><td>0,162</td><td>$\omega_3^{(2)}$</td><td>0,20</td></tr>
<tr><td>ი.ს. „ბრეგვაძე“</td><td>$\alpha_4^{(1)}$</td><td>0,172</td><td>$\omega_4^{(1)}$</td><td>0,23</td><td>$\alpha_4^{(2)}$</td><td>0,170</td><td>$\omega_4^{(2)}$</td><td>0,22</td></tr>
</table>

ასოციაცია „კომპლექსი“	$\alpha_5{}^{(1)}$	0,140	$\omega_5{}^{(1)}$	0,20	$\alpha_5{}^{(2)}$	0,135	$\omega_5{}^{(1)}$	0,19

გადაიარაღების კოეფიციენტები წლების მიხედვით მოცემულია შემდეგ ცხრილში:

საწარმოს დასახელება	გადაიარაღების კოეფიციენტი წინა წლის				გადაიარაღების კოეფიციენტი 2წლის წინ			
	მნიშვნელობა		წონა		მნიშვნელობა		წონა	
	აღნიშვნა	სიდიდე	აღნიშვნა	სიდიდე	აღნიშვნა	სიდიდე	აღნიშვნა	სიდიდე
შ.პ.ს. „მუხა“	$\beta_1{}^{(1)}$	0,081	$\gamma_1{}^{(1)}$	0,12	$\beta_1{}^{(2)}$	0,079	$\gamma_1{}^{(2)}$	0,07
შ.პ.ს. „ალფა“	$\beta_2{}^{(1)}$	0,078	$\gamma_2{}^{(1)}$	0,11	$\beta_2{}^{(2)}$	0,082	$\gamma_2{}^{(2)}$	0,08
ი.ს. „ნადიბაიძე“	$\beta_3{}^{(1)}$	0,093	$\gamma_3{}^{(1)}$	0,13	$\beta_3{}^{(2)}$	0,087	$\gamma_3{}^{(2)}$	0,06
ი.ს. „ბრეგვაძე“	$\beta_4{}^{(1)}$	0,092	$\gamma_4{}^{(1)}$	0,12	$\beta_4{}^{(2)}$	0,102	$\gamma_4{}^{(2)}$	0,07
ასოციაცია „კომპლექსი“	$\beta_5{}^{(1)}$	0,095	$\gamma_5{}^{(1)}$	0,10	$\beta_5{}^{(2)}$	0,073	$\gamma_5{}^{(2)}$	0,09

მოსახლეობის ცალკეული ჯგუფებისთვის რაიონების მახასიათებელი პარამეტრები მოცემულია შემდეგ ცხრილში:

საწარმოს დასახელება	I რაიონი				II რაიონი			
	მნიშვნელობა		წონა		მნიშვნელობა		წონა	
	აღნიშვნა	სიდიდე	აღნიშვნა	სიდიდე	აღნიშვნა	სიდიდე	აღნიშვნა	სიდიდე
№1	$\lambda_1{}^{(1)}$	0,87	$\eta^{(1)}$	0,061	$\lambda_2{}^{(1)}$	0,79	$\eta^{(1)}$	0,058
№2	$\lambda_1{}^{(2)}$	0,83	$\eta^{(2)}$	0,068	$\lambda_2{}^{(2)}$	0,85	$\eta^{(2)}$	0,070
№3	$\lambda_1{}^{(3)}$	0,76	$\eta^{(3)}$	0,062	$\lambda_2{}^{(3)}$	0,81	$\eta^{(3)}$	0,063
№4	$\lambda_1{}^{(4)}$	0,72	$\eta^{(4)}$	0,092	$\lambda_2{}^{(4)}$	0,82	$\eta^{(4)}$	0,094
№5	$\lambda_1{}^{(5)}$	0,71	$\eta^{(5)}$	0,076	$\lambda_2{}^{(5)}$	0,72	$\eta^{(5)}$	0,078
№6	$\lambda_1{}^{(6)}$	0,85	$\eta^{(6)}$	0,071	$\lambda_2{}^{(6)}$	0,80	$\eta^{(6)}$	0,067

ინფორმაცია იმის შესახებ, თუ რა მოცულობის ინვესტიციის ათვისება შეუძლია მოცემულ საწარმოს (ინფორმაცია საწარმოს ტევადობის შესახებ) მოცემულია შემდეგ ცხრილში:

საწარმოს დასახელება	აღნიშვნა	ტევადობა
შ.პ.ს. „მუხა“	V_1	2.500.000
შ.პ.ს. „ალფა“	V_2	1.800.000
ი.ს. „ნადიბაიძე“	V_3	2.200.000
ი.ს. „ბრეგვაძე“	V_4	1.750.000
ასოციაცია „კომპლექსი“	V_5	2.500.000
გარე არე	V_0	3.500.000

$\sigma_{sm}-$ ით აღვნიშნავთ q-ს მნიშვნელობას რომლისთვისაც s და m საწარმოს არიან ბმულნი. $\sigma_{sm}-$ ის მნიშვნელობა მოცემულია შემდეგი მატრიცით:

$$\sigma=\begin{pmatrix}3&1&1&1&1\\1&3&2&1&2\\1&2&3&1&2\\1&1&1&2&1\\1&2&2&1&4\end{pmatrix}$$

$d_{im}R_m$ - ით აღვნიშნოთ m-ურ საწარმოს (სიმპლექსის) განზომილება. ჩვენს შემთხვევაში საწარმოს განზომილებათა ვექტორს აქვს სახე

(3, 3, 3, 2, 4)

გამოვთვალოთ მიზიდვის ფუნქციის კოეფიციენტი შემდეგი ფორმულით:

$$\delta_{sm}=(\sum_{p=1}^{p}\lambda_k{}^{(p)}\,\eta^{(P)}+\sum_{i=1}^{I}\beta_m{}^{(i)}\gamma_m{}^{(i)}+\sum_{i=1}^{I}\alpha_m{}^{(i)}\,\omega^{(i)})\frac{\sigma_{sm}}{d_{im}R_m}$$

(აქ K აღნიშნავს იმ რაიონის ნომერს, რომლესაც ეკუთვის m-ური საწარმო)

მიზიდვის კოეფიციენტები მოიცემა შემდეგი მატრიცით:

$$\delta=\begin{pmatrix}0{,}404&0{,}1417&0{,}1373&0{,}2215&0{,}2255\\0{,}1347&0{,}425&0{,}2746&0{,}2215&0{,}1126\\0{,}1347&0{,}2834&0{,}412&0{,}2215&0{,}2255\\0{,}1347&0{,}1417&0{,}1373&0{,}443&0{,}1126\\0{,}1347&0{,}2834&0{,}2746&0{,}2215&0{,}451\end{pmatrix}$$

$X_{sm}(t)$ აღნიშნავს ინვესტიციის სიდიდეს, რომელიც დროის T მომენტისათვის S საწარმოდან იდება m საწარმოში. ეს სიდიდეები აკმაყოფილებენ პირობას

$$\sum_{s=0}^{M} X_{sm}(t)=h_m \cdot I_m(t) \qquad m \in \overline{[1,M]}$$

h_m პროპორციულობის კოეფიციენტები მოცემულია ვექტორის სახით

$$h\ (0{,}82;\ 0{,}74;\ 0{,}62;\ 0{,}71;\ 0{,}68)$$

თუ ჩაწერის გამარტივების მიზნით $\alpha_m{}^{(i)}$-ის ნაცვლად დავწერთ α_m-ს, მაშინ ინვესტიციის სიდიდეების გამოსათვლელად გვექნება დიფერენციალური განტოლებების შემდეგი სისტემა

$$\frac{dI_m(t)}{dt} = (\alpha_m - \mu h_m) I_m(t) + \mu \sum_{s=0}^{M} \frac{\nu_{sm}(t)}{\sum_{k=0}^{M} \nu_{sm}(t)} h_s I_s(t)$$

სადაც ν_{sm} არის მიზისდვის ფუნქცია, რომელიც მოიცემა ფორმულით:

$$\nu_{sm}(t) = \delta_{sm}\ I_m(t)(V_m - I_m(t))$$

ყოველი $m \in \overline{[1,M]}$-სთვის (ჩვენს შემთხვევაში $M = 5$) ვპოულობთ $I_m(t)$-ს (m–ურ საწარმოში ჩადებული ჯამური ინვესტიციის სიდიდე დროის T მომენტისათვის). ვაგებთ შესაბამის გრაფიკს.

რეგიონში განხორციელებული ყოველი ცვლილება იწვევს საწყისი პარამეტრების ცვლილებას, რაც თავის მხრივ ცვლის I_m ფუნქციას. განიხილება რეგიონის რეკონსტრუქციის ორი გეგმა. თითოეული გეგმის შემთხვევაში ვითვლით ინვესტიციის მოსალოდნელ სიდიდეს, ვაფასებთ ობიექტების დატვირთულობას (მათ შორის მომწოდებლებისა და გასაღების ობიექტების დატვირთულობას), ვეძებთ ოპტიმალურ კავშირებს იმ შემთხვევაში, თუ სისტემა ვერ უძლებს შეშფოთებას.

საწარმოებს შორის კავშირის გაუმჯობესების ოპტიმალური გზის საპოვნელად ვპოულობთ მომწოდებლებსა და საწარმოებს შორის კავშირის ამსახველი ინციდენტურობას Ω მატრიცის სტრუქტურის ვექტორს. ამისათვის ვპოულობთ მატრიცას:

$$\Omega \cdot \Omega^{\mathrm{T}} - \mathrm{I} = \begin{pmatrix} 3 & 1 & 1 & 1 & 1 \\ 1 & 3 & 1 & 1 & 2 \\ 1 & 1 & 3 & 1 & 2 \\ 1 & 1 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 & 4 \end{pmatrix}$$

სადაც Iარის მატრიცი, რომლის ყოველი ელემენტი 1-ის ტოლია. ვპოულობთ შესაბამის სტრუქტურის ვექტორის კოორდინატებს:

როცა $q = 4$ $\quad Q_4 = 1$

როცა $q = 3$ $\quad Q_3 = 4$

როცა $q = 2$ $\quad Q_2 = 3$

როცა $q = 1$ $\quad Q_1 = 1$

როცა $q = 0$ $\quad Q_0 = 1$

სტრუქტურის ვექტორია (1; 4; 3; 1; 1). თუ ერთმანეთთან დავაკავშირებთ №3 მომწოდებელსა და ი.ს. „ბრეგვაძეს“, მაშინ შესაბამისი სტრუქტურის ვექტორი იქნება (2,4,1,1,1) და ეს კავშირი არის ოპტიმალური.

იმისათვის, რომ პასუხი გავცეთ კითხვას, გაუძლებს თუ არა მომწოდებლებსა და საწარმოებს შორის კავშირის ქსელი დატვირთვას ამა თუ იმ გეგმის შემთხვევაში, გვჭირდება ვიცოდეთ თუ მაქსიმუმ რა მოცულობის ნედლეულის მოწოდება შეუძლია ამათ იმ მომწოდებელს; რა მოცულობის ნედლეულია საჭირო ამა თუ იმ საწარმოსათვის; როგორია თითოეული საწარმოს წილი მომწოდებლის მიერ მოწოდებულ ნედლეულის საერთო რაოდენობაში; როგორია თითოეული მომწოდებლის წილი საწარმოს მიერ მოხმარებული ნედლეულის რაოდენობაში.

ინფორმაცია მომწოდებლების დატვირთულობის შესახებ მოცემულია შემდეგ ცხრილში:

ნედლეულის მომწოდებელი	სულ რა რაოდენობის ნედლეულის მიწოდებაა შესაძლებელი			შ.პ.ს. „მუხა“ F_1=2.542.000				შ.პ.ს. „ალფა“ F_2=1.074.000				ი.ს. „ნადიბაიძე“ F_3=1.732.500				ი.ს. „ბრეგვაძე“ F_4=1.242.000				ასოციაცია „კომპლექსი“ F_5=1.000.000			
	აღნიშვნა	რაოდენობა	თანხა (ლარი)	აღნიშვნა	მომწოდებლის წილი	რაოდენობა	თანხა	აღნიშვნა	მომწოდებლის წილი	რაოდენობა	თანხა	აღნიშვნა	მომწოდებლის წილი	რაოდენობა	თანხა	აღნიშვნა	მომწოდებლის წილი	რაოდენობა	თანხა	აღნიშვნა	მომწოდებლის წილი	რაოდენობა	თანხა
№1	M_1	10.000	1.500.000	y_{11}	0.047	800	120.000	y_{12}	0.335	2.400	360.000	y_{13}	0.069	800	120.000	y_{14}	0.087	720	108.000	y_{15}	0.492	3280	492.000
№2	M_2	8.000	1.200.000	y_{21}	0.177	3.000	450.000	y_{22}	0	0	0	y_{23}	0.030	3.500	52.500	y_{24}	0	0	0	y_{25}	0	0	0
№3	M_3	15.000	2.250.000	y_{31}	0	0	0	y_{32}	0	0	0	y_{33}	0.485	5.600	840.000	y_{34}	0.599	4960	744.00	y_{35}	0.216	1440	216.000
№4	M_4	8.000	1.200.000	y_{41}	0.118	2.000	300.000	y_{42}	0.196	1.400	210.000	y_{43}	0	0	0	y_{44}	0.314	2.600	390.000	y_{45}	0	0	0
№5	M_5	3.000	450.000	y_{51}	0	0	0	y_{52}	0.112	800	120.000	y_{53}	0.416	480	72.000	y_{54}	0	0	0	y_{55}	0.108	720	108.000
№6	M_6	4.000	600.000	y_{61}	0	0	0	y_{62}	0.357	2.560	384.000	y_{63}	0	0	0	y_{64}	0	0	0	y_{65}	0.096	640	96.000
№7	M_7	15.000	2.250.000	y_{71}	0.658	11.147	1.672.000	y_{72}	0	0	0	y_{73}	0	0	0	y_{74}	0	0	0	y_{75}	0.088	587	88.000

ინფორმაცია გასაღების მაზრის დატვირთულობის შესახებ.

გასაღების ობიექტი	სულ რა მოცულობის საქონლის გასაღებაა შესაძლებელი			შ.პ.ს. „მუხა“ E_1=3.668.000				შ.პ.ს. „ალფა“ E_2=1.720.000				ი.ს. „ნადიბაიძე“ E_3=2.685.000				ი.ს. „ბრეგვაძე“ E_4=1.987.000				ასოციაცია „კომპლექსი“ E_5=1.710.000			
	აღნიშვნა	რაოდენობა	თანხა	აღნიშვნა	გასაღების ობიექტის წილი	რაოდენობა	თანხა	აღნიშვნა	გასაღების ობიექტის წილი	რაოდენობა	თანხა	აღნიშვნა	გასაღების ობიექტის წილი	რაოდენობა	თანხა	აღნიშვნა	გასაღების ობიექტის წილი	რაოდენობა	თანხა	აღნიშვნა	გასაღების ობიექტის წილი	რაოდენობა	თანხა
№1	G_1	6.000	2.400.000	Z_{11}	0.404	3700	1.480.000	Z_{12}	0.116	500	200.000	Z_{13}	0.045	300	120.000	Z_{14}	0	0	0	Z_{15}	0.117	500	200.000
№2	G_2	5.000	2.000.000	Z_{21}	0.282	2590	1.036.000	Z_{22}	0	0	0	Z_{23}	0.032	2125	85.000	Z_{24}	0.144	7175	287.000	Z_{25}	0.112	480	195.000
№3	G_3	4.000	1.600.000	Z_{31}	0.162	930	372.000	Z_{32}	0.070	300	120.000	Z_{33}	0	0	0	Z_{34}	0.252	1250	500.000	Z_{35}	0.145	620	248.000
№4	G_4	2.500	1.000.000	Z_{41}	0.016	150	60.000	Z_{42}	0.291	1250	500.000	Z_{43}	0.037	250	100.000	Z_{44}	0.050	250	100.000	Z_{45}	0.023	100	40.000
№5	G_5	2.500	1.000.000	Z_{51}	0.034	310	124.000	Z_{52}	0.291	1250	500.000	Z_{53}	0.042	280	112.000	Z_{54}	0	0	0	Z_{55}	0.038	160	64.000
№6	G_6	4.500	1.800.000	Z_{61}	0	0	0	Z_{62}	0	0	0	Z_{63}	0.067	450	180.000	Z_{64}	0.554	2925	1.100.000	Z_{65}	0.263	1125	450.000
№7	G_7	3.000	1.200.000	Z_{71}	0.162	1490	596.000	Z_{72}	0.232	1000	400.000	Z_{73}	0	0	0	Z_{74}	0	0	0	Z_{75}	0.119	510	204.000
№8	G_8	6.000	2.400.000	Z_{81}	0	0	0	Z_{82}	0	0	0	Z_{83}	0.777	5200	2.088.000	Z_{84}	0	0	0	Z_{85}	0.183	780	312.000

ექსცენტრისიტეტის გამოსათვლელად ვსარგებლობთ ფორმულით:

$$E_{cc_m} = \frac{d_{im}R_m - \check{q}}{\check{q} + 1}$$

სადაც $\check{q}$ აღნიშნავს q-ს უდიდეს მნიშვნელობას, რომლისთვისაც m–ური საწარმო q ბმულია ყველა დანარჩენ საწარმოსთან. ამიტომ მომწოდებლებთან არსებული კავშირების შემთხვევაში საწარმოების ექსცენტრისიტეტი იქნება (ამ შემთხვევაშიც $\check{q}$=1)

საწარმოს დასახელება	ექსცენტრისიტეტის მნიშვნელობა
შ.პ.ს. „მუხა“	1,00
შ.პ.ს. „ალფა“	1,00
ი.ს. „ნადიბაიძე“	1,00
ი.ს. „ბრეგვაძე“	0,50
ასოციაცია „კომპლექსი“	1,50

გეგმა 1-ის შემთხვევაში საწარმოთა ექსცენტრისიტეტი არ იცვლება.

გეგმა 2-ის შემთხვევაში არსებული ქსელი დატვირთვას ვერ გაუძლებს. საჭიროა შ.პ.ს. „მუხა“ მომარაგდეს №5 მომწოდებლითაც, ხოლო ასოციაცია „კომპლექსი“ – №4 მომწოდებლითაც. ამ შემთხვევაში ინციდენტურობის მატრიცას ექნება სახე:

$$\Omega = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$C = \Omega \cdot \Omega^{\mathrm{T}} = \begin{pmatrix} 5 & 3 & 2 & 2 & 4 \\ 3 & 4 & 3 & 2 & 4 \\ 2 & 3 & 4 & 2 & 3 \\ 2 & 2 & 2 & 3 & 3 \\ 4 & 4 & 3 & 3 & 6 \end{pmatrix}$$

$$\Omega \cdot \Omega^{\mathrm{T}} - \mathrm{I} = \begin{pmatrix} 4 & 2 & 1 & 1 & 3 \\ 2 & 3 & 2 & 1 & 3 \\ 1 & 2 & 3 & 1 & 2 \\ 1 & 1 & 1 & 2 & 2 \\ 3 & 3 & 2 & 2 & 5 \end{pmatrix}$$

შესაბამისი სტრუქტურის ვექტორის კოორდინატებია

როცა $q = 5$ $Q_5 = 1$ {№5}

როცა $q = 4$ $Q_4 = 4$ {№1}, {№5}

როცა $q = 3$ $Q_3 = 3$ {№1; №2; №5} {№3}

როცა $q = 2$ $Q_2 = 1$ {№1; №2; №3; №4; №5}

როცა $q = 1$ $Q_1 = 1$ {№1; №2; №3, №4, №5}

როცა $q = 0$ $Q_0 = 1$ {№1; №2; №3; №4; №5}

ამ შემთხვევაში $\breve{q}$=2.

საწარმოთა ექსცენტრისიტეტები მოიცემა ცხრილით:

საწარმოს დასახელება	ექსცენტრისიტეტის მნიშვნელობა
შ.პ.ს. „მუხა“	0.67
შ.პ.ს. „ალფა“	0.33
ი.ს. „ნადიბაიძე“	0.33
ი.ს. „ბრეგვაძე“	0
ასოციაცია „კომპლექსი“	1.00

π მოდელი და მოდელის ნაზრდი სხვადასხვა გეგმის შემთხვევაში

	მოდელი	შ.პ.ს. „მუხა“	შ.პ.ს. „ალფა“	ი.ს. „ნადიბაიძე“	ი.ს. „ბრეგვაძე“	ასოციაცია „კომპლექსი“	სულ
არსებულ პირობებში	π	3.668.000	1.740.000	2.700.000	2.050.000	1.800.000	11.958.000
გეგმა 1	π_1	3.820.000	1.770.000	2.685.000	2.125.000	1.850.000	12.250.000
	$\delta\pi_1$	152.000	30.000	-15.000	75.000	50.000	292.000
გეგმა 2	π_2	3.770.000	1.810.000	2.720.000	2.140.000	1.900.000	12.340.000
	$\delta\pi_2$	102.000	70.000	20.000	90.000	100.000	382.000

ექსპერტული სისტემა საინვესტიციო

პოლიტიკის შემუშავებისთვის

მისი დანიშნულებაა რეგიონების რეკონსტრუქციის პროექტების შეფასება.

- <u>პროგრამის ფუნქციები:</u>

პროგრამის ჩართვისას ეკრანზე გამოდის შემდეგი ფანჯარა

მენიუს ჩანართი „ზოგადი ინფორმაცია“:

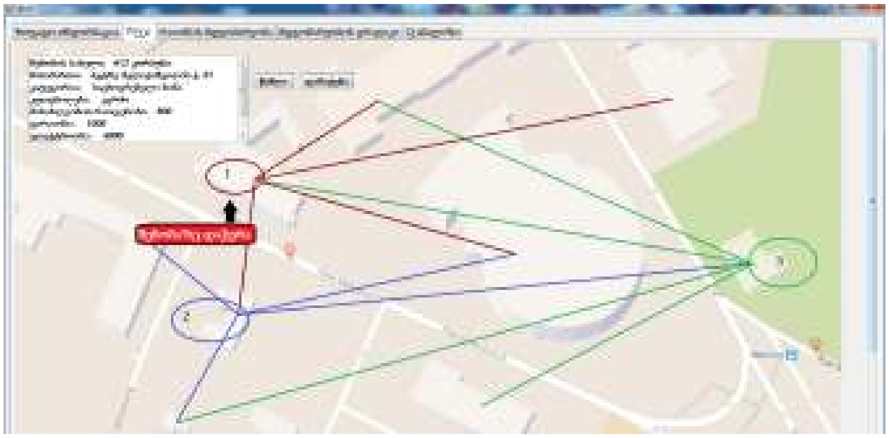
ჩანართი მოიცავს სხვადასხვა მონაცემებს რეგიონის შესახებ.მაგალითად, რეგიონში რაიონების რაოდენობა, რაიონებში მოსახლეობის რაოდენობა, რაიონების მახასიათებელი პარამეტრები, რეგიონში საწარმოების რაოდენობა, რეგიონის მახასიათებელი პარამეტრის მნიშვნელობა, მომწოდებლებსა და საწარმოებს შორის, ასევე საწარმოებსა და წარმოებული პროდუქციის გასაღების ობიექტებს შორის კავშირის შესაბამისი ინციდენტურობის მატრიცი.

მენიუს ჩანართი „რეგიონის რუკა“:

ეკრანზე გამოდის რუკა რომელზეც დატანილია სხვადასხვა ობიექტები: საცხოვრებელი კორპუსები ,

საწარმოები, მაღაზიები, ქუჩები და ა.შ. თითოეული ობიექტის შესახებ ინფორმაცია შენახულია მონაცემთა ბაზაში.

➔



პროგრამას ასევე შეუძლია(რუკაზე გადასვლისას) ობიექტის მონიშვნისას ეკრანზე გამოიტანოს ამ ობიექტის მახასიათებლები , ის ასევე გვიჩვენებს მონიშნული ობიექტის კკავშირს დანარჩენ ობიექტბთან მაგ. შენობის კავშირს მაღაზიასთაან , ბანკთან , სკოლასთან და ა.შ.

ასევე შესაძლებელია რუკაზე დატანილი ობიექტების წაშლა

სურათი 1:

სურათი 2:

<u>შედეგი:</u>

<u><==</u>

და ახლის დამეტება.

სურათი 1:

სურათი 2:

სურათი 3:

სურათი 4:

შედეგი

<==

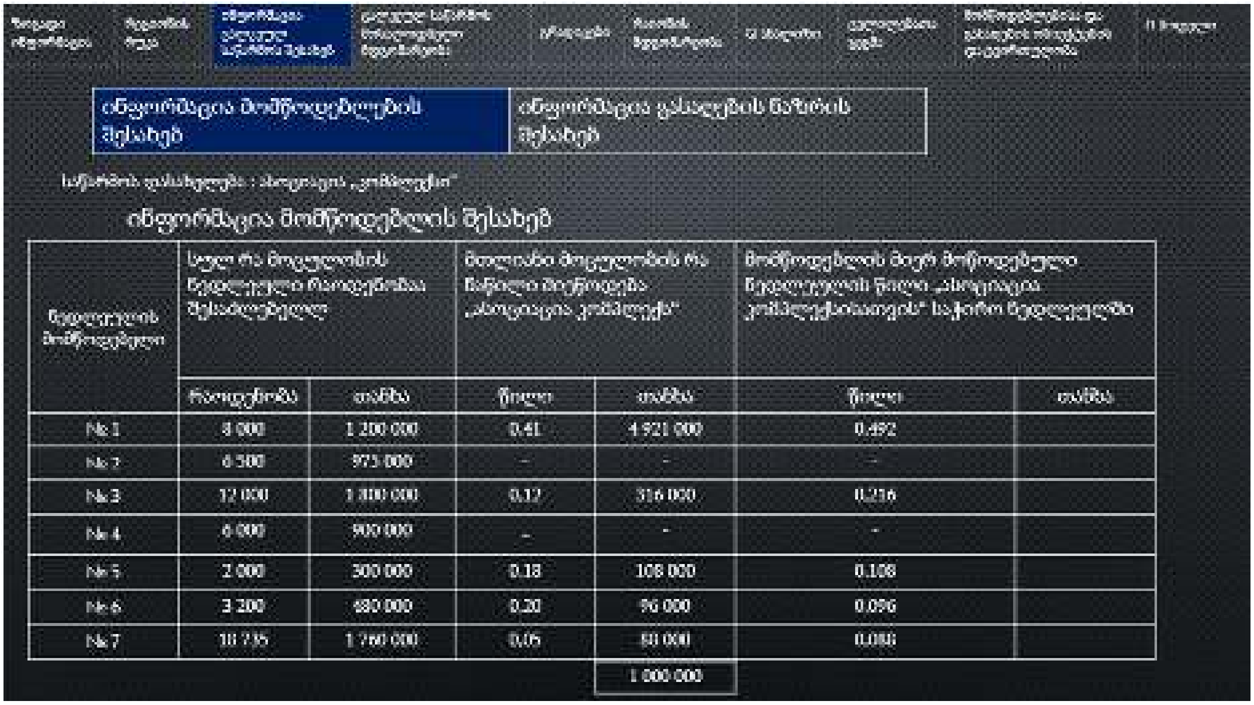
მენიუს ჩანართი „ინფორმაცია ცალკეულ საწარმოს შესახებ“:

ჩანართი „ინფორმაცია ცალკეულ საწარმოს შესახებ“ გვაწვდის სხვადასხვა ინფორმაციას ჩვენს მიერ არჩეული საწარმოს შესახებ, მაგალითად, ინფორმაციას საწარმოს ტევადობის შესახებ, წინა წლებში განხორციელებული ინვესტიციების შესახებ, წლების მიხედვით საწარმოს მოგების სიდიდისა და მოგების კოეფიციენტების შესახებ, წლების მიხედვით გადაიარაღების კოეფიციენთების სიდიძის შესახებ, საწარმოს ექსცენტრისიტეტის შესახებ.

ამავე ჩანართზე განთავსებულია 2 ფანჯარა: 1) და 2) ინფორმაცია გასაღების ბაზრის შესახებ.

ფანჯარაზე „ინფორმაცია მომწოდებლების შესახებ “ დაწკაპუნების შემდეგ ეკრანზე

გამოდის შემდეგი სურათი:



ნედლეულის მომწოდებელი	სულ რა მოცულობის ნედლეული რაოდენობაა შესაძლებელი		მთლიანი მოცულობის რა ნაწილი მიეწოდება „ალიკაცია კომპლექს“		მომწოდებლის მიერ მოწოდებული ნედლეულის წილი „ალიკაცია კომპლექსისათვის“ საჭირო ნედლეულში	
	რაოდენობა	თანხა	წილი	თანხა	წილი	თანხა
№ 1	8 000	1 200 000	0.41	4 921 000	0.492	
№ 2	6 500	975 000	-	-	-	
№ 3	12 000	1 800 000	0.12	316 000	0.216	
№ 4	6 000	900 000	–	-	-	
№ 5	2 000	300 000	0.18	108 000	0.108	
№ 6	3 200	480 000	0.20	96 000	0.096	
№ 7	18 735	1 760 000	0.05	80 000	0.088	
				1 000 000		

მასზე მოცემულია ინფორმაცია იმის შესახებ, თუ რომელი მომწოდებელი რა მოცულობის ნედლეულს აწვდის საწარმოს, როგორია ცალკეული მომწოდებლის მიერ საწარმოსთვის მოწოდებული ნედლეულის წილი მომწოდებლის მიერ გაცემული ნედლეულის მთლიან მოცულობაში და როგორია მომწოდებლის ნედლეულის წილი საწარმოსთვის საჭირო ნედლეულის მთლიან მოცულობაში

ფანჯარაზე „ინფორმაცია გასაღების ბაზრის შესახებ “ დაწკაპუნების შემდეგ ეკრანზე გამოდის შემდეგი სურათი:

გასაღების ობიექტი	სულ რა მოცულობის საქონლის გასაღებაა შესაძლებელი		„ალიკაცია კომპლექსის" მიერ წარმოებული საქონლის რა ნაწილის რეალიზაცია ხდება მოცემულ ობიექტზე		ობიექტის მიერ რეალიზებული პროდუქტის რა ნაწილს შეადგენს „ალიკაცია კომპლექსის" მიერ მოწოდებული საქონელი
	რაოდენობა	თანხა			
№ 1	50 000	2 000 000	0,10	200 000	0,114
№ 2	40 000	1 600 000	0,12	192 000	0,112
№ 3	31 000	1 240 000	0.20	248 000	0.145
№ 4	20 000	800 000	0,05	40 000	0,023
№ 5	20 000	800 000	0,08	64 000	0,037
№ 6	45 000	1 800 000	0,25	450 000	0,263
№ 7	30 000	1 200 000	0.17	204 000	0,119
№ 8	60 000	2 400 000	0,13	312 000	0,182
				1 710 000	

მასზე მოცემულია ინფორმაცია იმის შესახებ, თუ მოცემული საწარმოს მიერ წარმოებული საქონლის რა რაოდენობა და რა ნაწილის რეალიზაცია ხდება მოცეულ ობიექტზე, ასევე ინფორმაცია იმის შესახებ, თუ ობიექტის მიერ რეალიზებული მთლიანი საქონლის პროდუქტის რა ნაწილს შეადგენს მოცემული საწარმოს მიერ მიერ მოწოდებული საქონელი.

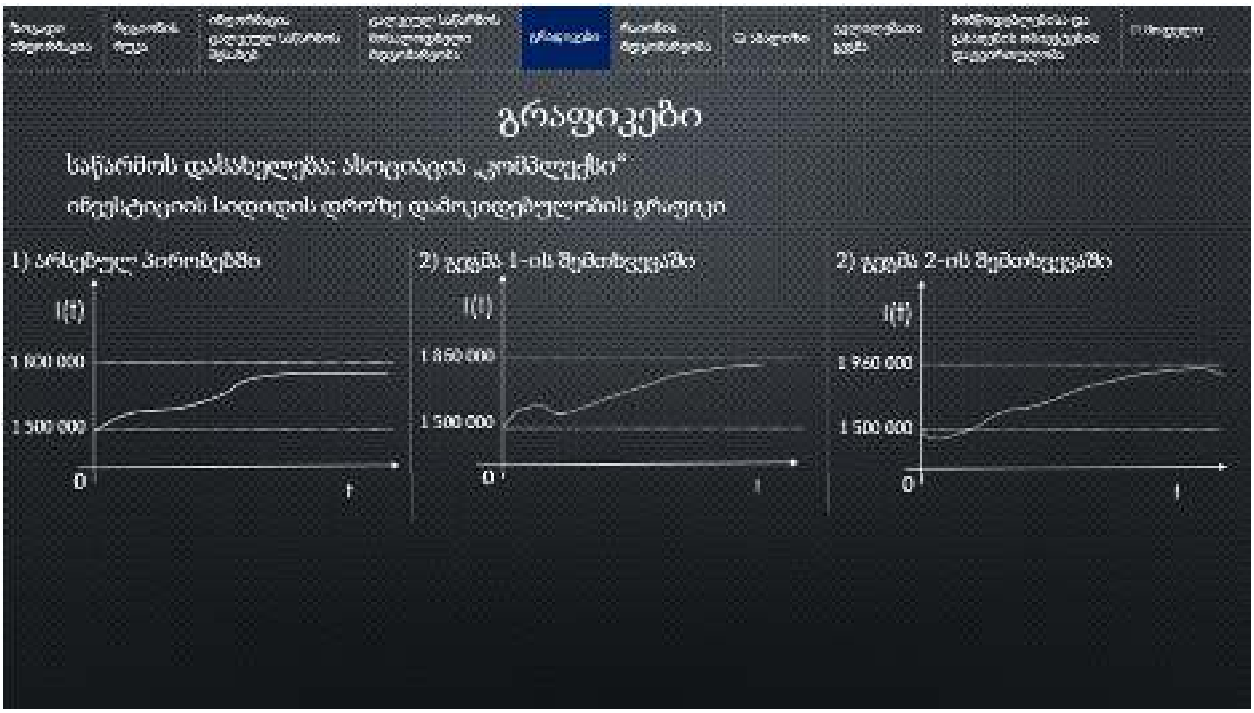
	მოსალოდნელი ინვესტიცია	წარმოებული პროდუქციის მოსალოდნელი მოცულობა	მოგების მოსალოდნელი სიდიდე
არსებული მონაცემებით	1 800 000	2 034 000	234 000
ვარიანტი 1-ის შემთხვევაში	1 850 000	2 090 500	240 000
ვარიანტი 2-ის შემთხვევაში	1 900 000	1 166 000	266 000

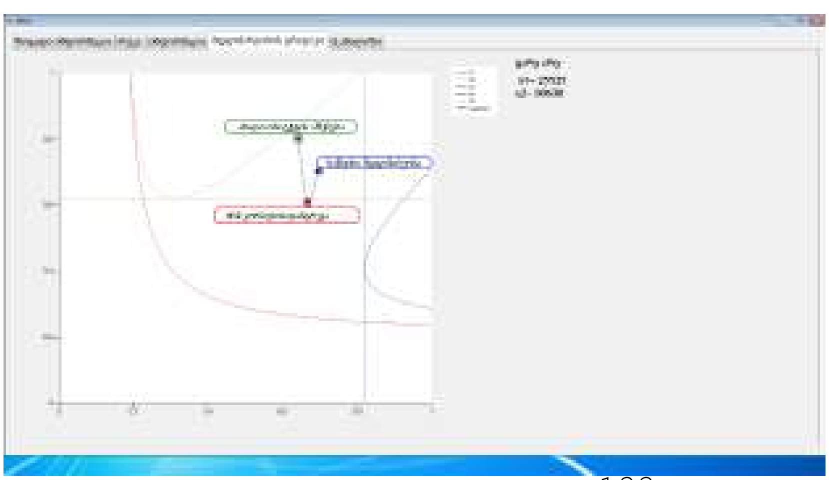
მენიუს ჩანართი „ცალკეული საწარმოს მოსალოდნელი მდგომარეობა“:

მენიუს ჩანართი „ცალკეული საწარმოს მოსალოდნელი მდგომარეობა“ გვაწვდის ინფორმაციას ცალკეულ საწარმოში განხორციელებული ინვესტიციების, წარმოებული პროდუქციისა და მოგების მოსალოდნელი სიდიდის შესახებ სხვადასხვა გეგმის შემთხვევაში, რაც საშუალებას გვაძლევს იმიტაციურად განვახორციელოთ რამოდენიმე გენ გეგმა.

მენიუს ჩანართი „გრაფიკები“:

პროგრამა საშუალებას იძლევა განხორციელებული გეგმის შესაბამისი ინფორმაცია გამოსახოს გრაფიკულად. ➔



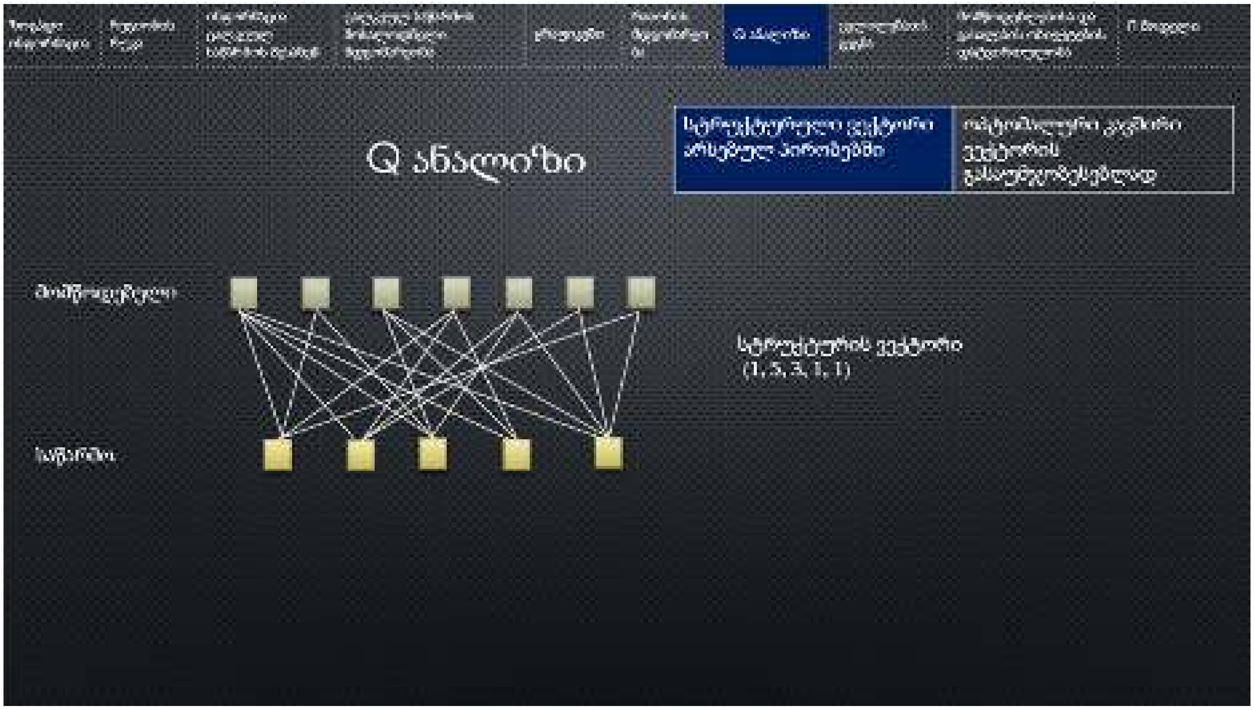


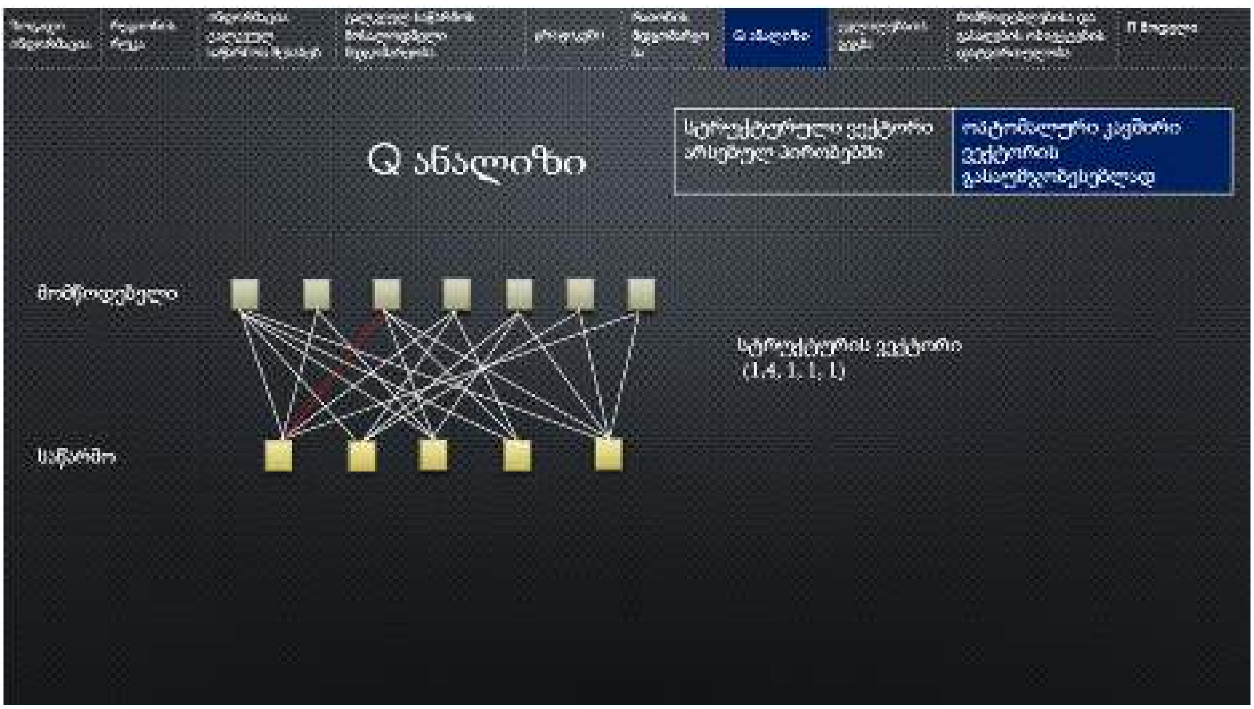
მენიუს ჩანართი „რაიონის მდგომარეობა“:

➔

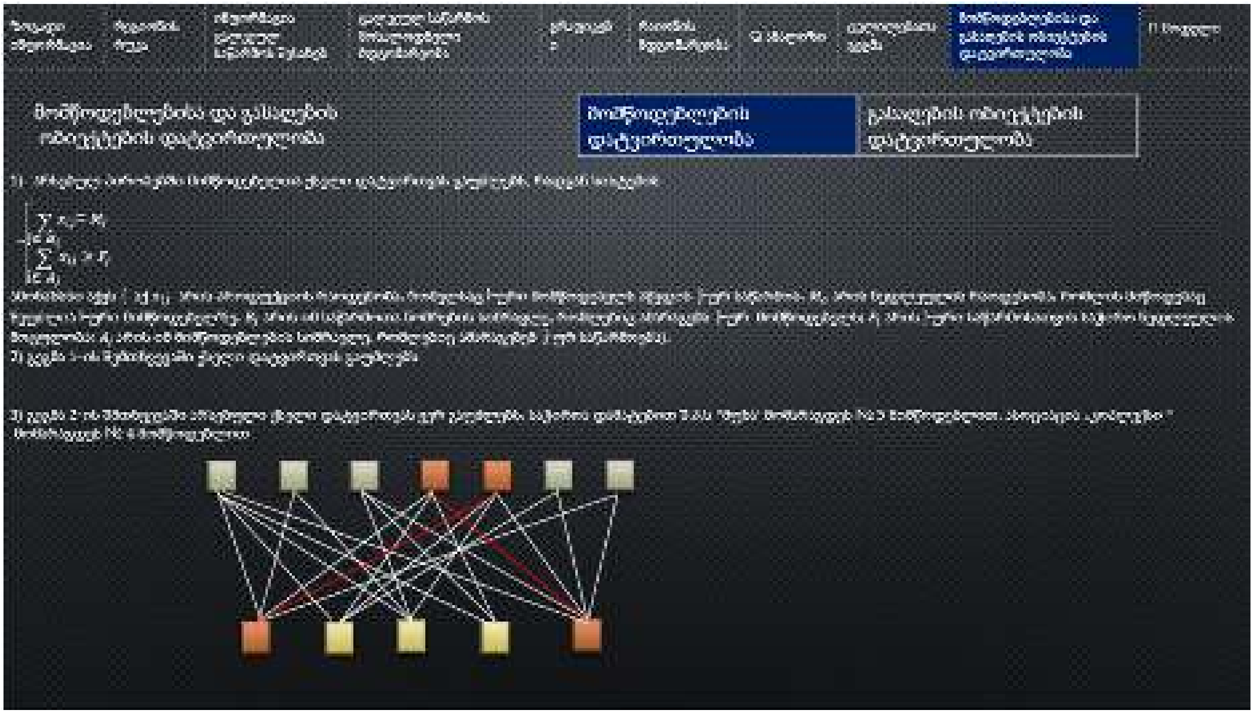
პროგრამას აქვს მენიუს ჩანართი რაიონის მდგომარეობა, რომელიც მოიცავს ინფორმაციას რაიონის ამჟამინდლი მდგომარეობის შესახებ.

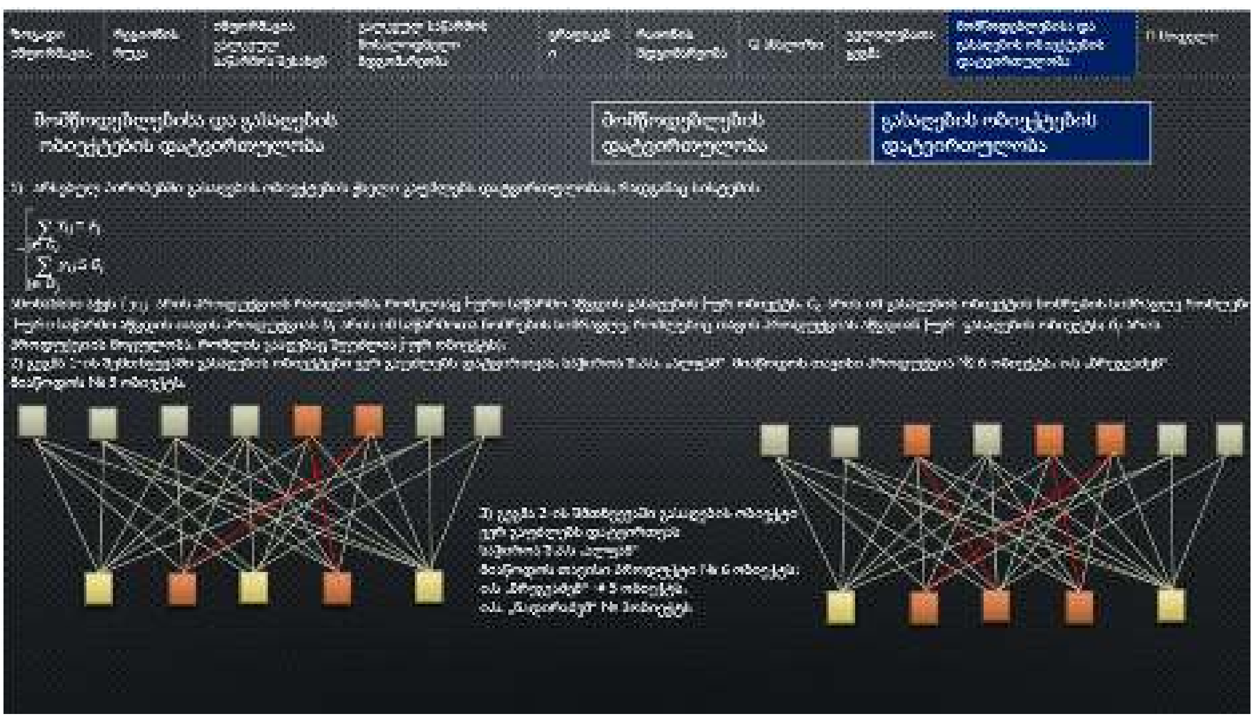
პროგრამას ასევე შეუძლია Q ანალიზის ჩატარება. Q ანალიზი ასახავს ობიექტების დამოკიდებულებას ერთმანეთთან. პროგრამა საშუალებას იძლევა ვიპოვოთ ოპტიმალური კავშირი სტრუქტურის ვექტორის გასაუმჯობესებლად.


Q ანალიზი
სტრუქტურული ვექტორი არსებულ პირობებში
ოპტიმალური კავშირი ვექტორის გაუმჯობესებლად
მომწოდებელი
საწარმო
სტრუქტურის ვექტორი
(1, 5, 3, 1, 1)


Q ანალიზი
სტრუქტურული ვექტორი არსებულ პირობებში
ოპტიმალური კავშირი ვექტორის გაუმჯობესებლად
მომწოდებელი
საწარმო
სტრუქტურის ვექტორი
(1,4, 1, 1, 1)

მენიუს ჩანართი „მომწოდებლებისა და გასაღების ობიექტების დატვირთულობა“:

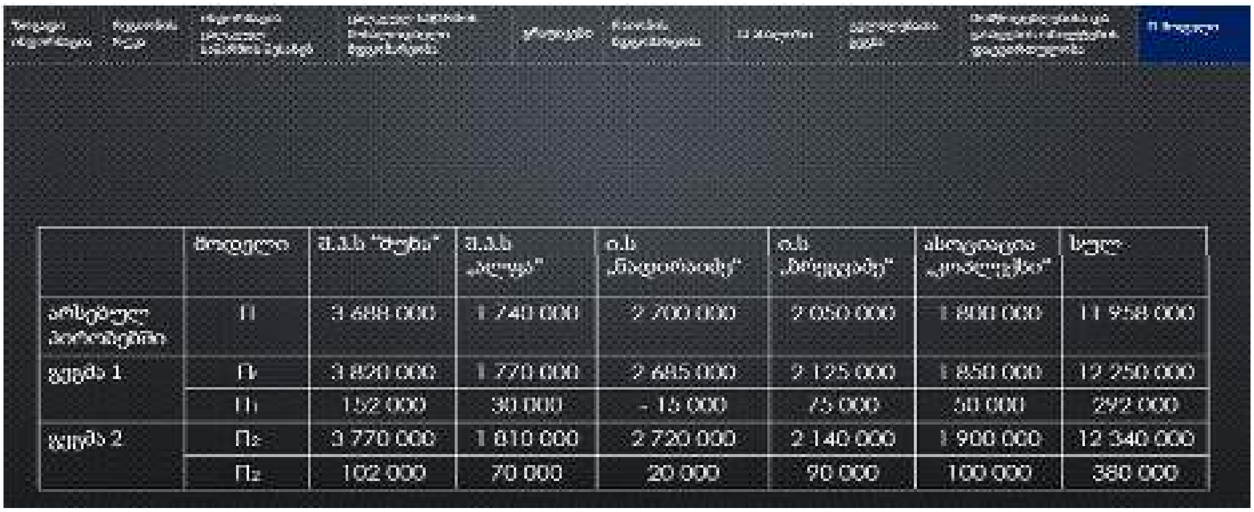




მენიუს ჩანართი „მომწოდებლებისა და გასაღების ობიექტების დათვირთულობა“ გვაწვდის ინფორმაციას იმის შესახებ, გაუძლებს თუ ვერა სხვადასხვა ობიექტები დატვირთულობას ამა თუ იმ გეგმის განხორციელების შემთხვევაში.

იმ შემთხვევაში, თუ რომელიმე ობიექტი ვერ გაუძლებს დატვირთვას, ჩანართი გვაწვდის ინფორმაციას დატვირთვის გადანაწილების ოპტიმალური ვარიანტის შესახებ.

მენიუს ჩანართი „П მოდელი“:



	მოდელი	შ.პ.ს [illegible]	შ.პ.ს [illegible]	ი.მ [illegible]	ი.მ [illegible]	[illegible]	სულ
[illegible]	П	3 688 000	1 740 000	2 700 000	2 050 000	1 800 000	11 958 000
გეგმა 1	$П_1$	3 820 000	1 770 000	2 685 000	2 125 000	1 850 000	12 250 000
	$П_1$	152 000	30 000	- 15 000	75 000	50 000	292 000
გეგმა 2	$П_2$	3 770 000	1 810 000	2 720 000	2 140 000	1 900 000	12 340 000
	$П_2$	102 000	70 000	20 000	90 000	100 000	380 000

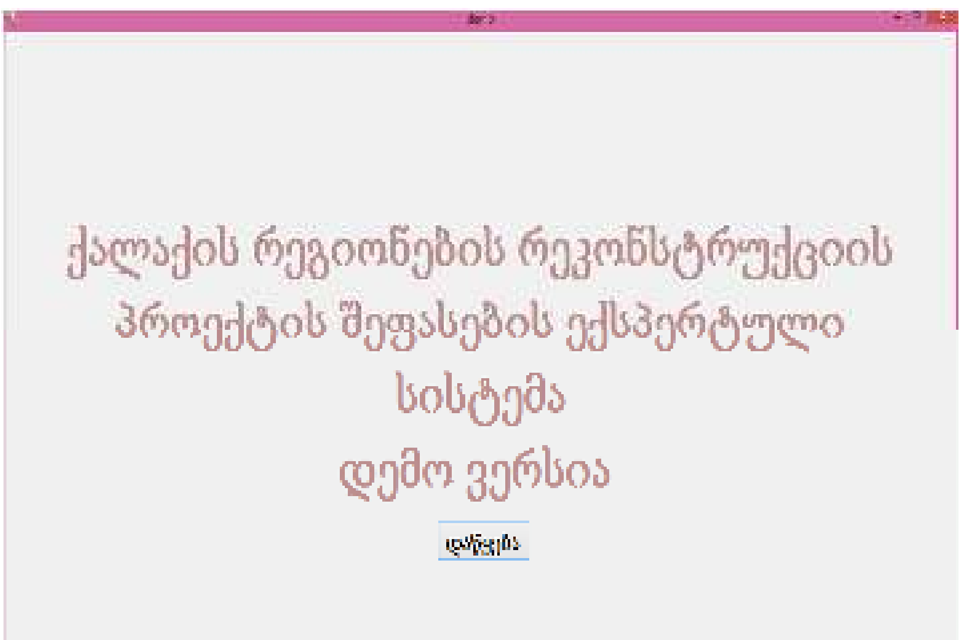
მენიუს ჩანართი „П მოდელი“ გვაწვდის ინფორმაციას მოდელისა და მოდელის ნაზრდის შესახებ სხვადასხვა გეგმის შემთხვევაში.

ქალაქის რეგიონების რეკონსტრუქციის პროექტის შეფასების ექსპერტული სისტემა

მისი დანიშნულებაა ქალაქის რეგიონების რეკონსტრუქციის პროექტების შეფასება.

- პროგრამის ფუნქციები:

პროგრამის ჩართვისას ეკრანზე გამოდის შემდეგი ფანჯარა



შემდეგ ვაწვებით ღილაკს „დაწყება“.

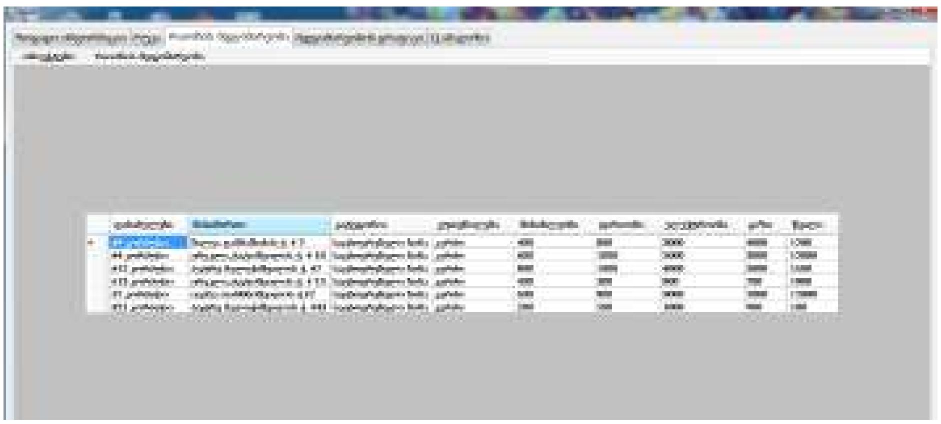
➔

ეკრანზე გამოდის რუკა რომელზეც დატანილია სხვადასხვა ობიექტები , რომლებიცაა საცხოვრებელი კორპუსები , მაღაზიები, ქუჩები და ა.შ. თითოეული ობიექტის შესახებ ინფორმაცია შენახულია მონაცემთა ბაზაში.

➔

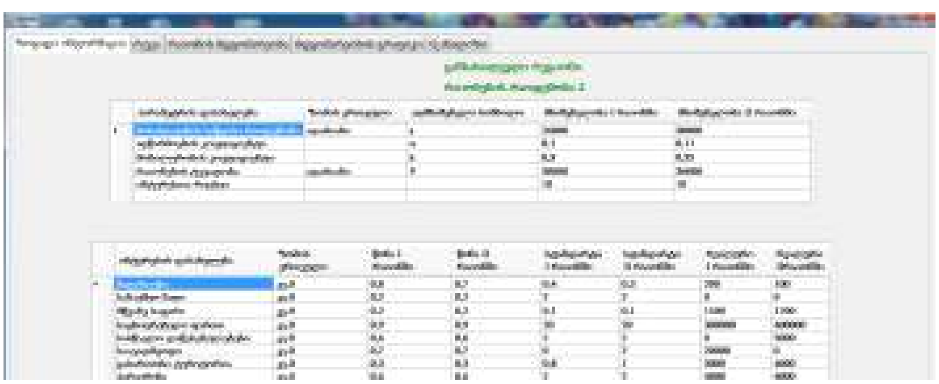
პროგრამას აქვს მენიუს ჩანართი რაიონის მდგომარეობა, რომელიც მოიცავს ინფორმაციას ობიექტებისა და რაიონის ამჟამინდლი მდგომარეობის შესახებ , ობიექტების შესახებ ინფორმაცია გულისხმობს რუკაზე არსებული ობიექტების კატეგირიებად დაყოფას.

მაგ: საგანმანათლებლო დაწერებულებები, ქუჩები, სასტუმროები, კორპუსები და ა.შ.



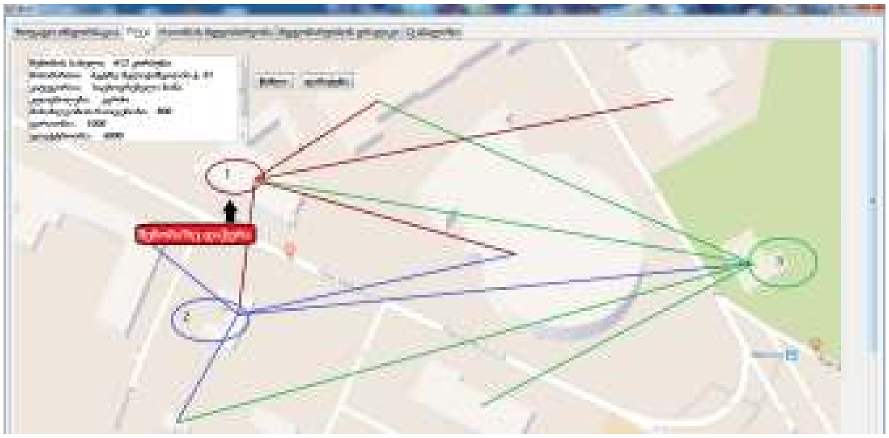
თითოეულის არჩევისას ცხრილის სახით გამოდის მონიშნულის შესაბამისი ინფორმაცია.

➔



მენიუს ჩანართი „ზოგადი ინფორმაცია“ მოიცავს სხვადასხვა მონაცემებს. მაგ. მოცემულია რამდენიმე ინტერესი, თითოეული ინტერესი ხასიათდება პარამეტრებით: ზომის ერთეული, წონა, სტანდარტი და სხვ.

➔



პროგრამას ასევე შეუძლია(რუკაზე გადასვლისას) ობიექტის მონიშვნისას ერანზე გამოიტანოს ამ ობიექტის მახასიათებლები , ის ასევე გვიჩვენებს მონიშნული ობიექტის კკავშირს დანარჩენ ობიექტბთან მაგ. შენობის კავშირს მაღაზიასთაან , ბანკთან , სკოლასთან და ა.შ.

ასევე შესაძლებელია რუკაზე დატანილი ობიექტების წაშლა

სურათი 1:

სურათი 2:

შედეგი:

<==

და ახლის დამეტება.

სურათი 1:

სურათი 2:

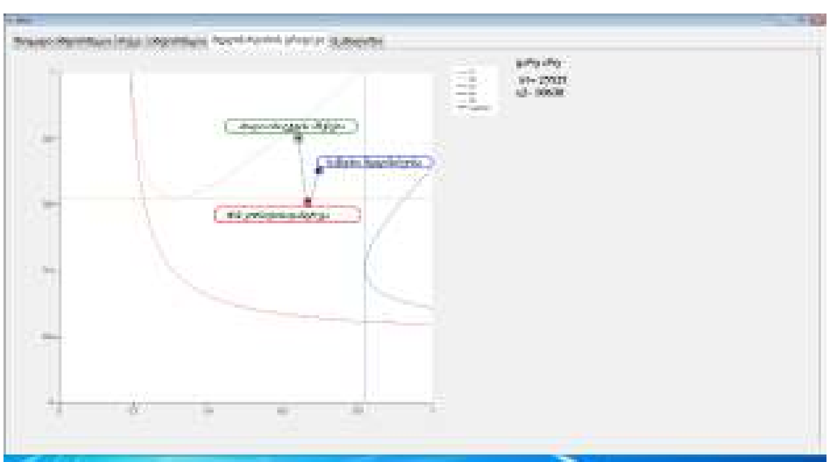
სურათი 3:

სურათი 4:

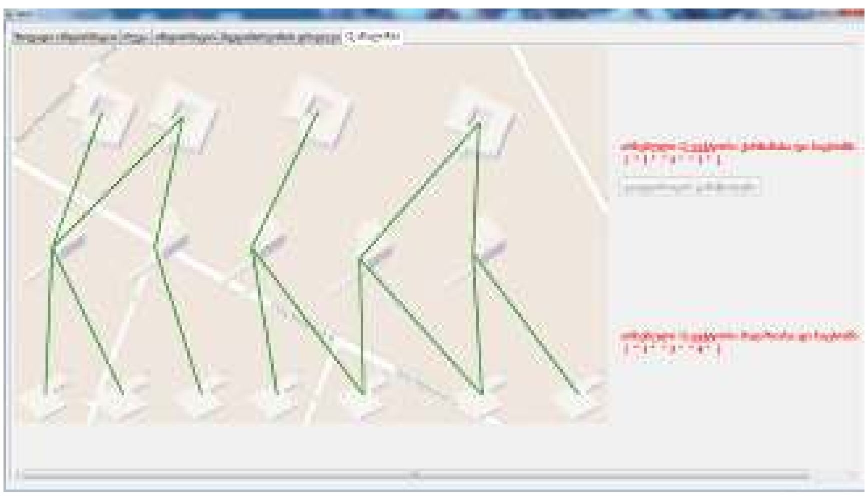
შედეგი

<==

რაც საშუალებას გვაძლევს იმიტაციურად განვახორციელოთ რამოდენიმე გენ გეგმა. პროგრამა საშუალებას იძლევა განხორციელებული გეგმის შესაბამისი ინფორმაცია გამოსახოს გრაფიკულად. ➔



პროგრამას ასევე შეუძლია Q ანალიზის ჩატარება. Q ანალიზი ასახავს ობიექტების დამოკიდებულებას ერთმანეთთან. მაგ: ერთმანეთთან დაკავშირებული საცხობის , მაღაზიისა და ფქვილის საწყობის კავშირს .Q ანალიზი საშუალებას იძლევა გადატვირთულ ობიექტს გაუკეთოს ოპტიმიზაცია მაგ. ერთ-ერთი მაღაზია რომლსაც მოსახლეობის გაზრდის გამო დატვირთულობა გაეზარდა , მოითხოვს დამატებით მომარაგებას ეს პროგრამა კი ავტომატრად ახდენს დატვირთულობის გადანაწილებას Q ანალიზის საშუალებით.

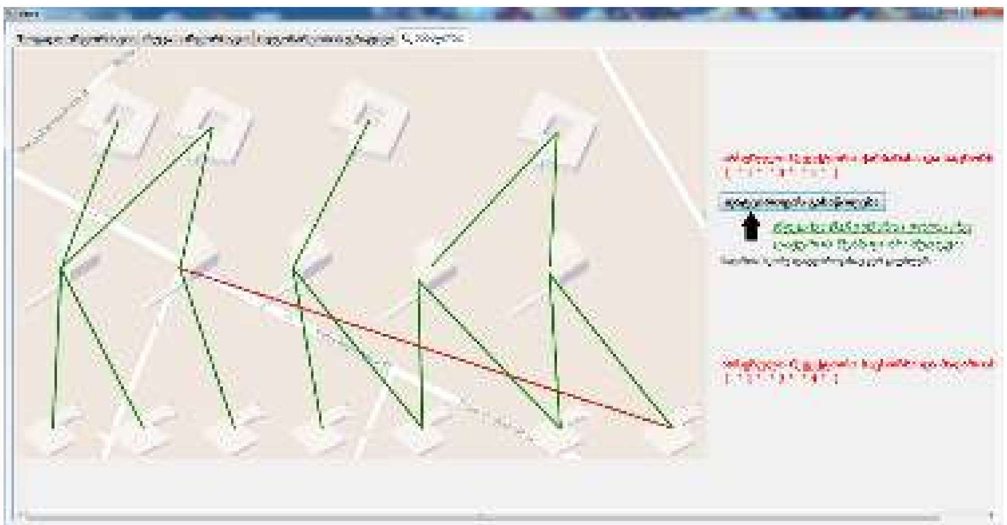


სურათი 1:

ვაწვებით ერთ-ერთ მაღაზიას იმის გასაგებად, გაუძლებს თუ არა ეს მაღაზია დატვირთვას, როგორც ვხედავთ

სურათი 2:

როგორც ვხედავთ მაღაზია დატვირთვას ვერ გაუძლებს

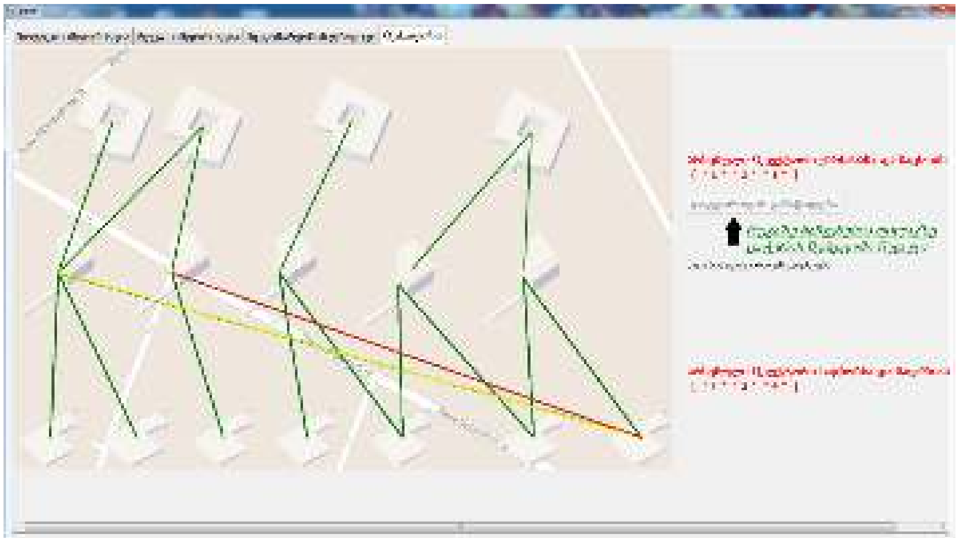


სურათი 3:

ვაწვებით ღილაკს „დატვირთვის განაწილება“

სურათი 4:

ღილაკზე დაწოლის შემდეგ ამჯერად მაღაზია ვეღარ გაუძლებს ძლიერ დატვირთვას ამიტომ საჭიროა კიდევ ერთხელ გადავანაწილოთ დატვირთვა.



სურათი 5:

საბოლოო დატვირთვის გადანაწილების შემდეგ პროგრამა გვიჩვენებს რომ ეს მაღაზია დატვირტვას გაუძლებს.

საქალაქო სისტემის – შენობების, ქუჩების, სკვერების და ა.შ. მოდელირებისა და მათემატიკური აღწერისათვის გამოყენებული იქნა TransNet პროგრამული სისტემა, რომლის დანიშნულებაა სატრანსპორტო ნაკადების გათვლა. აღნიშნული პროგრამული სისტემა მოიცავს მარტივ გრაფიკულ რედაქტორს.

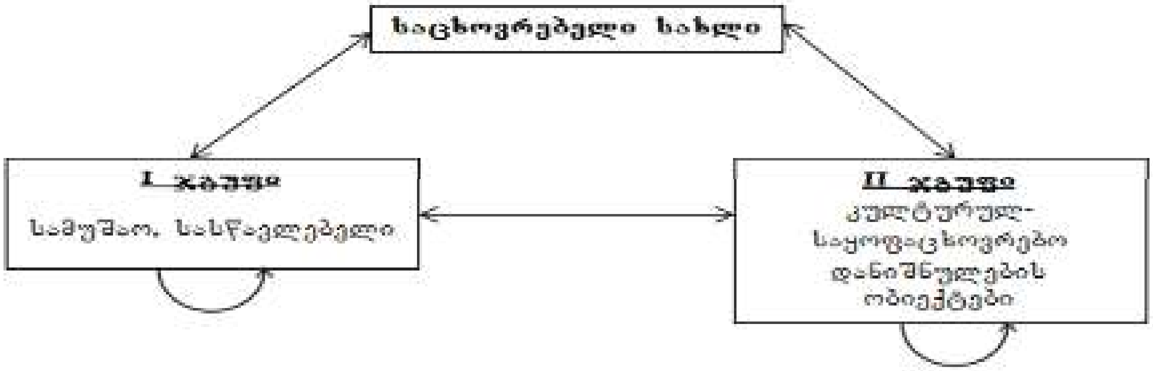
გრაფიკული რედაქტორი საშუალებას გვაძლევს შევიყვანოთ სატრანსპორტო ქსელის გრაფიკი. მოსახლეობის საცხოვრებელი, საყოფაცხოვრებო დანიშნულების და სხვა ობიექტების ადგილმდებარეობები და ურთიერთმიმართებანი.

<u>სამოქალაქო და სატრანსპორტო ნაკადების პროგნოზირების მოდელი</u>

მოდელი საშუალებას გვაძლევს პროგნოზირება გავუკეთოთ სამოქალაქო და საავტომობილო ნაკადებს, დღის სხვადასხვა დროისათვის.

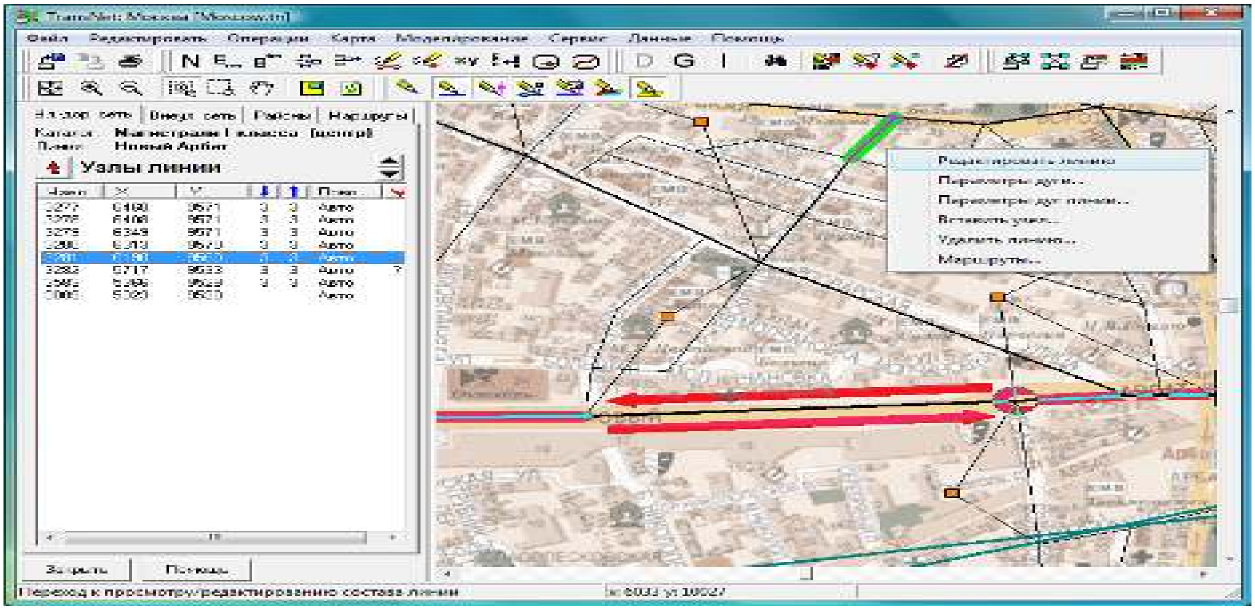
მოდელის მეთოდოლოგიური საფუძვლები

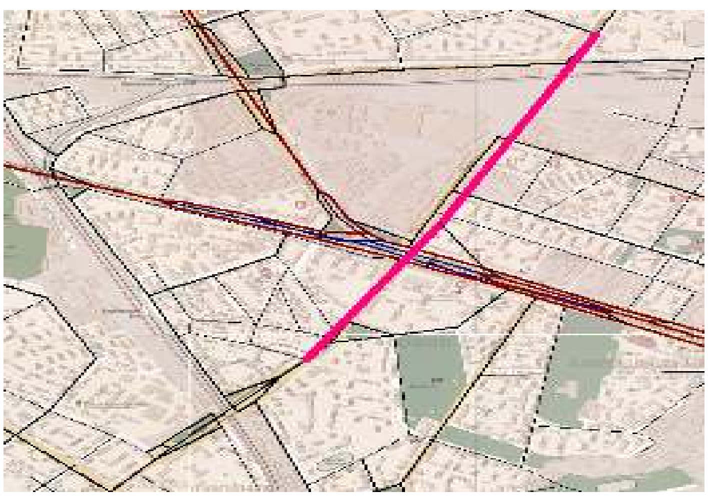
- <u>მოსახლეობის გადაადგილება</u> – გადაადგილებათა საშუალო რაოდენობა 24 საათის განმავლობაში.
- <u>გადაადგილების დანიშნულება</u> – გადაადგილება, განპირობებული სხვადასხვა მიზნებით.
- <u>გადაადგილების საშუალება</u> – ავტომობილი, საზოგადოებრივი ტრანსპორტი.

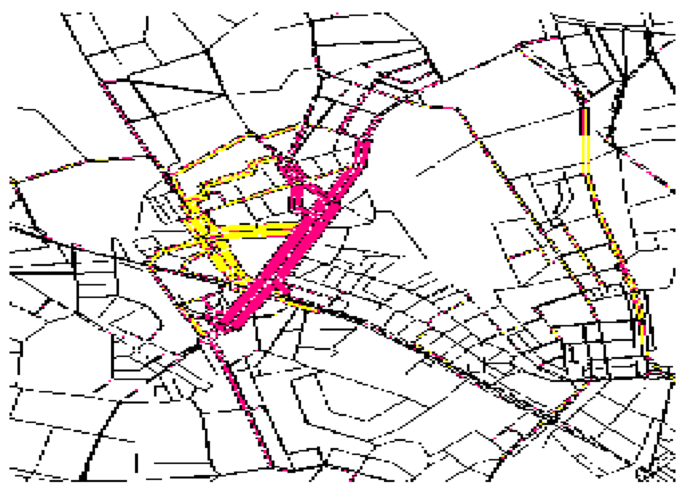


<u>რა ამოცანების გადაწყვეტაა შესაძლებელი აღნიშნული მოდელით</u>

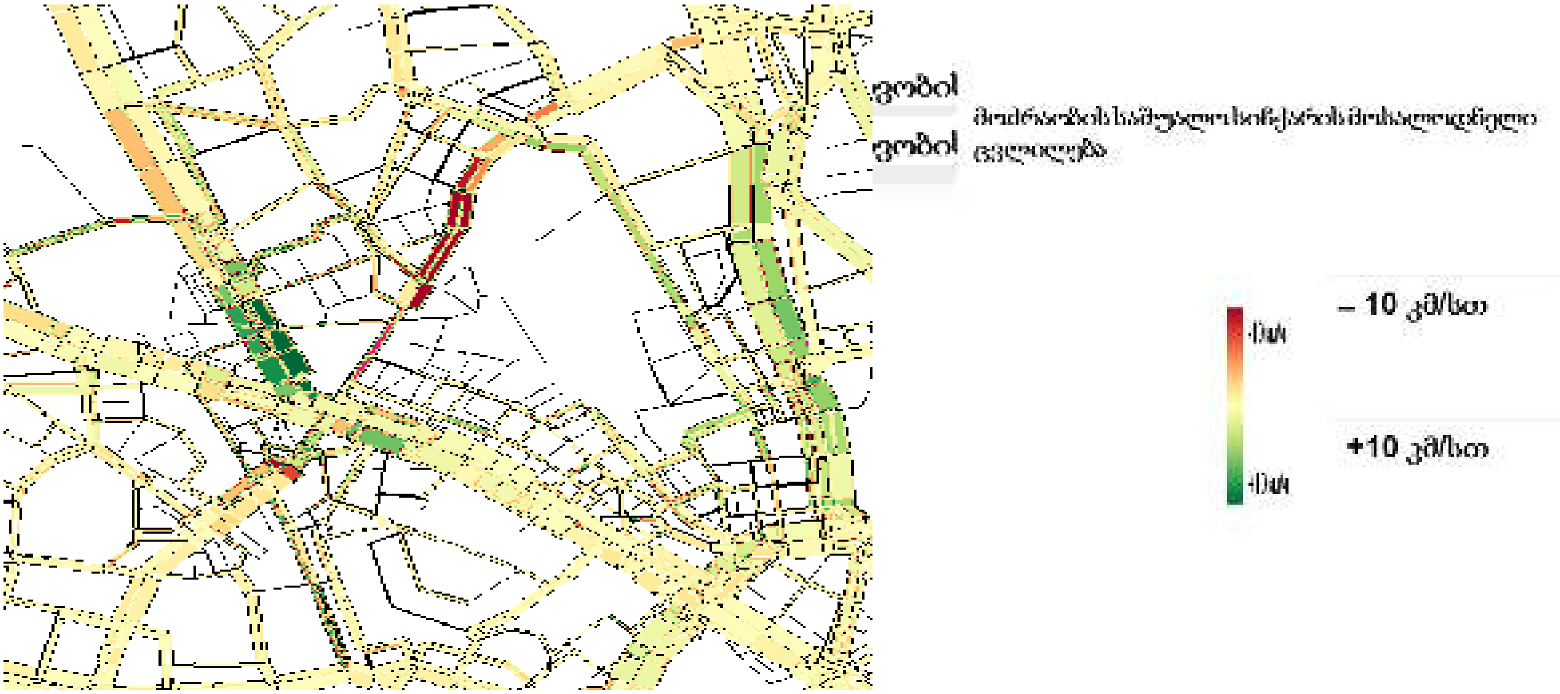
- საზოგადოებრივი ტრანსპორტის სამარშრუტო სისტემის ოპტიმიზაცია;
- საპროექტო საგზაო მარშრუტების დატვირთულობის პროგნოზი;
- ავტოსადგურების, ავტოსადგომების ოპტიმალური განაწილება;
- სატრანსპორტო ქსელის ზოგადი პარამეტრების შეფასება: საშუალო სიჩქარე, მონაკვეთის გარბენის დრო;
- ამა თუ იმ მარშრუტზე.
- ამა თუ იმ მარშრუტზე ზოგიერთი სატრანსპორტო საშუალებებზე შეზღუდვების დაწესება.

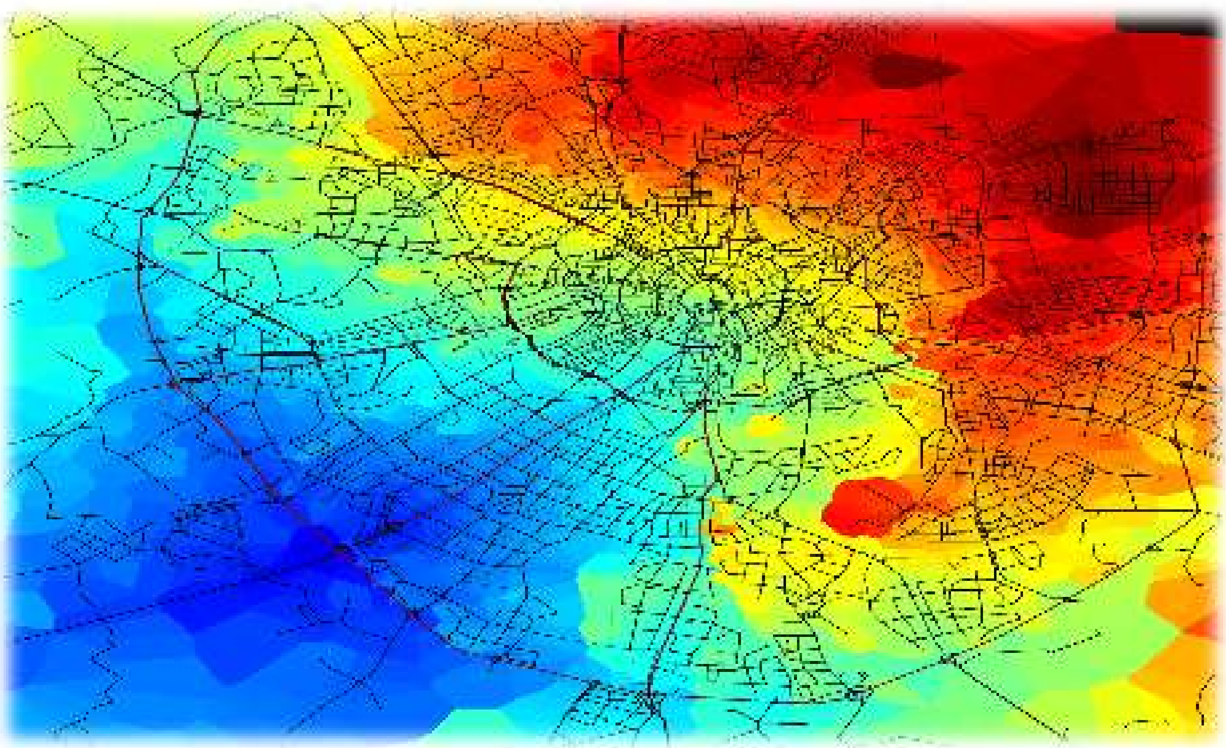




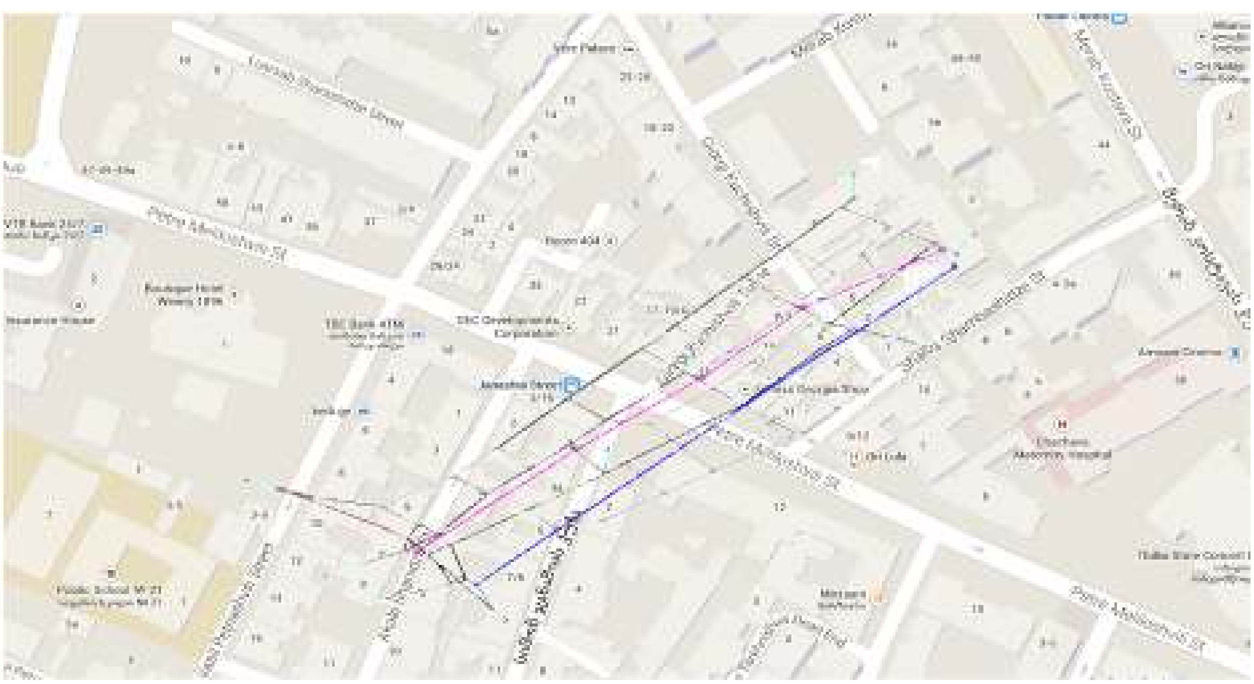


ავტომობილთა ნაკადის ინტენსიობის მოსალოდნელი ცვლილება, ახალი გზის გაყვანის შემდეგ


– 10 კმ/სთ
+10 კმ/სთ

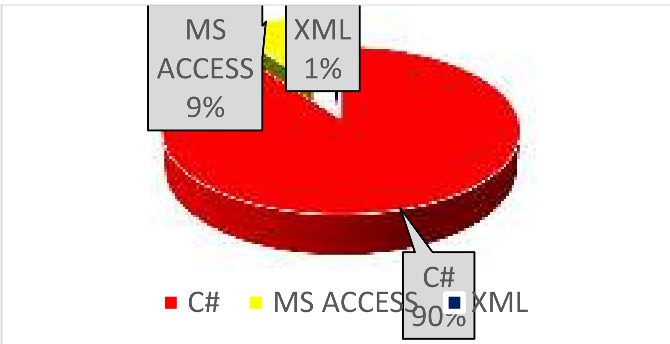


მოძრაობის ინტენსიობის გრაფიკული ასახვა



- <u>გამოყენებული ტექნოლოგიები:</u>

პროგრამა რეალიზებულია C#-ზე. მონაცემთა ბაზისათვის აღებულია MS ACCESS-ის მონაცემთა ბაზა, ასევე გამოყენებულია XML (სტატიკური ინფორმაციისათვის).



მინიმალური აპარატურული მოთხოვნები:

* პროცესორი: 1.6GHz (ან უკეთესი)

* ოპერატიული მეხსიერება: 2 GB

*თავისუფალი ადგილი მყარ დისკზე: 3GB

*მყარი დისკის ბრუნვის სიჩქარე: 7400 RPM

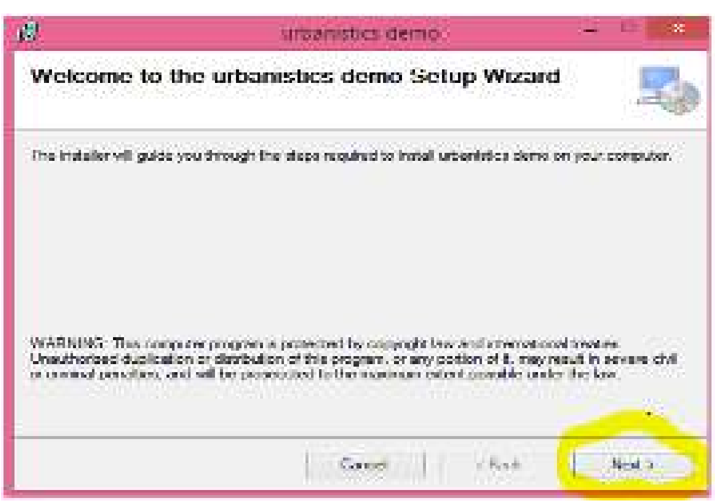
* მხარდაჭერა: DirectX 9

* ვიდეო კარტა: 512MB (1024 x 768 ან უკეთესი გაფართოების ეკრანი)

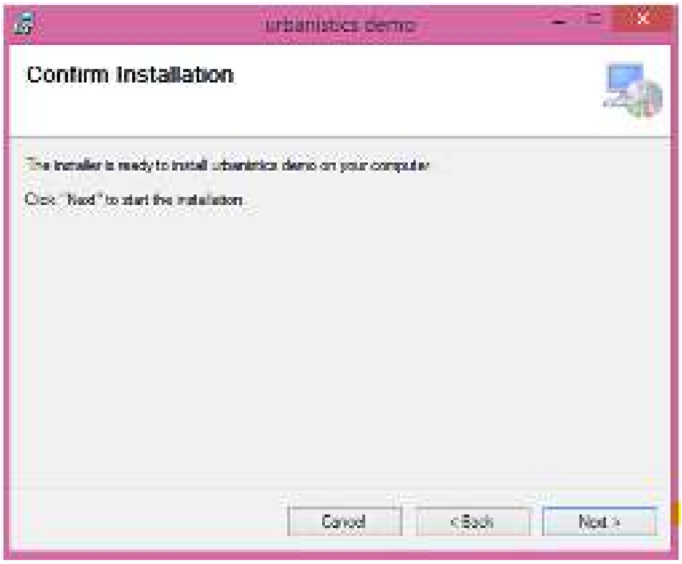
- <u>პროგრამის ინსტალაცია:</u>

- პროგრამის დასაინსტალირებლად ვრთავ საინსტალაციო ფაილს 

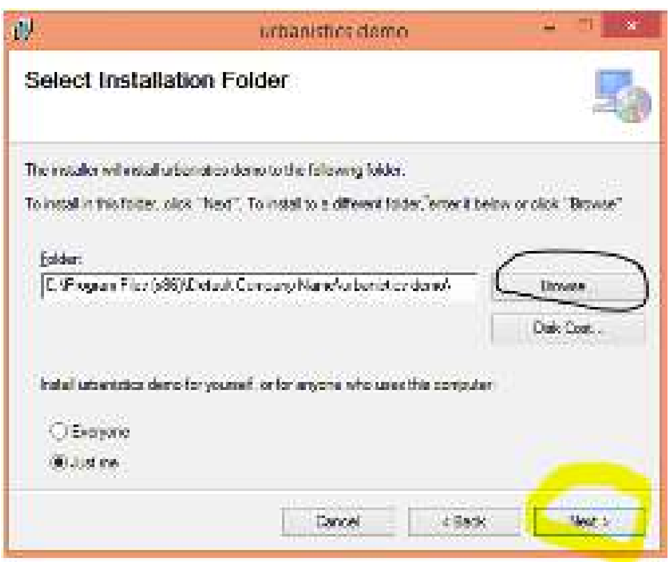
➔



რის შემდეგაც გამოდის შემდეგი ფანჯარა➔

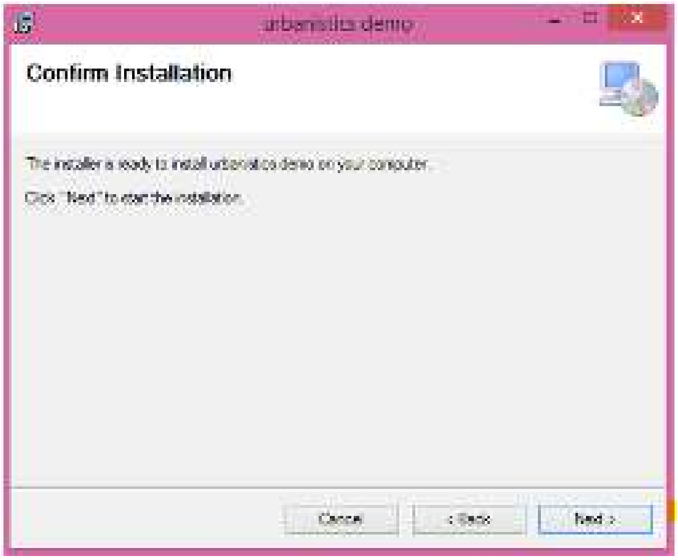


ამ ფანჯარის გამოსვლისას ვაწვებით Next>ღილაკს რომელსაც გადავყავართ შემდეგ განყოფილებაში ➔



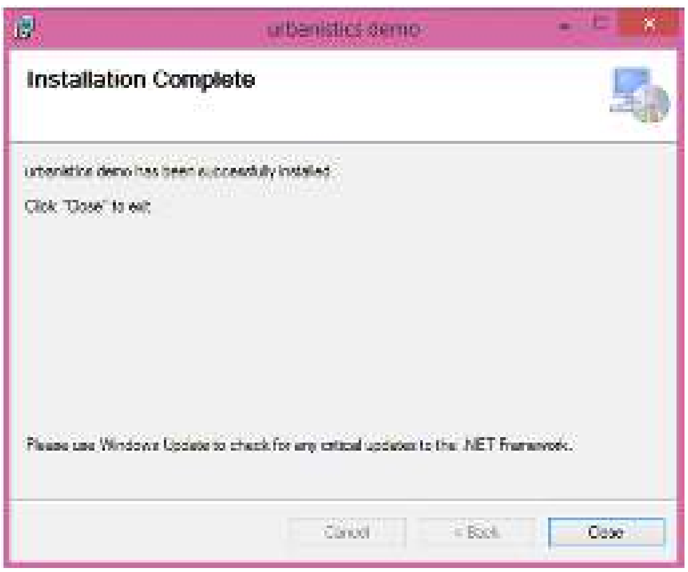
ამ ფანჯარაში Browse… ღილაკის დაჭერის შემთხვევაში საშუალება გვექნება ავირჩიოთ ჩვენთვის სასურველი საქაღალდე რომელშიც დაინსტალირდება ეს პროგრამა და შემდეგ დავაჭიროთ Next>ღილაკს, ან შეგვიძლია პირდაპირ დავაჭიროთ Next> ღილაკს და პროგრამა დაინსტალირდება ავტომატურად შემოთავაზებულ ფოლდერში.

შემდეგ გამოვა შემდეგი ფანჟარა ➔



ეს ფანჯარა გვატყობინებს რომ პროგრამა დასაინსტალირებლად მზადაა. ინსტალაციის გასაგრძელებლად ვაწვევით Next>ღილაკს

შემდეგ გამოდის შემდეგი ფანჯარა➔



ეს ფანჯარა აღნიშნავს რომ ინსტალაცია წარმატებით დასრულდა. ვაწვებით Close ღილაკს ინსტალაციის დასასრულებლად.

ამის შემდეგ სამუშაო მაგიდაზე გამოდის ამ პროგრამის აღმნიშვნელი სიმბოლო.

პროგრამის ჩასართავად ვაწვებით ამ სიმბოლოს 🡐რის შემდეგადაც ირთვება პროგრამა.

დანართი

ქალაქი წარმოადგენს სპეციფიურ სივრცულ გარემოს, რომლის ფორმირება ხორციელდება საზოგადოების განვითარების პროცესთან ერთად. ქალაქი წარმოადგენს საზოგადოებრივი განვითარების ყველა სფეროს მატერიალურ ასახვას.

ქალაქი წარმოადგენს კაცობრიობის არა მარტო მატერიალურ - სულიერ კულტურის ძეგლს, არამედ ის გვეხმარება ობიექტურად შევაფასოთ და გავიაზროთ ჩვენი თანამედროვეობა, განვსაზღვროთ კაცობრიობის განვითარების ახალი ტენდენციები და საზოგადოებრივი განვითარების სამომავლო ფორმები.

ქალაქური სისტემის ფორმირების პროცესი ხორციელდება ხანგრძლივი დროის განმავლობაში, რომელიც მოიცავს და აერთიანებს: პროექტებს, იდეებს, სტიქიურ მოვლენებს, შემთხვევით პროცესებს და ა.შ.

ქალაქი წარმოადგენს ცოცხალ ორგანიზმს, რომელიც როგორც მოიხმარს მატერიალურ, ფინანსურ, ინტელექტუალურ და სხვა რესურსებს, ასევე „იცილებს" მის მიერვე გადამუშავებულ მატერიალურ რესურსებს, გამოყენებულ პროდუქტებს (საწარმოო ნარჩენები, გადამუშავებული წყალი და სხვა) .

ქალაქი წარმოადგენს ცივილიზაციის უმაღლეს გამოვლინებას, რამეთუ მოიცავს ყოველივე იმას, რაც შეუქმნია კაცობრიობას თავისი არსებობის პერიოდში, მოიცავს იმ უნიკალურ გამოცდილებას, რაც გროვდება დასაბამიდან.

თანამედროვე მეცნიერული მიმართულების ქაოსის თეორიის თანახმად, ჩვენს მიერ დაშვებულ პატარა შეცდომას, შეიძლება შემდეგში მოყვეს გამოუსწორებელი შედეგები, რაც დაუშვებელია. აქედან გამომდინარე, მდგრადი განვითარების პრინციპი გვავალდებულებს შევქმნათ ისეთი ექსპერტული სისტემები, რომლებიც საშუალებას გვაძლევს შევაფასოთ ყოველი ჩვენი გადაწყვეტილება, რათა დაშვებული არ იქნას ისეთი შეცდომები, რომელიც შემდგომ კატასტროფამდე მიგვიყვანს.

საქართველოს ტექნიკურ უნივერსიტეტში მრავალი წლების განმავლობაში მეცნიერთა ჯგუფი (მათემატიკოსების, ინფორმატიკოსების და არქიტექტორების) მუშაობდა აღნიშნული მიმართულებით: შექმნილია ქალაქის (ურბანული სისტემის) იმიტაციური მოდელირების მათემატიკური მოდელი, რომელიც საშუალებას იძლევა შევაფასოთ ყველა ის ქმედება, რომელიც განხორციელდება ქალაქის

რეკონსტრუქციისას, აღმშენებლობისას. საშუალებას იძლევა შევაფასოთ ყველა რეკონსტრუქციისა თუ ახალი განაშენიანების პროექტი, ექსპერტიზა ჩაუტარდეს ქალაქის გენგეგმას. მათემატიკური მოდელი საშუალებას გვაძლევს განვსაზღვროთ პროექტის განხორციელების გზა და ეტაპები, რათა მივიღოთ ჩვენთვის სასურველი შედეგი სოციალურ ჭრილში.

ყველაზე საინტერესო, დამახასიათებელი, სპეციფიური პროცესები მიმდინარეობს სხვადასხვა მოვლენების ურთიერთქმედებათა დროს. რომლის დროსაც შეიძლება ქალაქმა შეიძინოს თვისობრივად სრულიად ახალი თვისებები (შეიძინოს ისეთი თვისება, რომელიც არც ერთ დაკვირვებად მოვლენაში არ შეიმჩნევა).

ქალაქური სისტემის მოდელირებისას, მისი სტრუქტურული და სისტემური ანალიზის დროს საქმე გვაქვს, ერთდროულად მიმდინარე პროცესებთან, როგორც სტოქასტიკურ (ქალაქის შემადგენელ თითოეული ელემენტისათვის დამახასიათებელი) ასევე აგრეგირებული დეტერმინირებულ (რომელიც შედეგია სტოქასტიკური ქმედებების ერთობლიობისა) პროცესებთან.

შეიძლება ითქვას, რომ ქალაქი მიეკუთვნება სტოქასტიკურ-დეტერმინირებულ სისტემათა კლასს და აქედან გამომდინარე მისი შესწავლა აუცილებლად უნდა მოხდეს შესაბამისი მათემატიკური აპარატების გამოყენებით.

ქალაქის განვითარების კონცეფცია ცხადია, ერთდროულად უნდა მოიცავდეს როგორც კონსერვატიულ, ასევე ახალ, დინამიკურ ფაქტორებს.

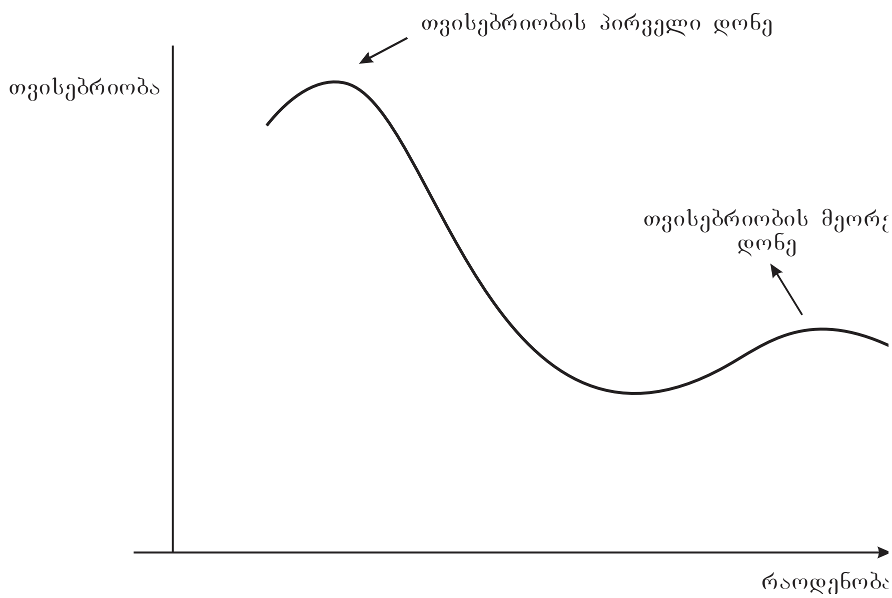
ქალაქის ფუნქციონალურ-სივრცული მოდელი დინამიკური სისტემური მოდელია. მასში ასახულია ქალაქში მიმდინარე სოციალური, ეკონომიკური, ეკოლოგიური, დემოგრაფიული და სხვა ფაქტორები.

ქალაქის, ურბანული სისტემის დინამიკური განვითარების კომპლექსური და ობიექტური შეფასებისათვის, აუცილებელია ევოლუციური განვიტარების და სისტემური მეთოდების გამოყენება.

ჩვენს ნაშრომში აღვნიშნეთ, რომ ქალაქის, ურბანული სისტემის განვითარების ადეკვატური მათემატიკური მოდელები მიეკუთვნებიან მაკროსისტემურ, არაწრფივ სისტემათა კლასს.

სოციალური სისტემების სხვადასხვა ნაწილების ურთიერთქმედებათა კანონების არაწრფივობა წარმოადგენს თვითორგანიზაციის ერთ-ერთ უმნიშვნელოვანეს პირობას დროსა და სივრცეში.

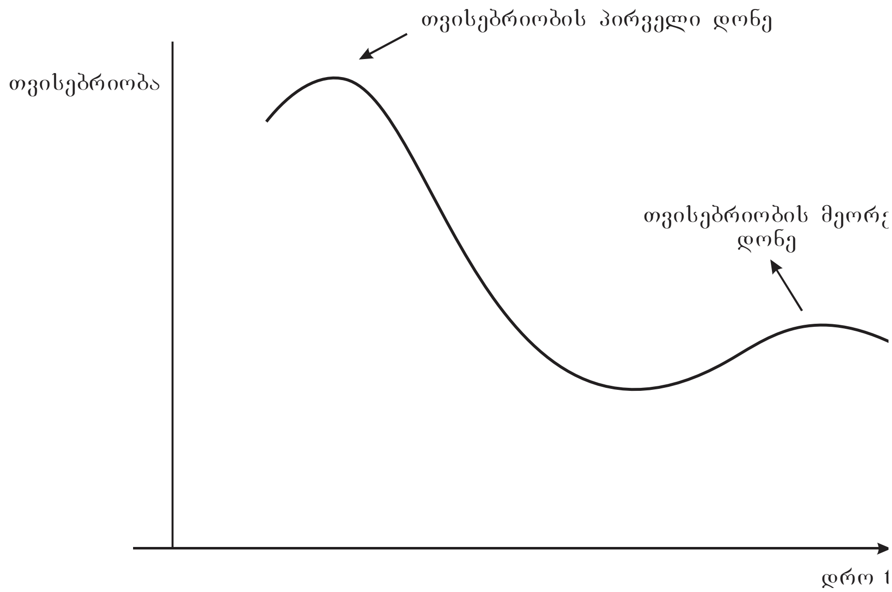
ურთიერთგადასვლის პროცესი ეფუძნება დიალექტიკის კანონს – რაოდენობრივი ცვლილების გადასვლისას თვისობრივში. ცნობილია, რომ თვისობრივი მახასიათებლის ერთი დონიდან მეორეზე გადასვლა ხდება



არაწრფივი კანონის შესაბამისად (ნახ. 1).

ნახ. 1

ანალოგიური კანონზომიერებაა დროში (ნახ. 2).



ნახ. 2

მოყვანილი

კანონზომიერებანი მნიშნველოვანია ქალაქური, ურბანული სისტემების განვითარების შესწავლისას.

ცნობილია გარკვეული კანონები, რომლის მიხედვითაც ყალიბდება სხვადასხვა სიმძლავრის ქალაქების ქსელი (გარკვეული სიდიდის კონცენტრაციები), რომლებიც დაკავშირებულია ურბანიზაციის პროცესის ევოლუციური განვითარების ეტაპებთან, სადაც აღინიშნება მოსახლეობის რაოდენობის (ასევე სხვამახასიათებლების) სიხევრისათვის ცვლილების ტალღისებრი ხასიათი.

“თვისობრიობის” - K_a და “რაოდენობა” (K_0) – დამოკიდებულებას აქვს შემდეგი სახე.

ლოჯისტიკური ფუნქციის სახე:

$$K_a = \frac{k}{(1 + be^{-CK_0})} \tag{1}$$

სადაცk - არის გაჯერადობის სიდიდე და K_a-ს საწყისი მნიშვნელობაა

$$K_a = \frac{k}{(1+b)}$$

როცაK_0=0.

C – კონსტანტაა.

“რაოდენობის” (K_0) ცვლილება დროში შეიძლება აღვწეროთ გამოსახულებით [1]:

$$K_0=R-ae^{nt} \qquad (2)$$

სადაცR – გაჯერების სიდიდეა. საწყისი პირობები $K_0=R-a$. n – მუდმივი სიდიდეა.

(1) და (2) ფროემულების გაერთიანებით მივიღებთ:

$$K_a = \frac{ke^R}{e^{ae^{nt}}(e^{CK_0}+b)} \qquad (3)$$

გამოსახულება საშუალებას გვაძლევს კომპლექსურად და ობიექტურად შევისწავლოთ ქალაქის განვითარების დინამიკა და ობიექტური ინტერპრეტაცია გავუკეთოთ მიღებულ შედეგებს.

თავდაპირველად ლოჯისტიკურ განტოლებას ჰქონდა შემდეგი სახე:

$$x(t) = \frac{k}{1+e^{a-b}}$$

a, b – წარმოადგენს ათვლის საწყისს და დროითი ცვლადების მასშტაბს.

ზრდის სიჩქარე მაქსიმუმი, როცა $t=a/b$. რის შემდეგ ეცემა ნულზე. k – არისასიმპტოტური მნიშვნელობა. ეს შეიძლება ჩაიწეროს

$$\dot{x} = rx(k-x)/k$$

როცა$x \ll k$ – გვაქვს ექსპონენციალურიზრდა x-ის. მაგრამ როცა $x \to k$, მაშინ ზრდის სიჩქარე ეცემა ნულამდე.

…

პერუს დედაქალაქ ლიმაში მილიონზე მეტ მოსახლეობას არა აქვს სასმელი წყალი. სან-პაულოში ნახევარი ქალაქის წყლები ჩაედინება, ყოველგვარი ფილტრის გარეშე მდინარეებში. ჰონგკონგში რეგულარულად სმოგია. ნიუ-დელიში ელექტროობის პრობლემა აქვთ. ნეაპოლი იხრჩობა ნაგავში. ლონდონში

და მოსკოვში "ათ ბალიანი" საცობებია, რომლებიც ავტომობილებს მეტალურ კუებად აქცევენ.

მაგრამ ყველგან ხალხი გარბის ქალაქისკენ (ცხადია რატომ?)

ხშირად მსხვილი ქალაქები წარმოადგენენ ეკონომიკის (მთავარ) მამოძრავებელს ქვეყნისა.

მეხიკო, სან-პაულო, ბანგკოკი, სადაც შიდა პროდუქტი (ნაციონალური) 30-50% იწარმოება.

უკვე ახლა 51% მსოფლიო მოსახლეობისა ცხოვრობს ქალაქებში. 2050 წლისათვის იცხოვრებს 2/3, 6.5 მილიარდი.

მაგრამ ამგვარი სტიქიური-ურბანის ძვირად გვიჯდება.

მეგაპოლისები უდიდეს ენერგიას და ნედლეულს მოიხმარენ. ქალაქებს უჭირავთ 3% პლანეტის ზედაპირის.

მაგრამ "სიმენსის" (ენცერნის შეკვეთით) გამომუშავებული ენერგიის 2/3 და სასმელი წყლის 60% მოიხმარენ. ქალაქები კი წარმოიშობენ დიდი რაოდენობით ნაგავს, მომწამვლელ ნიVთიერებებს, მხუთავ აირს, ფეკალურ წყლებს და ა.შ. ქალაქების წილი ნახშირორჟანგის აირის (углекислогогаза) მსოფლიო ემისიაში შეადგენს დაახლოებით 80%.ქალაქების ნორმალური კვებისათვის, სასმელი წყლიათვის კი საჭიროა უდიდესი ტერიტორიები.

ასე მაგალითად, ლონდონის მოსახლეობის გამოკვებისათვის სასოფლო პროდუქტებით, საჭიროა 125-ჯერ უფრო მეტი ტერიტორია, ვიდრე ლონდონს უკავია.

(ჟურნალი New Scientist ეკოლოგიური მიმომხილველი ფრედ პირსი).

ცხადია ამდაგვარი გამვითარების ტენდენციები შეგვიყვანენ ჩიხში.

როგორ უნდა მოვაგვაროთ ყოველივე ეს ზემოთ მოყვანილი პრობლემები და როგორ ავიცილოთ ის პრობლემები, რომლებიც სამომავლოდ გველოდება?

ერთი სიტყვით, ქალაქებს არ უნდა მივცეთ იმის უფლება, რომ დეფორმირება მოახდინოს ჩვენზე, ადამიანებზე. ქალაქი უნდა გვემსახურებოდეს ჩვენ და არა პირიქით.

თუ ასე გაგრძელდა, ეკოლოგიურ კრიზისს ვერ გადავურჩებით.

ყოველივე ამის აცილება შესაძლებელია მხოლოდ მეცნიერების, გამომგონებლების, ინჟინრების – ყველა იმათი ვინც ქმნის ახალს, ერთობლივი ქმედებები.

მეცნიერების დევნა ქვეყანაში იწვევს ქვეყნის დეგრადაციას. ღარიბი მეცნიერება – ცხოვრებას რახიტს.

ყოველივე ეს ერთმანეთთანაა გადაბმული, ამიტომ

ახლა ყოველი საცხოვრებელი რაიონი უნდა იყოს შეკაზმული ისე, როგორც ნამცხვარი – ოფისებით, მცირე საწარმოებით, და ყველა იმეებით, რომლებიც ქალაქურ გარემოს ზიანს არ აყენებს.

ქალაქი არ უნდა გავიგოთ, როგორც პოლიტიკოსების, მთავრობის იარაღი, როგორც "ხაზეინების" სუბიექტური ინსტრუმენტი.

ამისათვის სწაჭიროა გვქონდეს ისეთი ინსტრუმენტი, რომელიც გააკონტროლებს ხელისუფალთა გადაწყვეტილებას.

არ შეიძლება ყველაფრის კონტროლირება. ქალაქი რადგანაც არის ცოცხალი ორგანიზმი, ამიტომ უნდა გავითვალისწინოთ მისი "აზრიც". როგორ და რანაირად განვითარდეს!

საჭიროა როგორც ტექნიკური, ასევე სტრუქტურული ნოვაციები. აუცილებელია ჩავერიოთ, გარდავქმნათ ქალაქის გეგმარების პრინციპები (Smaart City).

ევროპის სამოქალაქო ინიციატივა აგროვებს პეტიციაზე ხელისმომწერთ, რომ ქალაქებში ტრანსპორის მოძრაობა იყოს 30კმ/სთ. 2030 წლისთვის უფრო იყოს ელექტრომობილები.

ჩვენ ვცხოვრობთ "ქალაქში – მანქანებისათვის" და არა ადამიანებს.

ადრე ვფრობდით, ტრანსპორტისათვის შესაბამისი ზოლების გამოყოფას. ახლა პირიქით ფეხითმოსიარულეებისათვის.

(ზნ-ს 3.14)

ქალაქების ფუნქციონალურ-სივრცული სტრუქტურის რეალიზაცია მიმდინარეობს დროის ორ მასშტაბში – ოპერატიული (მართვა) და სტრატეგიული (რეგულირება).

დინამიკური სისტემური ანალიზის პროცედურის ძირითადი იდეა მდგომარეობს შემდგომში:

1. ქალაქისფუნქციონალურ-სივრცულიპროცესებისსისტემურიმოდელირება. იმიტაციური მოდელის შექმნა.

2. რეალურსაქალაქოსისტემაშიმიმდინარეპროცესებისკონტროლიდაანალიზი.

3. ქალაქისგანვითარებისსისტემურიპროექტის (გენგეგმის) შემუშავება.

4. კონკრეტულიგადაწყვეტილებისმიღებაგენგეგმისაღსრულებისათვისდა აღსრულების გეგმის შემუშავება, რაც გულისხმობს მართველი და რეგულირების ზემოქმედებების სინთეზს და მისი ეფექტურობის შეფასებას, ქალაქის იმიტაციური მოდელის საშუალებით.

ამ პროცედურის სტრუქტურა და ძირითადი ბლოკები ნაჩვენებია ნახ. 1.

პროცედურა ორ დონიანია.

მისი ბირთვია თავად ქალაქური სისტემა (ქს), რომელიც ხასიათდება გარკვეული, მხოლოდ მისთვის დამახასიათებელი პარამეტრების ჯგუფით. ამ პარამეტრების დროში ცვლილება – არის ქალაქის განვითარების პროგრამების რეალიაციის და (ქს) მოქმედი გარე ფაქტორები (გფ) ზემოქმედების შედეგი, რომელთა დიდი ნაწილის კონტროლირება შეუძლებელია.

ქალაქური სისტემის პარამეტრების მდგომარეობის ანალიზი და ცვლილება ხორციელდება რეალური ეფექტურობის შეფასების (რეშ) ბლოკში, სადაც ფორმირდება იმ კრიტერიუმების ერთობლიობა, რომლის ფარგლებში ფარგლებში ტარდება ქალაქის განვითარების პროგრამები.

ქალაქის განვითარების პროგრამები (ქგპ) ძირითადად წარმოადგენენ საპროექტო-სამშენებლო მიმართულებებს, რომლებიც რეალიზდება ... საპროექტო-სამშენებლო კომპლექსების საშუალებით (სსკ). ე. ი. შეგვიძლია ვთქვათ, რომ t (სსკ) ასრულებენ შემსრულებელი ორგანოს ფუნქციას, რომელიც პრაქტიკულად ახორციელებს ქალაქის განვითარების პროგრამებს.

დინამიკური სისტემური ანალიზის პროცედურის მეორე დონე დანიშნულებაა ქალაქის ფუნქციონალურ-სივრცითი განვითარების ოპერატიული და სტრატეგიული მართვა.

მის ძირითად ბირთვს წარმოადგენს ქალაქური სისტემის ფუნქციონალურ-სივრცითი მოდელი (ფსმ).

ეს მოდელი საშუალებას გვაძლევს იმიტაცია გავუკეთოთ ქალაქური სისტემის რეაქციას მისი განვითარების პროგრამაზე, რაც თავის მხრივ საშუალებას გვაძლევს პროგნოზირება გავუკეთოთ თუ რა შედეგს მოიტანს ამათუიმ პროექტის რეალიზაცია.

(ფსმ) წარმოადგენს დინამიკურ სისტემურ მოდელს იმ გაგებით, რომ იქ გათვალისწინებულია სოციალური, ეკონომიკური, ეკოლოგიური, დემოგრაფიული და ა.შ. სხვა ფაქტორები. ყველა ის ფაქტორი, რომელიც მოქმედებს ქალაქის ფუნქციონალურ-სივრცულ განვითარებაზე.

ამ დონეზე ერთ-ერთი უმნიშვნელოვანესია განვითარების პროგრამის ფორმირების ბლოკი (გპფ) და სისტემური რეგულირების ფორმირების (სრფ) ბლოკი.ქალაქის ფუნქციონალურ-სივრცული განვითარების პროგრამა – ესაა დაგეგმილი პარამეტრების ნატურალურ გამოსახულებაში და ის ფინანსური რესურსები, რაც სჭირდება მათ შესრულებას.

განვითარების პროგრამა შედგება მმართველი და რეგულირებადი ნაწილებისაგან. პირველი შეიცავს მართველ ზემოქმედებას. კერძოდ, იმ მატერიალური და ფინანსური ნაკადების განაწილებას, რომლებიც მიეცემა (სსკ) მათი რეალიზაციისათვის.

მეორე ნაწილი მოიცავს მარეგულირებელ ზემოქმედებას, რომელიც წარმოადგენს ქალაქის ფუნქციონირები წესების ერთობლიობას. მათი აგრეგირება შეიძლება, ძირითადად, ხუთ კლასად: ადმინისტრაციულ-ნორმატიული, მიწის (საკადასტრო), საბიუჯეტო-საგადასახადო, ორგანიზაციულ-სამართებლივი, ხელშეკრულებითი, ისტორიული ფასეულობების, სოციალური სტრუქტურის და სხვა მიხედვით.

აღნიშნული პროცედურის ერთ-ერთ უმნიშვნელოვანეს ბლოკს წარმოადგენს ფუნქციონალურ-სივრცითი მოდელის კალიბრირების ბლოკი (კბ).

(ფსმ) პარამეტრების პერიოდული კალიბრირება აუცილებელია იმიტომ, რომ ამათუიმ პროგრამის განხორციელების დროს ქალაქის სისტემამ შეიძლება მიიღოს სულ სხვა თვისებები, რომლის იმიტირება უკვე შეუძლებელია იმ არსებული პარამეტრებით, რომლებიც გათვალისწინებულია (ფსმ). ამისათვის საჭიროა მისი კალიბრირება

მოცემული ან სხვა განვითარების პროგრამის რეალიზაციის, ასევე, შიდა გაქტორების ზემოქმედების შედეგია, რომელთა უმეტესობა არ ემორჩილება რეალურ კონტროლს.

პროგრამის ტექსტი:

- პროგრამის ამონაბეჭდი

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Xml;
using System.Data.OleDb;
using Microsoft.VisualBasic.PowerPacks;
using System.Drawing.Imaging;
using System.Windows.Forms.DataVisualization.Charting;
namespace urbanistics
{
  public partial class Form1 : Form
  {
    int mouse_x, mouse_y;
    bool can_click = false;
    DataGridView datagrid2 = new DataGridView();
    DataGridView datagrid = new DataGridView();
    PictureBox[] buildings = new PictureBox[15];
    PictureBox[] stations = new PictureBox[2];
    PictureBox[] streets = new PictureBox[6];
    Button button1 = new Button();
    Button button2 = new Button();
    bool is_checked = false;
    int index = 0;
    TabControl control = new TabControl();
    TabPage regionis_ganxilva_tabpage = new TabPage();
```

```
    //
    TabPage obieqtebi_tabpage = new TabPage(); //
    MenuStrip menuStrip1 = new MenuStrip();          //ჩამოშლადი მენიუს
გამოცხადება
//ჩამოშლადი მენიუს წევრების გამოცხადება
    ToolStripMenuItem სToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem საგანმანათლებლოდაწესებულბებიToolStripMenuItem = new
ToolStripMenuItem();
    ToolStripMenuItem პარკებიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem ქუჩებიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem სამრეწვლოდაწესებულებებიToolStripMenuItem = new
ToolStripMenuItem();
    ToolStripMenuItem toolStripMenuItem1 = new ToolStripMenuItem();
    ToolStripMenuItem საყოფაცხოვრებოობიექტებიToolStripMenuItem = new
ToolStripMenuItem();
    ToolStripMenuItem კულტურულიდაწესებულებებიToolStripMenuItem = new
ToolStripMenuItem();
    ToolStripMenuItem გასართობიცენტრებიToolStripMenuItem = new
ToolStripMenuItem();
    ToolStripMenuItem საცხოვრებელიშენობებიToolStripMenuItem = new
ToolStripMenuItem();
    ToolStripMenuItem სასტუმროებიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem სკოლებიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem ბაღებიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem უნივერსიტეტიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem კორპუსიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem კერძოსახლიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem რაიონისმდგომარეობაToolStripMenuItem = new
ToolStripMenuItem();

    ToolStripMenuItem ბანკიToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem სავაადმყოფოToolStripMenuItem = new ToolStripMenuItem();
    ToolStripMenuItem მაღაზიაToolStripMenuItem = new ToolStripMenuItem();
```

```
        //

        public Form1()
        {
            InitializeComponent();

            try
            {
//მონაცემა ბაზის ასლის გაკეთება
                System.IO.File.Copy(Application.StartupPath + @"\datab.accdb",
Application.StartupPath + @"\cache.accdb", true);
            }
            catch (Exception)
            {
//განსაკუთრებული სიტუაცია იმ შემთხვევისთვის თუ მონაცემთა ბაზასთან ვერ
მოხდა დაკავშირება
                MessageBox.Show("მონაცემთა ბაზის ფაილი ვერ მოიძებნა");

            }

        }

        private void Form1_Load(object sender, EventArgs e) //პროგრამის ჩატვირთვისას
შესრულებული მოქმედებები
        {

            label1.Text = "ქალაქის რეგიონების რეკონსტრუქციის პროექტის შეფასების
ექსპერტული სისტემა\nდემო ვერსია ";
            label1.MaximumSize = new Size(this.Width - 30, this.Height - 60);
            label1.Location = new Point(this.Width / 2 - label1.Width / 2, this.Height / 2 -
label1.Height / 2);
```

```
            btn.Location = new Point(label1.Location.X + label1.Width / 2 - btn.Width / 2,
label1.Location.Y + label1.Height+20);
            control.Location = new Point(10, 10);
            control.Size = new Size(this.Width - 25, this.Height - 55);
            control.Font = new Font("sylfaen",14);

        }

        private void dawyeba_Click(object sender, EventArgs e) ღილაკზე "დაწყება"-ს შემდეგ
შესრულებელი მოქმედებები
        {
            this.Opacity = 0;
            this.Controls.Remove(label1);  // label1-ს წაშლა
            this.Controls.Remove(btn); // btn-ს წაშლა
            this.Controls.Add(control); //control-ს დამატება
            regionis_ganxilva(); // კლასის გამოცხადება
            begin(); // კლასის გამოცხადება
            obieqtebi_tabpage_call(); // კლასის გამოცხადება
            q_analizi(); // კლასის გამოცხადება
        }
        bool chart_set = false;
        public void obieqtebi_tabpage_call()
        {

            bool set = false;
            //tabpage (obieqtebi_tabpage)
            obieqtebi_tabpage.AutoScroll = true;
            control.Controls.Add(this.obieqtebi_tabpage);// ობიექტების დამატება
            chart_tab.Text = "მდგომარეობის გრაფიკი";
            chart_tab.AutoScroll = true;
            control.Controls.Add(chart_tab); // გრაფიკის დამატება
            obieqtebi_tabpage.Text = "ინფორმაცია";
            obieqtebi_tabpage.Controls.Add(menuStrip1); //მენიუს დამატება
            this.obieqtebi_tabpage.UseVisualStyleBackColor = true;
            //
```

```
        obieqtebi_tabpage.Enter += (s, e) => { button1.SendToBack(); button2.SendToBack();
listBox1.SendToBack(); };
        chart_tab.Enter += (s, e) => {  button1.SendToBack(); button2.SendToBack();
listBox1.SendToBack(); };

        ////

        DataGridView grid = new DataGridView();

        obieqtebi_tabpage.BackColor = Color.Silver;
        grid.BackgroundColor = obieqtebi_tabpage.BackColor;
        grid.AutoSize = true;
        grid.Font = new System.Drawing.Font("sylfaen", 12, FontStyle.Bold);
        grid.MaximumSize = new System.Drawing.Size(this.Width - 50, this.Height - 100);
        obieqtebi_tabpage.Controls.Add(grid);
        grid.ColumnHeadersHeight = 40;
        grid.BorderStyle = BorderStyle.None;
        grid.AllowUserToAddRows = false;
        grid.AllowUserToDeleteRows = false;
        grid.AllowUserToOrderColumns = false;
        grid.AllowUserToResizeColumns = false;
        grid.Visible = false;



        ////


        // menuStrip1-ზე წევრების მინიჭება და აღწერა
        //
        this.menuStrip1.Font = new System.Drawing.Font("Sylfaen", 12F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
        this.სToolStripMenuItem,
```

```
        this.toolStripMenuItem1,
        this.რაიონისმდგომარეობაToolStripMenuItem});
        this.menuStrip1.Location = new System.Drawing.Point(3, 3);
        this.menuStrip1.Name = "menuStrip1";
        this.menuStrip1.Size = new System.Drawing.Size(987, 30);
        this.menuStrip1.TabIndex = 0;
        this.menuStrip1.Text = "menuStrip1";
        //
        // სToolStripMenuItem-ზე წევრების მინიჭება აღწერა
        //
        this.სToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.საგანმანათლებლოდაწესებულბებიToolStripMenuItem,
        this.პარკებიToolStripMenuItem,
        this.ქუჩებიToolStripMenuItem,
        this.სამრეწველოდაწესებულებებიToolStripMenuItem,
        this.საყოფაცხოვრებოობიექტებიToolStripMenuItem,
        this.კულტურულიდაწესებულებებიToolStripMenuItem,
        this.გასართობიცენტრებიToolStripMenuItem,
        this.საცხოვრებელიშენობებიToolStripMenuItem,
        this.სასტუმროებიToolStripMenuItem,
        this.ბანკიToolStripMenuItem,
        this.საავადმყოფოToolStripMenuItem});
        this.სToolStripMenuItem.Name = "სToolStripMenuItem";
        this.სToolStripMenuItem.Size = new System.Drawing.Size(111, 26);
        this.სToolStripMenuItem.Text = "ობიექტები";
        სToolStripMenuItem.Click += (a, d) => { obieqtebi_tabpageareoba("hide"); };
        //
        // საგანმანათლებლოდაწესებულბებიToolStripMenuItem
        //

this.საგანმანათლებლოდაწესებულბებიToolStripMenuItem.DropDownItems.AddRange(
new System.Windows.Forms.ToolStripItem[] {
        this.სკოლებიToolStripMenuItem,
        this.ბაღებიToolStripMenuItem,
```

```
        this.უნივერსიტეტიToolStripMenuItem});
        this.საგანმანათლებლოდაწესებულებებიToolStripMenuItem.Name =
"საგანმანათლებლოდაწესებულებებიToolStripMenuItem";
        this.საგანმანათლებლოდაწესებულებებიToolStripMenuItem.Size = new
System.Drawing.Size(373, 26);
        this.საგანმანათლებლოდაწესებულებებიToolStripMenuItem.Text =
"საგანმანათლებლო დაწესებულებები";
        //
        // პარკებიToolStripMenuItem
        //
        this.პარკებიToolStripMenuItem.Name = "პარკებიToolStripMenuItem";
        this.პარკებიToolStripMenuItem.Size = new System.Drawing.Size(373, 26);
        this.პარკებიToolStripMenuItem.Text = "პარკები";
        //
        // ქუჩებიToolStripMenuItem
        //
        this.ქუჩებიToolStripMenuItem.Name = "ქუჩებიToolStripMenuItem";
        this.ქუჩებიToolStripMenuItem.Size = new System.Drawing.Size(373, 26);
        this.ქუჩებიToolStripMenuItem.Text = "ქუჩები";

        ქუჩებიToolStripMenuItem.Click += (s, k) =>   // ქუჩებზე დაჭერის შემდგომი
მოქმედებები
        {
          grid.Visible = true;

          DataSet dataSet = new DataSet();
          dataSet.ReadXml(Application.StartupPath + "\\quchebi.xml"); // მონაცემების
წაკითხვა
// ცხრილის დახაზვა >>
          grid.DataSource = dataSet.Tables[0];
          grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          int width = 0;
```

```
            for (int i = 0; i < grid.Columns.Count; i++)
            {
                width += grid.Columns[i].Width;
            }
            grid.Width = width;
            // ცხრილის მდებარეობის განსაზღვრა
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
        };

        //
        // სამრეწველოდაწესებულებებიToolStripMenuItem
        //

this.სამრეწველოდაწესებულებებიToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
        this.მაღაზიაToolStripMenuItem});
        this.სამრეწველოდაწესებულებებიToolStripMenuItem.Name =
"სამრეწველოდაწესებულებებიToolStripMenuItem";
        this.სამრეწველოდაწესებულებებიToolStripMenuItem.Size = new
System.Drawing.Size(373, 26);
        this.სამრეწველოდაწესებულებებიToolStripMenuItem.Text = "სამრეწველო
დაწესებულებები";
        //
        // toolStripMenuItem1
        //
        this.toolStripMenuItem1.Name = "toolStripMenuItem1";
        this.toolStripMenuItem1.Size = new System.Drawing.Size(12, 26);
        //

        // საყოფაცხოვრებოობიექტებიToolStripMenuItem
        //
        this.საყოფაცხოვრებოობიექტებიToolStripMenuItem.Name =
"საყოფაცხოვრებოობიექტებიToolStripMenuItem";
```

```
        this.საყოფაცხოვრებოობიექტებიToolStripMenuItem.Size = new
System.Drawing.Size(373, 26);
        this.საყოფაცხოვრებოობიექტებიToolStripMenuItem.Text = "საყოფაცხოვრებო
ობიექტები";
        //
        // კულტურულიდაწესებულებებიToolStripMenuItem
        //
        this.კულტურულიდაწესებულებებიToolStripMenuItem.Name =
"კულტურულიდაწესებულებებიToolStripMenuItem";
        this.კულტურულიდაწესებულებებიToolStripMenuItem.Size = new
System.Drawing.Size(373, 26);
        this.კულტურულიდაწესებულებებიToolStripMenuItem.Text = "კულტურული
დაწესებულებები";
        კულტურულიდაწესებულებებიToolStripMenuItem.Click += (s, k) =>
        {
            grid.Visible = true;

            DataSet dataSet = new DataSet();
            dataSet.ReadXml(Application.StartupPath + "\\kulturuli.xml");
            grid.DataSource = dataSet.Tables[0];
            grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
        };
        //
        // გასართობიცენტრებიToolStripMenuItem
```

```
//
this.გასართობიცენტრებიToolStripMenuItem.Name =
"გასართობიცენტრებიToolStripMenuItem";
this.გასართობიცენტრებიToolStripMenuItem.Size = new System.Drawing.Size(373,
26);
this.გასართობიცენტრებიToolStripMenuItem.Text = "გასართობი ცენტრები";
//
// საცხოვრებელიშენობებიToolStripMenuItem
//
this.საცხოვრებელიშენობებიToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
this.კორპუსიToolStripMenuItem,
this.კერძოსახლიToolStripMenuItem});
this.საცხოვრებელიშენობებიToolStripMenuItem.Name =
"საცხოვრებელიშენობებიToolStripMenuItem";
this.საცხოვრებელიშენობებიToolStripMenuItem.Size = new
System.Drawing.Size(373, 26);
this.საცხოვრებელიშენობებიToolStripMenuItem.Text = "საცხოვრებელი
შენობები";
//
// სასტუმროებიToolStripMenuItem
//
this.სასტუმროებიToolStripMenuItem.Name = "სასტუმროებიToolStripMenuItem";
this.სასტუმროებიToolStripMenuItem.Size = new System.Drawing.Size(373, 26);
this.სასტუმროებიToolStripMenuItem.Text = "სასტუმროები";
სასტუმროებიToolStripMenuItem.Click += (s, k) =>
{
   grid.Visible = true;

   DataSet dataSet = new DataSet();
   dataSet.ReadXml(Application.StartupPath + "\\sastumro.xml");
   grid.DataSource = dataSet.Tables[0];
   grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
   grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
   grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
```

```
            grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
        };
        //
        // სკოლებიToolStripMenuItem
        //
        this.სკოლებიToolStripMenuItem.Name = "სკოლებიToolStripMenuItem";
        this.სკოლებიToolStripMenuItem.Size = new System.Drawing.Size(206, 26);
        this.სკოლებიToolStripMenuItem.Text = "სკოლა";
        სკოლებიToolStripMenuItem.Click += (s, k) =>
        {
            grid.Visible = true;

            DataSet dataSet = new DataSet();
            dataSet.ReadXml(Application.StartupPath + "\\skola.xml");
            grid.DataSource = dataSet.Tables[0];
            grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
```

```
        };
        //
        // ბალებიToolStripMenuItem
        //
        this.ბალებიToolStripMenuItem.Name = "ბალებიToolStripMenuItem";
        this.ბალებიToolStripMenuItem.Size = new System.Drawing.Size(206, 26);
        this.ბალებიToolStripMenuItem.Text = "ბაღი";
        ბალებიToolStripMenuItem.Click += (s, k) =>
        {
            grid.Visible = true;

            DataSet dataSet = new DataSet();
            dataSet.ReadXml(Application.StartupPath + "\\sabavshvo_bagi.xml");
            grid.DataSource = dataSet.Tables[0];
            grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
        };

        //
        // უნივერსიტეტიToolStripMenuItem
        //
        this.უნივერსიტეტიToolStripMenuItem.Name =
"უნივერსიტეტიToolStripMenuItem";
        this.უნივერსიტეტიToolStripMenuItem.Size = new System.Drawing.Size(206, 26);
        this.უნივერსიტეტიToolStripMenuItem.Text = "უნივერსიტეტი";
```

```
        //
        // კორპუსიToolStripMenuItem
        //
        this.კორპუსიToolStripMenuItem.Name = "კორპუსიToolStripMenuItem";
        this.კორპუსიToolStripMenuItem.Size = new System.Drawing.Size(187, 26);
        this.კორპუსიToolStripMenuItem.Text = "კორპუსი";
        კორპუსიToolStripMenuItem.Click += (s, k) =>
        {
            grid.Visible = true;

            DataSet dataSet = new DataSet();
            dataSet.ReadXml(Application.StartupPath + "\\binebi.xml");
            grid.DataSource = dataSet.Tables[0];
            grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
        };
        //
        // კერძოსახლიToolStripMenuItem
        //
        this.კერძოსახლიToolStripMenuItem.Name = "კერძოსახლიToolStripMenuItem";
        this.კერძოსახლიToolStripMenuItem.Size = new System.Drawing.Size(187, 26);
        this.კერძოსახლიToolStripMenuItem.Text = "კერძო სახლი";
        //
        // რაიონისმდგომარეობაToolStripMenuItem
        //
```

```
        this.რაიონისმდგომარეობაToolStripMenuItem.Name =
"რაიონისმდგომარეობაToolStripItem";
        this.რაიონისმდგომარეობაToolStripMenuItem.Size = new
System.Drawing.Size(194, 26);
        this.რაიონისმდგომარეობაToolStripMenuItem.Text = "რაიონის მდგომარეობა";
        რაიონისმდგომარეობაToolStripMenuItem.Click += (s, k) => { control.SelectedIndex
= 2; obieqtebi_tabpageareoba("show"); grid.Visible = false; };
        //


        // ბანკიToolStripMenuItem
        //
        this.ბანკიToolStripMenuItem.Name = "ბანკიToolStripMenuItem";
        this.ბანკიToolStripMenuItem.Size = new System.Drawing.Size(194, 26);
        this.ბანკიToolStripMenuItem.Text = "ბანკი";
        ბანკიToolStripMenuItem.Click += (s, k) =>
        {
          grid.Visible = true;

          DataSet dataSet = new DataSet();
          dataSet.ReadXml(Application.StartupPath + "\\banki.xml");
          grid.DataSource = dataSet.Tables[0];
          grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
          grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
          dataSet.Dispose();
        };
```

```
            //
            // საავადმყოფოToolStripMenuItem
            //
            this.საავადმყოფოToolStripMenuItem.Name = "საავადმყოფოToolStripMenuItem";
            this.საავადმყოფოToolStripMenuItem.Size = new System.Drawing.Size(194, 26);
            this.საავადმყოფოToolStripMenuItem.Text = "საავადმყოფო";
            საავადმყოფოToolStripMenuItem.Click += (s, k) =>
            {
                grid.Visible = true;

                DataSet dataSet = new DataSet();
                dataSet.ReadXml(Application.StartupPath + "\\saavadmyofo.xml");
                grid.DataSource = dataSet.Tables[0];
                grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
                grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
    obieqtebi_tabpage.Height / 2 - grid.Height / 2);
                dataSet.Dispose();
            };
            //
            // მაღაზიაToolStripMenuItem
            //
            this.მაღაზიაToolStripMenuItem.Name = "მაღაზიაToolStripMenuItem";
            this.მაღაზიაToolStripMenuItem.Size = new System.Drawing.Size(194, 26);
            this.მაღაზიაToolStripMenuItem.Text = "მაღაზია";
            მაღაზიაToolStripMenuItem.Click += (s, k) =>
            {
                grid.Visible = true;
```

```
            DataSet dataSet = new DataSet();
            dataSet.ReadXml(Application.StartupPath + "\\magaziebi.xml");
            grid.DataSource = dataSet.Tables[0];
            grid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
            grid.Location = new Point(obieqtebi_tabpage.Width / 2 - grid.Width / 2,
obieqtebi_tabpage.Height / 2 - grid.Height / 2);
            dataSet.Dispose();
         };
         //

      }


      public void regionis_ganxilva()
      {

         Label gansaxilveli_regioni = new Label();
         Label raionebis_raodenoba = new Label();
         regionis_ganxilva_tabpage.Text = "ზოგადი ინფორმაცია";
         regionis_ganxilva_tabpage.AutoScroll = true;
         control.Controls.Add(regionis_ganxilva_tabpage);

         regionis_ganxilva_tabpage.Enter += (s, e) => { button1.SendToBack();
button2.SendToBack(); listBox1.SendToBack(); };

         gansaxilveli_regioni.ForeColor = Color.Green;
```

```
        gansaxilveli_regioni.Font = new System.Drawing.Font("sylfaen", 14, FontStyle.Bold);
        gansaxilveli_regioni.Text = "განსახილველი რეგიონი";
        gansaxilveli_regioni.AutoSize = true;
        gansaxilveli_regioni.Location = new Point(regionis_ganxilva_tabpage.Width / 2 -
gansaxilveli_regioni.Width / 2, 10);
        regionis_ganxilva_tabpage.Controls.Add(gansaxilveli_regioni);


        raionebis_raodenoba.ForeColor = Color.Green;
        raionebis_raodenoba.Font = new System.Drawing.Font("sylfaen", 14, FontStyle.Bold);
        raionebis_raodenoba.Text = "რაიონების რაოდენობა 2";
        raionebis_raodenoba.AutoSize = true;
        raionebis_raodenoba.Location = new Point(regionis_ganxilva_tabpage.Width / 2 -
raionebis_raodenoba.Width/2,gansaxilveli_regioni.Location.Y+gansaxilveli_regioni.Height+1
0);
        regionis_ganxilva_tabpage.Controls.Add(raionebis_raodenoba);



        datagrid.AutoSize = true;
        datagrid.Font = new System.Drawing.Font("sylfaen", 12, FontStyle.Bold);
        datagrid.MaximumSize = new System.Drawing.Size(this.Width-50,this.Height-100);
        DataSet dataSet = new DataSet();
        dataSet.ReadXml(Application.StartupPath+"\\data.xml");
        datagrid.DataSource = dataSet.Tables[0];
        regionis_ganxilva_tabpage.Controls.Add(datagrid);
        datagrid.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid.Columns[0].HeaderText = "პარამეტრის დასახელება";
        datagrid.Columns[1].HeaderText = "ზომის ერთეული";
        datagrid.Columns[2].HeaderText = "აღმნიშვნელი სიმბოლო";
        datagrid.Columns[3].HeaderText = "მნიშვნელობა I რაიონში";
```

```
        datagrid.Columns[4].HeaderText = "მნიშვნელობა II რაიონში";
        datagrid.ColumnHeadersHeight = 40;
        datagrid.AutoResizeRow(2, DataGridViewAutoSizeRowMode.AllCellsExceptHeader);
        datagrid.Location = new Point(regionis_ganxilva_tabpage.Width / 2 - datagrid.Width
/ 2, raionebis_raodenoba.Location.Y + raionebis_raodenoba.Height + 10);
        datagrid.BackgroundColor = regionis_ganxilva_tabpage.BackColor;
        datagrid.BorderStyle = BorderStyle.None;
        datagrid.AllowUserToAddRows = false;
        datagrid.AllowUserToDeleteRows = false;
        datagrid.AllowUserToOrderColumns = false;
        datagrid.AllowUserToResizeColumns = false;
        dataSet.Dispose();



       /* DataGridView datagrid1 = new DataGridView();
        datagrid1.AutoSize = true;
        datagrid1.Font = new System.Drawing.Font("sylfaen", 12, FontStyle.Bold);
        datagrid1.MaximumSize = new System.Drawing.Size(this.Width - 50, this.Height -
100);
       dataSet = new DataSet();
        dataSet.ReadXml(Application.StartupPath + "\\data1.xml");
       datagrid1.DataSource = dataSet.Tables[0];
       regionis_ganxilva_tabpage.Controls.Add(datagrid1);
        datagrid1.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid1.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid1.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
        datagrid1.Columns[0].HeaderText = "პარამეტრის დასახელება";
        datagrid1.Columns[1].HeaderText = "ზომის ერთეული";
        datagrid1.Columns[2].HeaderText = "აღმნიშვნელი სიმბოლო";
        datagrid1.ColumnHeadersHeight = 40;
       datagrid1.AutoResizeRow(2,
DataGridViewAutoSizeRowMode.AllCellsExceptHeader);
        datagrid1.Location = new Point(regionis_ganxilva_tabpage.Width / 2 -
datagrid1.Width / 2, datagrid.Location.Y + datagrid.Height + 10);
```

```
datagrid1.BackgroundColor = regionis_ganxilva_tabpage.BackColor;
 datagrid1.BorderStyle = BorderStyle.Fixed3D;
 datagrid1.AllowUserToAddRows = false;
 datagrid1.AllowUserToDeleteRows = false;
 datagrid1.AllowUserToOrderColumns = false;
 datagrid1.AllowUserToResizeColumns = false;
 dataSet.Dispose();*/




 datagrid2.AutoSize = true;
 datagrid2.Font = new System.Drawing.Font("sylfaen", 12, FontStyle.Bold);
 datagrid2.MaximumSize = new System.Drawing.Size(this.Width - 50, this.Height -
100);
 dataSet = new DataSet();
 dataSet.ReadXml(Application.StartupPath + "\\data2.xml");
 datagrid2.DataSource = dataSet.Tables[0];
 regionis_ganxilva_tabpage.Controls.Add(datagrid2);
 datagrid2.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
 datagrid2.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
 datagrid2.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
 datagrid2.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
 datagrid2.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
 datagrid2.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;
 datagrid2.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.AllCells;

 datagrid2.Columns[0].HeaderText = "ინტერესის დასახელება";
 datagrid2.Columns[1].HeaderText = "ზომის ერთეული";
 datagrid2.Columns[2].HeaderText = "წონა I რაიონში";
 datagrid2.Columns[3].HeaderText = "წონა II რაიონში";
 datagrid2.Columns[4].HeaderText = "სტანდარტი I რაიონში";
 datagrid2.Columns[5].HeaderText = "სტანდარტი II რაიონში";
 datagrid2.Columns[6].HeaderText = "რეალური I რაიონში";
 datagrid2.Columns[7].HeaderText = "რეალური IIრაიონში";
```

```
            datagrid2.ColumnHeadersHeight = 60;
            datagrid2.AutoResizeRow(2,
DataGridViewAutoSizeRowMode.AllCellsExceptHeader);
            datagrid2.Location = new Point(regionis_ganxilva_tabpage.Width / 2 -
datagrid2.Width / 2, datagrid.Location.Y + datagrid.Height + 10);
           datagrid2.BackgroundColor = regionis_ganxilva_tabpage.BackColor;
           datagrid2.BorderStyle = BorderStyle.None;
            datagrid2.AllowUserToAddRows = false;
            datagrid2.AllowUserToDeleteRows = false;
            datagrid2.AllowUserToOrderColumns = false;
            datagrid2.AllowUserToResizeColumns = false;
            dataSet.Dispose();




        }

        ListBox listBox1 = new ListBox();
        Panel panel1 = new Panel();
        TabPage tab1 = new TabPage();

        public void begin() // კლასი რომლის საშუალებითაც პროგრამის ჩართვისა და
ღილაკზე დაწყება დაჭერის შემდგომ იწყება რუკით აქ აღწერილია ამ
განყოფილებისთვის განკუთვნილი ობიექტები
        {

            tab1.AutoScroll = true;
            tab1.Text = "რუკა";
            panel1.Location = new Point(0, 0);
```

```
        tab1.Controls.Add(panel1); //panel1-ის დამატება.. რუკისთვის დამატებით ახალი
პანელია დამატებული
        panel1.AutoSize = true;
        control.Controls.Add(tab1); //უზრუნველყოფს ამ ახალი პანელის დამატებას
        listBox1.Location = new Point(40, 70);
        listBox1.Font = new System.Drawing.Font("sylfaen", 12, FontStyle.Bold);
        listBox1.Size = new System.Drawing.Size(this.Width / 4, this.Height / 5);
        panel1.BackgroundImage = global::urbanistics.Properties.Resources.rukaaaaaaaaaaa;
        this.Controls.Add(listBox1); // ლისტბოქსის დამატება რომელიც ობიექტის
დახასიათებისთვისაა განკუთვნილი
        button1.Location = new Point(listBox1.Location.X + listBox1.Width + 10, 100);
        button1.Font = new Font("sylfaen", 12, FontStyle.Bold);
        button1.Text = "წაშლა";
        button1.AutoSize = true;
        button1.Click += new EventHandler(button1_Click);
        this.Controls.Add(button1); // ღილაკის დამატება
        Visible = true;
        button1.BringToFront();
       button2.Text = "დამატება";
        button2.AutoSize = true;
        button2.Font = new System.Drawing.Font("sylfaen", 12, FontStyle.Bold);
        button2.Location = new Point(button1.Location.X + button1.Width + 10,
button1.Location.Y);
        this.Controls.Add(button2);
        button2.Click+=new EventHandler(button2_Click);
        button2.BringToFront();

//ეს კოდი უზრუნველყოფს წინ გადმოტანას ანუ ხილულს ხდის listBox1-ს,button1-ს
და button-ს
        tab1.Enter += (s, e) => { listBox1.BringToFront(); button1.BringToFront();
button2.BringToFront(); };


        control.SelectedIndex = 1;
```

```
        draw_map(); //კლასის გამოძახება რომელიც უზრუნველყოფს რუკაზე
ობიექტების დატანას
      }
      public void draw_map()
      {

        for (int i = 0; i < 15; i++)
        {

          buildings[i] = new PictureBox();
          buildings[i].BackColor = System.Drawing.Color.Transparent;
          buildings[i].TabStop = false;
          panel1.Controls.Add(this.buildings[i]);



        }
        this.buildings[0].Image = global::urbanistics.Properties.Resources._1;
        this.buildings[0].Location = new System.Drawing.Point(3, 3);
        this.buildings[0].Size = new System.Drawing.Size(144, 127);
        this.buildings[0].TabIndex = 0;
        this.buildings[0].Click += (s, e) =>
        {
          index = 0; is_checked = true; int width =
panel1.Size.Width;display_buildings_data(index); display_connect(index);
        };


        this.buildings[1].BackgroundImage =
global::urbanistics.Properties.Resources.pilarmonia2;
        this.buildings[1].Location = new System.Drawing.Point(726, 111);
        this.buildings[1].Size = new System.Drawing.Size(428, 439);
        this.buildings[1].TabIndex = 13;
        this.buildings[1].Click += (s, e) =>
        {
```

```
            index = 1; is_checked = true; display_buildings_data(index);
display_connect(index);
         };
         this.buildings[1].BringToFront();
         this.buildings[2].Image = global::urbanistics.Properties.Resources._4;
         this.buildings[2].Location = new System.Drawing.Point(154, 23);
         this.buildings[2].Size = new System.Drawing.Size(182, 158);
         this.buildings[2].TabIndex = 2;
         this.buildings[2].Click += (s, e) =>
         {
            index = 2; is_checked = true; display_buildings_data(index);
display_connect(index);
         };

         this.buildings[3].BackgroundImage = global::urbanistics.Properties.Resources._14;
         this.buildings[3].Location = new System.Drawing.Point(1282, 341);
         this.buildings[3].Size = new System.Drawing.Size(128, 118);
         this.buildings[3].TabIndex = 14;
         this.buildings[3].Click += (s, e) =>
         {
            index = 3; is_checked = true; display_buildings_data(index);
display_connect(index);
         };

         this.buildings[4].BackgroundImage = global::urbanistics.Properties.Resources._54;
         this.buildings[4].Location = new System.Drawing.Point(921, 821);
         this.buildings[4].Size = new System.Drawing.Size(319, 203);
         this.buildings[4].TabIndex = 15;
         this.buildings[4].Click += (s, e) =>
         {
            index = 4; is_checked = true; display_buildings_data(index);
display_connect(index);
         };

         this.buildings[5].BackgroundImage = global::urbanistics.Properties.Resources._30;
```

```
        this.buildings[5].Location = new System.Drawing.Point(713, 572);
        this.buildings[5].Size = new System.Drawing.Size(239, 194);
        this.buildings[5].TabIndex = 16;
        this.buildings[5].Click += (s, e) =>
        {
            index = 5; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[6].BackgroundImage = global::urbanistics.Properties.Resources._9;
        this.buildings[6].BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
        this.buildings[6].Location = new System.Drawing.Point(526, -25);
        this.buildings[6].Size = new System.Drawing.Size(426, 244);
        this.buildings[6].TabIndex = 6;
        this.buildings[6].Click += (s, e) =>
        {
            index = 6; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[7].BackgroundImage = global::urbanistics.Properties.Resources._10;
        this.buildings[7].BackgroundImageLayout =
System.Windows.Forms.ImageLayout.Stretch;
        this.buildings[7].Location = new System.Drawing.Point(320, 104);
        this.buildings[7].Size = new System.Drawing.Size(211, 240);
        this.buildings[7].TabIndex = 7;
        this.buildings[7].Click += (s, e) =>
        {
            index = 7; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[8].BackgroundImage = global::urbanistics.Properties.Resources._36;
        this.buildings[8].Location = new System.Drawing.Point(368, 663);
```

```
        this.buildings[8].Size = new System.Drawing.Size(154, 211);
        this.buildings[8].TabIndex = 17;
        this.buildings[8].Click += (s, e) =>
        {
            index = 8; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[9].BackgroundImage = global::urbanistics.Properties.Resources._33;
        this.buildings[9].Location = new System.Drawing.Point(266, 670);
        this.buildings[9].Size = new System.Drawing.Size(61, 50);
        this.buildings[9].TabIndex = 18;
        this.buildings[9].Click += (s, e) =>
        {
            index = 9; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[10].BackgroundImage = global::urbanistics.Properties.Resources._45;
        this.buildings[10].Location = new System.Drawing.Point(547, 729);
        this.buildings[10].Size = new System.Drawing.Size(131, 124);
        this.buildings[10].TabIndex = 19;
        this.buildings[10].Click += (s, e) =>
        {
            index = 10; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[11].BackgroundImage = global::urbanistics.Properties.Resources._21;
        this.buildings[11].Location = new System.Drawing.Point(1, 215);
        this.buildings[11].Size = new System.Drawing.Size(282, 335);
        this.buildings[11].TabIndex = 11;
        this.buildings[11].Click += (s, e) =>
        {
```

```
            index = 11; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[12].BackgroundImage = global::urbanistics.Properties.Resources._12;
        this.buildings[12].Location = new System.Drawing.Point(1095, 3);
        this.buildings[12].Size = new System.Drawing.Size(230, 189);
        this.buildings[12].TabIndex = 12;
        this.buildings[12].Click += (s, e) =>
        {
            index = 12; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[13].BackgroundImage = global::urbanistics.Properties.Resources._37;
        this.buildings[13].Location = new System.Drawing.Point(726, 860);
        this.buildings[13].Size = new System.Drawing.Size(106, 139);
        this.buildings[13].TabIndex = 20;
        this.buildings[13].Click += (s, e) =>
        {
            index = 13; is_checked = true; display_buildings_data(index);
display_connect(index);
        };

        this.buildings[14].BackgroundImage = global::urbanistics.Properties.Resources._47;
        this.buildings[14].Location = new System.Drawing.Point(338, 447);
        this.buildings[14].Size = new System.Drawing.Size(105, 123);
        this.buildings[14].TabIndex = 21;
        this.buildings[14].Click += (s, e) =>
        {
            index = 14; is_checked = true; display_buildings_data(index);
display_connect(index);
        };
        connect_begin();
```

```
for (int i = 0; i < 2; i++)
{
    stations[i] = new PictureBox();
    stations[i].BackColor = System.Drawing.Color.Transparent;
    stations[i].TabStop = false;
    panel1.Controls.Add(this.stations[i]);
    stations[i].BringToFront();

}


this.stations[0].Location = new System.Drawing.Point(1078, 729);
this.stations[0].Size = new System.Drawing.Size(22, 21);
this.stations[0].TabIndex = 22;
this.stations[0].Click += (s, e) => { index = 15; is_checked = true;
display_station_data(index); };

this.stations[1].Location = new System.Drawing.Point(1404, 562);
this.stations[1].Size = new System.Drawing.Size(21, 20);
this.stations[1].TabIndex = 23;
this.stations[1].Click += (s, e) => { index = 16; is_checked = true;
display_station_data(index); };



for (int i = 0; i < 5; i++)
{
    streets[i] = new PictureBox();
    streets[i].BackColor = System.Drawing.Color.Transparent;
    streets[i].TabStop = false;
    panel1.Controls.Add(this.streets[i]);
```

```
        }

        this.streets[0].BackgroundImage = global::urbanistics.Properties.Resources.marker;
        this.streets[0].Location = new System.Drawing.Point(487, 368);
        this.streets[0].Size = new System.Drawing.Size(25, 31);
        this.streets[0].TabIndex = 29;
        this.streets[0].Click += (s, e) => { index = 17; is_checked = true;
display_street_data(index); };

        this.streets[1].BackgroundImage = global::urbanistics.Properties.Resources.marker;
        this.streets[1].Location = new System.Drawing.Point(1475, 798);
        this.streets[1].Size = new System.Drawing.Size(25, 31);
        this.streets[1].TabIndex = 25;
        this.streets[1].Click += (s, e) => { index = 18; is_checked = true;
display_street_data(index); };

        this.streets[2].BackgroundImage = global::urbanistics.Properties.Resources.marker;
        this.streets[2].Location = new System.Drawing.Point(1442, 539);
        this.streets[2].Size = new System.Drawing.Size(25, 31);
        this.streets[2].TabIndex = 26;
        this.streets[2].Click += (s, e) => { index = 19; is_checked = true;
display_street_data(index); };

        this.streets[3].BackgroundImage = global::urbanistics.Properties.Resources.marker;
        this.streets[3].Location = new System.Drawing.Point(66, 136);
        this.streets[3].Size = new System.Drawing.Size(25, 31);
        this.streets[3].TabIndex = 30;
        this.streets[3].Click += (s, e) => { index = 20; is_checked = true;
display_street_data(index); };

        this.streets[4].BackgroundImage = global::urbanistics.Properties.Resources.marker;
        this.streets[4].Location = new System.Drawing.Point(154, 729);
        this.streets[4].Size = new System.Drawing.Size(25, 31);
        this.streets[4].TabIndex = 28;
```

```
    this.streets[4].Click += (s, e) =>
    {
        index = 21; is_checked = true; display_street_data(index);
    };

}


Panel panel2 = new Panel();

ShapeContainer container = new ShapeContainer();
//ქვემოთ მოცემული მატრიცის საშუალებით უკავშირდებიან ერთმანეთს რუკაზე
არსებული 15 ობიექტი
LineShape[,] lines = new LineShape[15, 15];
int[] connection = new int[15];
int[,] connect = new int[,]{


    {0,0,0,1,1,0,1,0,0,1,0,0,0,0,1},
                {1,0,1,1,0,0,0,1,1,0,1,1,0,1,0},
                {1,0,0,1,0,0,0,1,1,0,0,1,0,1,0},
                {1,1,1,0,1,0,0,1,1,0,1,0,0,1,0},
                {1,0,0,0,0,0,0,1,1,0,1,1,0,1,0},
                {1,0,0,0,0,0,0,1,1,0,1,0,0,1,0},
                {1,0,0,0,0,0,0,1,1,0,1,1,0,1,0},
                {0,1,1,0,0,1,0,0,0,1,0,0,1,0,1},
                {0,1,1,1,1,1,1,0,0,1,0,0,1,1,0},
                {0,0,1,0,0,1,0,1,0,0,1,0,0,1,0},
                {0,1,0,1,1,0,1,0,0,1,0,0,1,1,1},
                {0,0,1,0,1,0,0,0,0,1,0,0,1,0,0},
                {1,0,0,1,0,1,0,1,1,0,1,1,0,1,0},
                {0,1,0,0,0,1,1,0,0,1,0,0,0,1,1},
                {1,0,0,0,0,0,0,1,0,0,1,0,0,1,0}

    };
```

```
private void connect_begin()
{

    panel2.Location = new Point(0, 0);
    panel2.BackgroundImage =
global::urbanistics.Properties.Resources.TestDrawToBitmap;
    panel2.Visible = false;
    panel1.Controls.Add(panel2);
    panel2.Size = panel1.Size;
    panel2.BringToFront();
    panel2.Click += (s, e) => { panel2.Visible = false; };
    container.Location = new Point(0, 0);
    container.Size = new System.Drawing.Size(panel1.Width,panel1.Height);
    container.TabIndex = 100;
   panel2.Controls.Add(container);
    container.BringToFront();

    int x = 0, y = 0;




    for (int i = 0; i < 15; i++)
    {
        for (int j = 0; j < 15; j++)
        {
```

```
            lines[i, j] = new LineShape();
            lines[i, j].BorderColor = Color.Blue;
            lines[i, j].BorderWidth = 3;
            lines[i, j].X1 = buildings[i].Location.X + buildings[i].Width / 2;
            lines[i, j].Y1 = buildings[i].Location.Y + buildings[i].Height / 2;
            lines[i, j].X2 = buildings[j].Location.X + buildings[j].Width / 2;
            lines[i, j].Y2 = buildings[j].Location.Y + buildings[j].Height / 2;
          //  container.Shapes.Add(lines[i, j]);
          // lines[i, j].Visible = false;
          // lines[i, j].BringToFront();
        }
    }

    Timer opac = new Timer();
    opac.Interval = 1;
    opac.Tick += (s, k) => { this.Opacity = 100; opac.Stop(); };
    opac.Start();
}



public void display_connect(int id)
{
    panel2.Visible = true;
    int id_2 = 0;
    for (int i = 0; i < 15; i++)
    {
        for (int j = 0; j < 15; j++)
        {
            try
            {
                container.Shapes.Remove(lines[i, j]); //უზრუნველყოს კავშირის გაწყვეტას
            }
            catch (Exception) { };
        }
```

```
        }
        for (int i = 0; i < 15; i++)
        {
            if (connect[id, i] == 1)
            {
                connection[i] = 1;

                container.Shapes.Add(lines[id, i]);

            }
        }

//ქვემოთ მოცემული კოდის საშუალებით ყოველი მომდევნო კავშირის საფეხური 2
წამითაა დაშორებული ერთმანეთს
        Timer time = new Timer();
        time.Enabled = true;
        time.Interval = 2000;
        time.Tick += (s, e) => { display_connect_2(id_2); id_2++; if (id_2 == 14) { time.Stop(); }
};
    }
    public void display_connect_2(int id)
    {
        for (int i = 0; i < 15; i++)
        {
            if (connect[id, i] == 1)
            {
                container.Shapes.Add(lines[id, i]);

            }
        }
    }
```

```
        public void display_buildings_data(int id)
        {
            if (!chart_set) { grafiki_begin(); chart_set = true; }
            OleDbConnection con = new OleDbConnection();
            OleDbCommand cmd = new OleDbCommand();
            OleDbDataReader reader;
            // con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\ssc\Documents\Visual Studio 2010\Projects\ruka\ruka\datab.accdb;Persist
Security Info=True";
//ბაზასთან კავშირი
    con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
Application.StartupPath + @"\cache.accdb;Persist Security Info=True";

            cmd.Connection = con;
//შენობების ინფორმაციის გამოძახება სქლ ბრძანების საშუალებით
            cmd.CommandText = "select * from shenobebi where shenoba_ID=" + (id +
1).ToString();
            try
            {
                con.Open();
```

```
            reader = cmd.ExecuteReader();
            listBox1.Items.Clear(); //შლის აქამდე არსებულ ტექსტს

            while (reader.Read()) //ინფორმაციის ამოწურვამდე კითხულობს
ინფორმაციას მონაცემთა ბაზიდან.
            {
               listBox1.Items.Add("შენობის სახელი:   " + reader["shenoba_name"]);
               listBox1.Items.Add("მისამართი:   " + reader["shenoba_misamarti"]);
               listBox1.Items.Add("კატეგორია:    " + reader["shenoba_kategoria"]);
               listBox1.Items.Add("კუთვნილება:    " + reader["shenoba_kutvnileba"]);

               listBox1.Items.Add("მოსახლეობის რაოდენობა:   " +
reader["shenoba_mosaxleobis_raod"]);
               listBox1.Items.Add("ფართობი:    " + reader["shenoba_farti"]);
               listBox1.Items.Add("ელექტროობა:    " + reader["shenoba_eleqtrooba"]);
               listBox1.Items.Add("გაზი:   " + reader["shenoba_gazi"]);
               listBox1.Items.Add("წყალი:    " + reader["shenoba_wyali"]);



            }




            reader.Close();
            con.Close();
         }
         catch (Exception ex)
         {
            MessageBox.Show(ex.Message);
         }
      }
```

```
        public void display_station_data(int id)
        {
            OleDbConnection con = new OleDbConnection();
            OleDbCommand cmd = new OleDbCommand();
            OleDbDataReader reader;
            con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
Application.StartupPath + @"\cache.accdb;Persist Security Info=True";
            cmd.Connection = con;
            cmd.CommandText = "select * from gacherebebi where gacherebebi_ID=" + (id +
1).ToString();
            try
            {
                con.Open();

                reader = cmd.ExecuteReader();
                listBox1.Items.Clear();
                while (reader.Read())
                {

                    listBox1.Items.Add("გაჩერების სახელი(ID):     " + reader["gacherebebi_name"]);
                    listBox1.Items.Add("ავტობუსის ნომრები:    " + reader["avtobusis_numbers"]);
                }



                reader.Close();
                con.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
```

```
    }



    public void display_street_data(int id)
    {
        OleDbConnection con = new OleDbConnection();
        OleDbCommand cmd = new OleDbCommand();
        OleDbDataReader reader;
        con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
Application.StartupPath + @"\cache.accdb;Persist Security Info=True";
        cmd.Connection = con;
        cmd.CommandText = "select * from quchebi where qucha_ID=" + (id + 1).ToString();
        try
        {
            con.Open();

            reader = cmd.ExecuteReader();
            listBox1.Items.Clear();
            while (reader.Read())
            {

                listBox1.Items.Add("დასახელება:    " + reader["qucha_name"]);
                listBox1.Items.Add("გამტარუნარიანობა:    " +
reader["qucha_gamtarunarianoba"]);
                listBox1.Items.Add("მოძრაობის მიმართულების რაოდენობა:    " +
reader["qucha_modzraobis mimartulebis_raod"]);
            }



            reader.Close();
            con.Close();
        }
```

```
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }


    private void button1_Click(object sender, EventArgs e) //უზრუნველყოფს ობიექტის
წაშლას
    {
        if (!chart_set) { grafiki_begin(); chart_set = true; }
        listBox1.BringToFront();
        panel2.Visible = false;
        if (is_checked)
        {
            initialize_input_param();
            panel1.Controls.Remove(buildings[index]); //შლის ინდექსის მიხედვით
ობიექტს

            deleterows(index);
        }

    }


    private void deleterows(int index)
    {
      /* try
        {
            OleDbConnection con = new OleDbConnection();
            OleDbCommand cmd = new OleDbCommand();
            OleDbDataReader reader;
```

```
            // con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\ssc\Documents\Visual Studio 2010\Projects\ruka\ruka\datab.accdb;Persist
Security Info=True";
            con.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
Application.StartupPath + @"\cache.accdb;Persist Security Info=True";

            cmd.Connection = con;
            cmd.CommandText = "DELETE FROM shenobebi WHERE shenoba_ID=" + (index
+ 1).ToString();
            con.Open();

            reader = cmd.ExecuteReader();
            reader.Close();
            con.Close();
          }
          catch (Exception) { }*/


        }

        private void button2_Click(object sender, EventArgs e) //უზრუნველყოფს ახალი
ობიექტის დამატებას
        {
          if (!chart_set) { grafiki_begin(); chart_set = true; }
          panel2.Visible = false;
          DialogResult dialog = MessageBox.Show("დააწკაპეთ მაუსი სასურველ ადგილზე
სადაც გსურთ შენობის დამატება", "", MessageBoxButtons.OKCancel);

          if (dialog == DialogResult.OK)
          {
            initialize_input_param();
            can_click = true;
            panel1.MouseClick += new MouseEventHandler(panel1_MouseClick);
          }
```

```
    //Form2 form2 = new Form2();
    // form2.ShowDialog();




}

void panel1_MouseClick(object sender, MouseEventArgs e)
{

    if (can_click)
    {
        try
        {
            mouse_x = e.X;
            mouse_y = e.Y;
          //  Form2 form2 = new Form2();
          //  form2.ShowDialog();
            insert_building();
        }
        catch (Exception)
        {

        }
    }

}
public void insert_building() //ობიექტის დამატება
{
    int array_index = buildings.Length + 1;
    if (can_click)
```

```
{

    Array.Resize(ref buildings, array_index);
    buildings[array_index - 1] = new PictureBox();
    buildings[array_index - 1].BackColor = System.Drawing.Color.Transparent;
    buildings[array_index - 1].Image = global::urbanistics.Properties.Resources._1;
    buildings[array_index - 1].Size = new System.Drawing.Size(144, 127);
    mouse_x -= buildings[array_index - 1].Width / 2;
    mouse_y -= buildings[array_index - 1].Height / 2;
    buildings[array_index - 1].Location = new System.Drawing.Point(mouse_x,
mouse_y);

    buildings[array_index - 1].TabIndex = 0;
    buildings[array_index - 1].Click += (s, e) => { index = array_index - 1; is_checked =
true; display_buildings_data(array_index - 1); };
    panel1.Controls.Add(buildings[array_index - 1]);
    can_click = false;

}
else
{

}

}

bool label_set = false;
ShapeContainer xazebi = new ShapeContainer();
Label mos_raod = new Label();
Label maR_obieqtebi_tabpage = new Label();
Label safari_obieqtebi_tabpageareoba = new Label();
Label saswavlo_obieqtebi_tabpageareoba = new Label();
Label saavadmyofo_obieqtebi_tabpageareoba = new Label();
Label gasartobi_obieqtebi_tabpageareoba = new Label();
```

```
Label parkireba_obieqtebi_tabpageareoba = new Label();


public void obieqtebi_tabpageareoba(string hide)
{
    if (!label_set)
    {
        label_set = true;
        xazebi.Location = new Point(0, 0);
        xazebi.Size = obieqtebi_tabpage.Size;
        obieqtebi_tabpage.Controls.Add(xazebi);
        double x1 = Convert.ToDouble(datagrid.Rows[0].Cells[3].Value); double x2 =
Convert.ToDouble(datagrid.Rows[0].Cells[4].Value);
        double maRazia1 = Convert.ToDouble(datagrid2.Rows[0].Cells[6].Value), maRazia2
= Convert.ToDouble(datagrid2.Rows[0].Cells[7].Value);
        double safari1 = Convert.ToDouble(datagrid2.Rows[2].Cells[6].Value), safari2 =
Convert.ToDouble(datagrid2.Rows[2].Cells[7].Value);
        double saswavlo1 = Convert.ToDouble(datagrid2.Rows[4].Cells[6].Value),
saswavlo2 = Convert.ToDouble(datagrid2.Rows[4].Cells[7].Value);
        double saavadmyofo1 = Convert.ToDouble(datagrid2.Rows[5].Cells[6].Value),
saavadmyofo2 = Convert.ToDouble(datagrid2.Rows[5].Cells[7].Value);
        double gasarTobi1 = Convert.ToDouble(datagrid2.Rows[6].Cells[6].Value),
gasarTobi2 = Convert.ToDouble(datagrid2.Rows[6].Cells[7].Value);
        double parkireba1 = Convert.ToDouble(datagrid2.Rows[7].Cells[6].Value),
parkireba2 = Convert.ToDouble(datagrid2.Rows[7].Cells[7].Value);
        double maRazia = Convert.ToDouble(datagrid2.Rows[0].Cells[5].Value), safari =
Convert.ToDouble(datagrid2.Rows[2].Cells[5].Value), saswavlo =
Convert.ToDouble(datagrid2.Rows[4].Cells[5].Value), saavadmyofo =
Convert.ToDouble(datagrid2.Rows[5].Cells[5].Value), gasarTobi =
Convert.ToDouble(datagrid2.Rows[6].Cells[5].Value), parkireba =
Convert.ToDouble(datagrid2.Rows[7].Cells[5].Value);

        obieqtebi_tabpage.Enter += (s, e) => { button1.SendToBack();
button2.SendToBack(); listBox1.SendToBack(); };
        obieqtebi_tabpage.Text = "რაიონის მდგომარეობა";
```

```
            obieqtebi_tabpage.AutoScroll = true;
            //control.Controls.Add(obieqtebi_tabpage);

            mos_raod.AutoSize = true;
            mos_raod.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
            mos_raod.Font = new Font("sylfaen", 14, FontStyle.Bold);
            mos_raod.MaximumSize = obieqtebi_tabpage.Size;
            mos_raod.Text = "I რაიონში ცხოვრობს " + x1.ToString() + " ადამიანი\n II
რაიონში ცხოვრობს" + x2.ToString() + "  ადამიანი";
            obieqtebi_tabpage.Controls.Add(mos_raod);
            mos_raod.Location = new Point(this.Width / 2 - mos_raod.Width / 2, 50);



            maR_obieqtebi_tabpage.AutoSize = true;
            maR_obieqtebi_tabpage.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            maR_obieqtebi_tabpage.Font = new Font("sylfaen", 14, FontStyle.Bold);
            maR_obieqtebi_tabpage.MaximumSize = obieqtebi_tabpage.Size;

            maR_obieqtebi_tabpage.Text = "I რაიონში არსებული მაღაზიები მოთხოვნას
აკმაყოფილებენ " + Math.Round((maRazia1 / (x1 * maRazia)) * 100, 3).ToString() + " % ით\n
II რაიონში მაღაზიები მოთხოვნას აკმაყოფილებენ " + Math.Round((maRazia2 / (x2 *
maRazia)) * 100, 3).ToString() + " % ით\n I რაიონში საჭიროა დამატებით " + (x1 *
maRazia - maRazia1).ToString() + " კვ.მ მაღაზია\n II რაიონში საჭიროა დამატებით " +
(x2 * maRazia - maRazia2).ToString() + " კვ.მ მაღაზია";
            obieqtebi_tabpage.Controls.Add(maR_obieqtebi_tabpage);
            maR_obieqtebi_tabpage.Location = new Point(this.Width / 2 -
maR_obieqtebi_tabpage.Width / 2, mos_raod.Location.Y + mos_raod.Height + 20);


            LineShape shape1 = new LineShape();
            shape1.X1 = 0;
            shape1.X2 = obieqtebi_tabpage.Width;
            shape1.Y1 = shape1.Y2 = maR_obieqtebi_tabpage.Location.Y;
```

```
            shape1.BorderColor = Color.Gray;
            shape1.BorderWidth = 3;
            shape1.BorderStyle = System.Drawing.Drawing2D.DashStyle.Dash;
            xazebi.Shapes.Add(shape1);



            safari_obieqtebi_tabpageareoba.AutoSize = true;
            safari_obieqtebi_tabpageareoba.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            safari_obieqtebi_tabpageareoba.Font = new Font("sylfaen", 14, FontStyle.Bold);
            safari_obieqtebi_tabpageareoba.MaximumSize = obieqtebi_tabpage.Size;
            safari_obieqtebi_tabpageareoba.Text = "I რაიონში არსებული მწვანე საფარი
მოთხოვნას აკმაყოფილებს " + Math.Round((safari1 / (x1 * safari)) * 100, 3).ToString() + " %
ით\n II რაიონში არსებული მწვანე საფარი მოთხოვნას აკმაყოფილებს " +
Math.Round((safari2 / (x2 * safari)) * 100, 3).ToString() + " % ით\n I რაიონში საჭიროა
დამატებით " + (x1 * safari - safari1).ToString() + " კვ.მ მწვანე საფარი\n II რაიონში
საჭიროა დამატებით " + (x2 * safari - safari2).ToString() + " კვ.მ მწვანე საფარი";
            obieqtebi_tabpage.Controls.Add(safari_obieqtebi_tabpageareoba);
            safari_obieqtebi_tabpageareoba.Location = new Point(this.Width / 2 -
safari_obieqtebi_tabpageareoba.Width / 2, maR_obieqtebi_tabpage.Location.Y +
maR_obieqtebi_tabpage.Height + 20);

            LineShape shape2 = new LineShape();
            shape2.X1 = 0;
            shape2.X2 = obieqtebi_tabpage.Width;
            shape2.Y1 = shape2.Y2 = safari_obieqtebi_tabpageareoba.Location.Y;
            shape2.BorderColor = Color.Gray;
            shape2.BorderWidth = 3;
            shape2.BorderStyle = System.Drawing.Drawing2D.DashStyle.Dash;
            xazebi.Shapes.Add(shape2);


            saswavlo_obieqtebi_tabpageareoba.AutoSize = true;
```

```
            saswavlo_obieqtebi_tabpageareoba.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            saswavlo_obieqtebi_tabpageareoba.Font = new Font("sylfaen", 14, FontStyle.Bold);
            saswavlo_obieqtebi_tabpageareoba.MaximumSize = obieqtebi_tabpage.Size;
            saswavlo_obieqtebi_tabpageareoba.Text = "I რაიონში არსებული სასწავლო
ობიექტები მოთხოვნას აკმაყოფილებენ " + Math.Round((saswavlo1 / (x1 * saswavlo)) *
100, 3).ToString() + " % ით\n II რაიონში სასწავლო ობიექტები მოთხოვნას
აკმაყოფილებენ " + Math.Round((saswavlo2 / (x2 * saswavlo)) * 100, 3).ToString() + " %
ით\n I რაიონში საჭიროა დამატებით " + (x1 * saswavlo - saswavlo1).ToString() + " კვ.მ
სასწავლო ობიექტი\n II რაიონში საჭიროა დამატებით " + (x2 * saswavlo -
saswavlo2).ToString() + " კვ.მ სასწავლო ობიექტი";
            obieqtebi_tabpage.Controls.Add(saswavlo_obieqtebi_tabpageareoba);
            saswavlo_obieqtebi_tabpageareoba.Location = new Point(this.Width / 2 -
saswavlo_obieqtebi_tabpageareoba.Width / 2, safari_obieqtebi_tabpageareoba.Location.Y +
safari_obieqtebi_tabpageareoba.Height + 20);

            LineShape shape3 = new LineShape();
            shape3.X1 = 0;
            shape3.X2 = obieqtebi_tabpage.Width;
            shape3.Y1 = shape3.Y2 = saswavlo_obieqtebi_tabpageareoba.Location.Y;
            shape3.BorderColor = Color.Gray;
            shape3.BorderWidth = 3;
            shape3.BorderStyle = System.Drawing.Drawing2D.DashStyle.Dash;
            xazebi.Shapes.Add(shape3);
            //1

            saavadmyofo_obieqtebi_tabpageareoba.AutoSize = true;
            saavadmyofo_obieqtebi_tabpageareoba.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            saavadmyofo_obieqtebi_tabpageareoba.Font = new Font("sylfaen", 14,
FontStyle.Bold);
            saavadmyofo_obieqtebi_tabpageareoba.MaximumSize = obieqtebi_tabpage.Size;
            saavadmyofo_obieqtebi_tabpageareoba.Text = "I რაიონში არსებული
საავადმყოფოები მოთხოვნას აკმაყოფილებენ " + Math.Round((saavadmyofo1 / (x1 *
saavadmyofo)) * 100, 3).ToString() + " % ით\n II რაიონში საავადმყოფოები მოთხოვნას
```

```
აკმაყოფილებენ " + Math.Round((saavadmyofo2 / (x2 * saavadmyofo)) * 100, 3).ToString() + 
" % ით\n I რაიონში საჭიროა დამატებით " + (x1 * saavadmyofo - 
saavadmyofo1).ToString() + " კვ.მ საავადმყოფო \n II რაიონში საჭიროა დამატებით " + 
(x2 * saavadmyofo - saavadmyofo2).ToString() + " კვ.მ საავადმყოფო";
            obieqtebi_tabpage.Controls.Add(saavadmyofo_obieqtebi_tabpageareoba);
            saavadmyofo_obieqtebi_tabpageareoba.Location = new Point(this.Width / 2 - 
saavadmyofo_obieqtebi_tabpageareoba.Width / 2, 
saswavlo_obieqtebi_tabpageareoba.Location.Y + saswavlo_obieqtebi_tabpageareoba.Height + 
20);

            LineShape shape4 = new LineShape();
            shape4.X1 = 0;
            shape4.X2 = obieqtebi_tabpage.Width;
            shape4.Y1 = shape4.Y2 = saavadmyofo_obieqtebi_tabpageareoba.Location.Y;
            shape4.BorderColor = Color.Gray;
            shape4.BorderWidth = 3;
            shape4.BorderStyle = System.Drawing.Drawing2D.DashStyle.Dash;
            xazebi.Shapes.Add(shape4);
            //1

            gasartobi_obieqtebi_tabpageareoba.AutoSize = true;
            gasartobi_obieqtebi_tabpageareoba.TextAlign = 
System.Drawing.ContentAlignment.MiddleCenter;
            gasartobi_obieqtebi_tabpageareoba.Font = new Font("sylfaen", 14, FontStyle.Bold);
            gasartobi_obieqtebi_tabpageareoba.MaximumSize = obieqtebi_tabpage.Size;
            gasartobi_obieqtebi_tabpageareoba.Text = "I რაიონში არსებული გასართობი 
ადგილები მოთხოვნას აკმაყოფილებენ " + Math.Round((gasarTobi1 / (x1 * gasarTobi)) * 
100, 3).ToString() + " % ით\n II რაიონში გასართობი ადგილები მოთხოვნას 
აკმაყოფილებენ " + Math.Round((gasarTobi2 / (x2 * gasarTobi)) * 100, 3).ToString() + " % 
ით\n I რაიონში საჭიროა დამატებით " + (x1 * gasarTobi - gasarTobi1).ToString() + " კვ.მ 
გასართობი ადგილი\n II რაიონში საჭიროა დამატებით " + (x2 * gasarTobi - 
gasarTobi2).ToString() + " კვ.მ გასართობი ადგილი";
            obieqtebi_tabpage.Controls.Add(gasartobi_obieqtebi_tabpageareoba);
            gasartobi_obieqtebi_tabpageareoba.Location = new Point(this.Width / 2 - 
gasartobi_obieqtebi_tabpageareoba.Width / 2, 
```

```
saavadmyofo_obieqtebi_tabpageareoba.Location.Y +
saavadmyofo_obieqtebi_tabpageareoba.Height + 20);

            LineShape shape5 = new LineShape();
            shape5.X1 = 0;
            shape5.X2 = obieqtebi_tabpage.Width;
            shape5.Y1 = shape5.Y2 = gasartobi_obieqtebi_tabpageareoba.Location.Y;
            shape5.BorderColor = Color.Gray;
            shape5.BorderWidth = 3;
            shape5.BorderStyle = System.Drawing.Drawing2D.DashStyle.Dash;
            xazebi.Shapes.Add(shape5);
            //1

            parkireba_obieqtebi_tabpageareoba.AutoSize = true;
            parkireba_obieqtebi_tabpageareoba.TextAlign =
System.Drawing.ContentAlignment.MiddleCenter;
            parkireba_obieqtebi_tabpageareoba.Font = new Font("sylfaen", 14, FontStyle.Bold);
            parkireba_obieqtebi_tabpageareoba.MaximumSize = obieqtebi_tabpage.Size;
            parkireba_obieqtebi_tabpageareoba.Text = "I რაიონში არსებული
პარკირებისთვის განკუთვნილი ადგილი მოთხოვნას აკმაყოფილებს " +
Math.Round((parkireba1 / (x1 * parkireba)) * 100, 3).ToString() + " % ით\n II რაიონში
პარკირებისთვის განკუთვნილი ადგილი მოთხოვნას აკმაყოფილებს " +
Math.Round((parkireba2 / (x2 * parkireba)) * 100, 3).ToString() + " % ით\n I რაიონში
საჭიროა დამატებით " + (x1 * parkireba - parkireba1).ToString() + " კვ.მ პარკირებისთვის
განკუთვნილი ადგილი\n II რაიონში საჭიროა დამატებით " + (x2 * parkireba -
parkireba2).ToString() + " კვ.მ პარკირებისთვის განკუთვნილი ადგილი";
            obieqtebi_tabpage.Controls.Add(parkireba_obieqtebi_tabpageareoba);
            parkireba_obieqtebi_tabpageareoba.Location = new Point(this.Width / 2 -
parkireba_obieqtebi_tabpageareoba.Width / 2,
gasartobi_obieqtebi_tabpageareoba.Location.Y + gasartobi_obieqtebi_tabpageareoba.Height +
20);

            LineShape shape6 = new LineShape();
            shape6.X1 = 0;
            shape6.X2 = obieqtebi_tabpage.Width;
```

```
        shape6.Y1 = shape6.Y2 = parkireba_obieqtebi_tabpageareoba.Location.Y;
        shape6.BorderColor = Color.Gray;
        shape6.BorderWidth = 3;
        shape6.BorderStyle = System.Drawing.Drawing2D.DashStyle.Dash;
        xazebi.Shapes.Add(shape6);
    }
     if (hide == "hide")
    {
        xazebi.Visible = false;
        mos_raod.Visible = false;
        maR_obieqtebi_tabpage.Visible = false;
        safari_obieqtebi_tabpageareoba.Visible = false;
        saswavlo_obieqtebi_tabpageareoba.Visible = false;
        saavadmyofo_obieqtebi_tabpageareoba.Visible = false;
        gasartobi_obieqtebi_tabpageareoba.Visible = false;
        parkireba_obieqtebi_tabpageareoba.Visible = false;
    }
     if (hide == "show")
    {
        xazebi.Visible = true;
        mos_raod.Visible = true;
        maR_obieqtebi_tabpage.Visible = true;
        safari_obieqtebi_tabpageareoba.Visible = true;
        saswavlo_obieqtebi_tabpageareoba.Visible = true;
        saavadmyofo_obieqtebi_tabpageareoba.Visible = true;
        gasartobi_obieqtebi_tabpageareoba.Visible = true;
        parkireba_obieqtebi_tabpageareoba.Visible = true;
    }



}
```

```
bool graphics_set = false;
Random rand = new Random();


public void initialize_input_param()
{
    double[] x;
    double[] alfa, h, v,teta , r_standarti;
    double gare_v;
    string[] interesebis_dasaxeleba;
    double[,] r_realuri;

    /*************************************
        * საწყისი პარამეტრებით აქ აღწერილი ის პარამეტრები რომელიც ჭირდება
პროგრამას ოპერაციის ჩასატარებლად  *
        * **********************************/
    x = new double[2];
    x[0] = Convert.ToDouble ( datagrid.Rows[0].Cells[3].Value);
    x[1] = Convert.ToDouble(datagrid.Rows[0].Cells[4].Value);

    alfa = new double[2];
    alfa[0] = Convert.ToDouble(datagrid.Rows[1].Cells[3].Value);
    alfa[1] = Convert.ToDouble(datagrid.Rows[1].Cells[4].Value);
    h = new double[2];
    h[0] = Convert.ToDouble(datagrid.Rows[2].Cells[3].Value);
    h[1] = Convert.ToDouble(datagrid.Rows[2].Cells[4].Value);
    v = new double[2];
    v[0] = Convert.ToDouble(datagrid.Rows[3].Cells[3].Value);
    v[1] = Convert.ToDouble(datagrid.Rows[3].Cells[4].Value);
    gare_v = 9600;
```

```
        interesebis_dasaxeleba = new string[] { "interesi1", "interesi3", "interesi4", "interesi5",
"interesi6", "interesi7", "interesi8", "interesi9", "interesi10" };
        teta = new double[10];// {0.2,0.12,0.3,0.24,0.3,0.4,0.3,0.2,0.5,0.41};
        r_standarti = new double[10];//{0.1,0.3,0.12,0.5,0.24,0.3,0.4,0.2,0.32,0.4};
        r_realuri = new double[10,
2];//{{0.1,0.2},{0.21,0.18},{0.15,0.18},{0.13,0.26},{0.6,0.58},{0.3,0.21},{0.1,0.16},{0.2,0.22},{0.32,
0.33},{0.5,0.6}};


        /***************************
         * მოცემული ციკლი ავსებს ინტერესების მასიებს შემთხვევითი რიცხვებით
*
         * ************************/
        for (int i = 0; i < 10; i++)
        {
           teta[i] = (rand.Next(10));
           teta[i] = teta[i] / 10;
           r_standarti[i] = rand.Next(10);
           r_standarti[i] = r_standarti[i] / 10;
           for (int j = 0; j < 2; j++)
           {
              r_realuri[i, j] = rand.Next(10);
              r_realuri[i, j] = r_realuri[i, j] / 10;
           }



        }

        check_data(ref  teta);

        calculate_produced_parameters(r_realuri,r_standarti, alfa, v, gare_v, h, teta, x);

     }
```

```
/********************************
 * check data ამოწმებს მნიშვნელობებს *
 * ნორმალურობის პირობებს          *
 * და საჭიროების შემთხვევაში ასწორებს მათ*
 * ******************************/
public void check_data(ref double[] data_array)
{
    double sum = 0;
    for (int i = 0; i < 10; i++)
    {
        sum += data_array[i];
    }
    if (sum != 1)
    {
        for (int i = 0; i < 10; i++)
        {
            data_array[i] /= sum;
        }
    }


}
/****************************************
 * ითვლის წარმოებული პარამეტრების მნიშვნელობებს
 * **************************************/
public void calculate_produced_parameters(double[,] r_realuri , double[] r_standarti,
double[] alfa , double[] v , double gare_v , double []h, double[] teta,double[]x  )
{
    double a = 0;
   double[,] r_dakmayofilebis_saWiro_norma = new double[10, 2];
  double[,]  r_dakmayofilebis_done = new double[10, 2];
   double[] e = new double[2];
   double[] k = new double[2];
   double[] lambda = new double[2];
```

```
for (int i = 0; i < 10; i++)
{

    for (int j = 0; j < 2; j++)
    {
        r_dakmayofilebis_saWiro_norma[i, j] = ((x[j] * r_standarti[i]));


        if (r_realuri[i, j] > r_standarti[i])
        {
            r_dakmayofilebis_done[i, j] = 1;
        }
        else
        {
            r_dakmayofilebis_done[i, j] = r_realuri[i, j] / r_standarti[i];
        }




    }

}


for (int i = 0; i < 2; i++)
{
    e[i] = v[i] / gare_v;
    k[i] = (alfa[i] * (h[i] - alfa[i]));
    a += (alfa[i] * e[i]);
    lambda[i] = 0;
    for (int j = 0; j < 10; j++)
    {
```

```
            lambda[i] += (r_dakmayofilebis_done[j, i] * teta[j]);
          }




        }
        build_graphics(k,alfa,e,lambda,v,gare_v,h,a);



      }
```

```
      double l1_x_coord, l2_y_coord, l3_y_coord, l4_x_coord, l5_y_coord;
      Chart chart1 = new Chart();
      ChartArea chartArea1 = new ChartArea();
//გრაფიკის დახაზვა
      Legend legend1 = new Legend();
```

```
        System.Windows.Forms.DataVisualization.Charting.Series series1 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.Series series2 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.Series series3 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.Series series4 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.Series series5 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        System.Windows.Forms.DataVisualization.Charting.Series series6 = new
System.Windows.Forms.DataVisualization.Charting.Series();
        TabPage chart_tab = new TabPage();
        Label chart_label1 = new Label();
        Label chart_label2 = new Label();
        public void grafiki_begin()
        {

            chart_setting();
        }
        public void chart_setting()
        {
            this.chart1.BackColor = System.Drawing.Color.Transparent;
            chartArea1.Name = "ChartArea1";
            this.chart1.ChartAreas.Add(chartArea1);
            legend1.Name = "Legend1";
            this.chart1.Legends.Add(legend1);
            this.chart1.Location = new System.Drawing.Point(12, 12);
            this.chart1.Name = "chart1";
            series1.ChartArea = "ChartArea1";
            series1.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series1.Legend = "Legend1";
            series1.Name = "l1";
            series2.ChartArea = "ChartArea1";
```

```
            series2.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series2.Legend = "Legend1";
            series2.Name = "l2";
            series3.ChartArea = "ChartArea1";
            series3.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series3.Legend = "Legend1";
            series3.Name = "l3";
            series4.ChartArea = "ChartArea1";
            series4.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series4.Legend = "Legend1";
            series4.Name = "l4";
            series5.ChartArea = "ChartArea1";
            series5.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
            series5.Legend = "Legend1";
            series5.Name = "l5";
            series6.ChartArea = "ChartArea1";
            series6.ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;

            series6.Legend = "Legend1";
            series6.MarkerSize = 10;

            series6.MarkerStyle =
System.Windows.Forms.DataVisualization.Charting.MarkerStyle.Circle;
            series6.Name = "position";
            this.chart1.Series.Add(series1);
            this.chart1.Series.Add(series2);
            this.chart1.Series.Add(series3);
            this.chart1.Series.Add(series4);
            this.chart1.Series.Add(series5);
            this.chart1.Series.Add(series6);
```

```
        this.chart1.Size = new System.Drawing.Size(960, 738);
        this.chart1.TabIndex = 0;
        this.chart1.Text = "chart1";
        chart1.ChartAreas[0].AxisX.Minimum = 0;
        chart1.ChartAreas[0].AxisY.Minimum = 0;
        chart1.ChartAreas[0].AxisX.Maximum = 1;
        chart1.ChartAreas[0].AxisY.Maximum = 1;
        chart1.ChartAreas[0].AxisX.MajorGrid.LineWidth = 0;
        chart1.ChartAreas[0].AxisY.MajorGrid.LineWidth = 0;
        chart_tab.Controls.Add(chart1);
        chart_label1.AutoSize = true;
        chart_label2.AutoSize = true;
        chart_label1.Font = new Font("sylfaen", 14, FontStyle.Bold);
        chart_label2.Font = new Font("sylfaen", 14, FontStyle.Bold);
        chart_tab.Controls.Add(chart_label1);
        chart_tab.Controls.Add(chart_label2);
        initialize_input_param();
      }

    public void build_graphics(double[] k, double[] alfa, double[] e, double[] lambda,
double[] v, double gare_v, double[] h ,double a)
    {

        if (!graphics_set)
        {
          graphics_set = true;
          chart1.Series["l1"].Points.Clear();
          chart1.Series["l2"].Points.Clear();
          chart1.Series["l3"].Points.Clear();
          chart1.Series["l4"].Points.Clear();
          chart1.Series["l5"].Points.Clear();

          //l1 გრაფიკის დახაზვა
          l1_x_coord = (4 * k[0]) / (alfa[0] * alfa[0] * e[0] * e[0]);
          chart1.Series["l1"].Points.AddXY(l1_x_coord, 0);
```

```
            chart1.Series["l1"].Points.AddXY(l1_x_coord, 1);


            //l2 გრაფიკის დახაზვა

            l2_y_coord = (4 * k[1]) / (alfa[1] * alfa[1] * e[1] * e[1]);
            chart1.Series["l2"].Points.AddXY(0, l2_y_coord);
            chart1.Series["l2"].Points.AddXY(1, l2_y_coord);

            //l3 გრაფიკის დახაზვა

            for (double i = 0; i < 1; i += 0.002)
            {
                l3_y_coord = ((4 * k[1]) / (a * a)) * (1 + (((4 * k[0]) / (a * a)) / ((i) - ((4 * k[0]) / (a *
a)))));


                if (l3_y_coord > 0)
                {
                    chart1.Series["l3"].Points.AddXY(i, l3_y_coord);
                }
            }

            //l4 გრაფიკის დახაზვა


            for (double i = 0; i < 1; i += 0.002)
            {

                l4_x_coord = (k[0] * alfa[1] * alfa[1] * e[1] * e[1] * (i) * (i)) / (k[1] * (alfa[1] * e[1] *
alfa[0] * e[0] * (i) - k[1]));


                if (l4_x_coord > 0)
                {
```

```
                chart1.Series["l4"].Points.AddXY(l4_x_coord, i);

            }
        }

        //l5 გრაფიკის დახაზვა

        for (double i = 0; i < 1; i += 0.002)
        {
            l5_y_coord = (k[1] * alfa[0] * alfa[0] * e[0] * e[0] * i * i) / (k[0] * (alfa[0] * e[0] *
alfa[1] * e[1] * i - k[0]));


            if (l5_y_coord > 0)
            {
                chart1.Series["l5"].Points.AddXY(i, l5_y_coord);

            }
        }

        //გრაფიკზე რაიონის წონასწორობის წერტილის დასმა
    }
    build_lambda(lambda , k ,alfa ,e ,a ,v ,gare_v ,h);
    //წონასწორობის არის განსაზღვრა




}
public void build_lambda(double[] lambda ,double[] k , double[] alfa , double[] e , double
a , double[] v , double gare_v , double[] h )
{
    chart1.Series["position"].Points.AddXY(lambda[0], lambda[1]);
    check_area(l1_x_coord, l2_y_coord, k, alfa, e, lambda, a, v, gare_v, h);
}
```

```
    public void check_area(double l1_x_coord, double l2_y_coord, double[] k, double[] alfa,
double[] e, double[] lambda, double a, double[] v, double gare_v, double[] h)
    {

        double param = (Math.Pow(a, 2) - 4 * ((k[0] / lambda[0]) + (k[1] / lambda[1])));

        double[] x = new double[2];
        string area;
        if (lambda[1] > l2_y_coord && param < 0)
        {
            area = "A არე";
            x[0] = 0;
            x[1] = ((v[1] / 2) + ((gare_v / (2 * alfa[1])) * Math.Sqrt((Math.Pow(alfa[1], 2) *
Math.Pow(e[1], 2)) - ((4 * Math.Pow(k[1], 2)) / Math.Pow(lambda[1], 2)))));
        }
        else if (lambda[0] < l1_x_coord && lambda[1] < l2_y_coord && param < 0)
        {
            area = "H არე";
            x[0] = 0;
            x[1] = 0;
        }
        else if (l1_x_coord > 0 && param < 0)
        {
            area = "M არე";
            x[0] = ((v[0] / 2) + ((gare_v / (2 * alfa[0])) * Math.Sqrt((Math.Pow(alfa[0], 2) *
Math.Pow(e[0], 2)) - ((4 * Math.Pow(k[0], 2)) / Math.Pow(lambda[0], 2)))));
            x[1] = 0;
        }
        else
        {


            area = "გარე არე";
            x[0] = v[0] - (gare_v * (h[0] - alfa[0]) * ((a - Math.Sqrt(Math.Pow(a, 2) - 4 * (k[0] /
lambda[0] + k[1] / lambda[1]))) / ((2 * lambda[0]) * (k[0] / lambda[0] + k[1] / lambda[1]))));
```

```
            x[1] = v[1] - (gare_v * (h[1] - alfa[1]) * ((a - Math.Sqrt(Math.Pow(a, 2) - 4 * (k[0] / 
lambda[0] + k[1] / lambda[1]))) / ((2 * lambda[0]) * (k[0] / lambda[0] + k[1] / lambda[1]))));
        }
        chart_label1.Text = area;
        chart_label2.Text = " x1= " + ((int)x[0]).ToString() + "\nx2= " + ((int)x[1]).ToString();
        chart_label1.Location = new Point(chart1.Location.X + chart1.Width, 
chart1.Location.Y);
        chart_label2.Location = new Point(chart_label1.Location.X, chart_label1.Location.Y + 
chart_label1.Height + 10);

    }

    Panel panel3 = new Panel();
    ShapeContainer q_container = new ShapeContainer();
    LineShape[,] sacxobi_maRazia_line = new LineShape[5, 7];
    LineShape[,] qarxana_sacxobi_line = new LineShape[4, 5];
    int[,] sacxobi_maRazia = new int[5, 
7]{{1,1,0,0,0,0,0},{0,0,1,0,0,0,0},{0,0,0,1,1,0,0},{0,0,0,0,1,1,0},{0,0,0,0,0,1,1}};
    int[,] sacxobi_maRazia_trans;
    string sacxobi_maRazia_q;
    int[,] qarxana_sacxobi = new int[4,5] {{1,0,0,0,0},{1,1,0,0,0},{0,0,1,0,0},{0,0,0,1,1} };
    int[,] qarxana_sacxobi_trans;
    string qarxana_sacxobi_q;
    PictureBox[] q_sacxobi = new PictureBox[5];
    PictureBox[] q_maRazia = new PictureBox[7];
    PictureBox[] q_qarxana = new PictureBox[4];
    TabPage q_analizi_tab = new TabPage();
    int pic_index = 0;
    string pic_type = "";
    Label situacia = new Label();
    Button ganawileba = new Button();
    public void q_analizi()
    {
        transponate1 t = new transponate1();
        sacxobi_maRazia_trans = t.transponate(sacxobi_maRazia);
```

```
qarxana_sacxobi_trans = t.transponate(qarxana_sacxobi);
multiple1 q = new multiple1();
sacxobi_maRazia_q = q.multiple(sacxobi_maRazia, sacxobi_maRazia_trans);
qarxana_sacxobi_q = q.multiple(qarxana_sacxobi_trans, qarxana_sacxobi);



panel3.BackgroundImage = global::urbanistics.Properties.Resources.ruka;
for (int i = 0; i < 7; i++)
{
   q_maRazia[i] = new PictureBox();
   q_maRazia[i].BackColor = System.Drawing.Color.Transparent;
   q_maRazia[i].TabStop = false;
   q_maRazia[i].BackgroundImageLayout = ImageLayout.Stretch;
   panel3.Controls.Add(this.q_maRazia[i]);
   add_event(ref q_maRazia[i], i, "maRazia");

}
this.q_maRazia[0].BackgroundImage = global::urbanistics.Properties.Resources._61;
this.q_maRazia[0].Location = new System.Drawing.Point(10, 600);
this.q_maRazia[0].Size = new System.Drawing.Size(100, 100);

this.q_maRazia[1].BackgroundImage = global::urbanistics.Properties.Resources._61;
this.q_maRazia[1].Location = new System.Drawing.Point(150, 600);
this.q_maRazia[1].Size = new System.Drawing.Size(110, 100);

this.q_maRazia[2].BackgroundImage = global::urbanistics.Properties.Resources._61;
this.q_maRazia[2].Location = new System.Drawing.Point(300, 600);
this.q_maRazia[2].Size = new System.Drawing.Size(100, 100);

this.q_maRazia[3].BackgroundImage = global::urbanistics.Properties.Resources._61;
this.q_maRazia[3].Location = new System.Drawing.Point(440, 600);
this.q_maRazia[3].Size = new System.Drawing.Size(100, 100);

this.q_maRazia[4].BackgroundImage = global::urbanistics.Properties.Resources._61;
```

```
this.q_maRazia[4].Location = new System.Drawing.Point(600, 600);
this.q_maRazia[4].Size = new System.Drawing.Size(100, 100);


this.q_maRazia[5].BackgroundImage = global::urbanistics.Properties.Resources._61;
this.q_maRazia[5].Location = new System.Drawing.Point(820, 600);
this.q_maRazia[5].Size = new System.Drawing.Size(100, 100);



this.q_maRazia[6].BackgroundImage = global::urbanistics.Properties.Resources._61;
this.q_maRazia[6].Location = new System.Drawing.Point(1000, 600);
this.q_maRazia[6].Size = new System.Drawing.Size(100, 100);
```

```
for (int i = 0; i < 4; i++)
{
   q_qarxana[i] = new PictureBox();
   q_qarxana[i].BackColor = System.Drawing.Color.Transparent;
   q_qarxana[i].TabStop = false;
   q_qarxana[i].BackgroundImageLayout = ImageLayout.Stretch;
   panel3.Controls.Add(this.q_qarxana[i]);

}
```

```
        this.q_qarxana[0].BackgroundImage =
global::urbanistics.Properties.Resources.qarxana;
        this.q_qarxana[0].Location = new System.Drawing.Point(100, 50);
        this.q_qarxana[0].Size = new System.Drawing.Size(131, 147);


        this.q_qarxana[1].BackgroundImage =
global::urbanistics.Properties.Resources.qarxana;
        this.q_qarxana[1].Location = new System.Drawing.Point(250, 60);
        this.q_qarxana[1].Size = new System.Drawing.Size(131, 147);


        this.q_qarxana[2].BackgroundImage =
global::urbanistics.Properties.Resources.qarxana;
        this.q_qarxana[2].Location = new System.Drawing.Point(500, 50);
        this.q_qarxana[2].Size = new System.Drawing.Size(131, 147);

        this.q_qarxana[3].BackgroundImage =
global::urbanistics.Properties.Resources.qarxana;
        this.q_qarxana[3].Location = new System.Drawing.Point(800, 70);
        this.q_qarxana[3].Size = new System.Drawing.Size(131, 147);




        for (int i = 0; i < 5; i++)
        {
          q_sacxobi[i] = new PictureBox();
          q_sacxobi[i].BackColor = System.Drawing.Color.Transparent;
          q_sacxobi[i].TabStop = false;
          q_sacxobi[i].BackgroundImageLayout = ImageLayout.Stretch;
          panel3.Controls.Add(this.q_sacxobi[i]);
```

```
    add_event(ref q_sacxobi[i], i, "sacxobi");

}
this.q_sacxobi[0].BackgroundImage = global::urbanistics.Properties.Resources.sacxobi;
this.q_sacxobi[0].Location = new System.Drawing.Point(10, 300);
this.q_sacxobi[0].Size = new System.Drawing.Size(126, 153);

this.q_sacxobi[1].BackgroundImage = global::urbanistics.Properties.Resources.sacxobi;
this.q_sacxobi[1].Location = new System.Drawing.Point(200, 300);
this.q_sacxobi[1].Size = new System.Drawing.Size(126, 153);


this.q_sacxobi[2].BackgroundImage = global::urbanistics.Properties.Resources.sacxobi;
this.q_sacxobi[2].Location = new System.Drawing.Point(380, 300);
this.q_sacxobi[2].Size = new System.Drawing.Size(126, 153);


this.q_sacxobi[3].BackgroundImage = global::urbanistics.Properties.Resources.sacxobi;
this.q_sacxobi[3].Location = new System.Drawing.Point(580, 320);
this.q_sacxobi[3].Size = new System.Drawing.Size(126, 153);


this.q_sacxobi[4].BackgroundImage = global::urbanistics.Properties.Resources.sacxobi;
this.q_sacxobi[4].Location = new System.Drawing.Point(790, 310);
this.q_sacxobi[4].Size = new System.Drawing.Size(126, 153);


q_analizi_tab.AutoScroll = true;
q_analizi_tab.Text = "Q ანალიზი";
control.Controls.Add(q_analizi_tab);
panel3.Location = new Point(0, 0);
panel3.AutoSize = true;
panel3.BackgroundImageLayout = ImageLayout.Stretch;
q_analizi_tab.Controls.Add(panel3);
```

```
        q_container.Location = new Point(0, 0);
        q_container.Size = panel3.Size;
        panel3.Controls.Add(q_container);
        q_container.BringToFront();
        Label sacxobi_maRazia_lbl = new Label();
        sacxobi_maRazia_lbl.Location = new Point(panel3.Location.X + panel3.Width + 20,
(int)(panel3.Location.Y+panel3.Height*0.75));
        sacxobi_maRazia_lbl.Font = new System.Drawing.Font("sylfaen", 14, FontStyle.Bold);
        sacxobi_maRazia_lbl.ForeColor = Color.Red;
        sacxobi_maRazia_lbl.Text ="არსებული Q ვექტორი მაღაზიასა და საცხობს
შორის\n { "+ sacxobi_maRazia_q+" }";
        sacxobi_maRazia_lbl.AutoSize = true;
        q_analizi_tab.Controls.Add(sacxobi_maRazia_lbl);
        sacxobi_maRazia_lbl.BringToFront();
        situacia.AutoSize = true;
        situacia.Location = new Point(panel3.Location.X + panel3.Width + 20,
(int)(panel3.Location.Y + panel3.Height * 0.5));
        situacia.Font = new System.Drawing.Font("sylfaen", 12);
        q_analizi_tab.Controls.Add(situacia);
       // situacia.MaximumSize = new System.Drawing.Size(200, 50);
        situacia.BringToFront();
        //situacia.Text = "ksljdhfkdlsajfhsdkjfh";

        Label qarxana_sacxobi_lbl = new Label();
        qarxana_sacxobi_lbl.Location = new Point(panel3.Location.X + panel3.Width + 20,
(int)(panel3.Location.Y + panel3.Height * 0.25));
        qarxana_sacxobi_lbl.Font = new System.Drawing.Font("sylfaen", 14, FontStyle.Bold);
        qarxana_sacxobi_lbl.ForeColor = Color.Red;
        qarxana_sacxobi_lbl.Text = "არსებული Q ვექტორი ქარხანასა და საცხობს
შორის\n { " + qarxana_sacxobi_q + " }";
        qarxana_sacxobi_lbl.AutoSize = true;
        q_analizi_tab.Controls.Add(qarxana_sacxobi_lbl);
        qarxana_sacxobi_lbl.BringToFront();
        q_analizi_tab.Enter += (s, e) => { listBox1.SendToBack(); button1.SendToBack();
button2.SendToBack(); };
```

```
        ganawileba.AutoSize = true;
        ganawileba.Location = new Point(qarxana_sacxobi_lbl.Location.X,
qarxana_sacxobi_lbl.Location.Y + qarxana_sacxobi_lbl.Height + 20);
        ganawileba.Font = new Font("sylfaen", 14, FontStyle.Bold);
        ganawileba.Text = "დატვირთვის განაწილება";
        q_analizi_tab.Controls.Add(ganawileba);
        ganawileba.Enabled = false;
        ganawileba.BringToFront();
        ganawileba.Click += (s, e) =>
          {
            int lower = 0;
            int lower_index = 0;
            if (pic_type == "maRazia")
            {
              //lower = sacxobi_maRazia[0, 0];
              int sum=0;
              bool lower_set = false;
             for(int i=0 ; i<sacxobi_maRazia.GetLength(0) ; i++)
              {
                for (int j = 0; j < sacxobi_maRazia.GetLength(1); j++)
                {
                  sum += sacxobi_maRazia[i, j];
                }
                if (sacxobi_maRazia[i, pic_index] == 0)
                {
                  if (!lower_set)
                  {
                    lower_set = true;
                    lower = sum;
                  }
                  else if (lower > sum)
                  {
                    lower = sum;
                    lower_index = i;
                  }
```

```
        }
        sum = 0;

      }
      sacxobi_maRazia_line[lower_index, pic_index].Visible = true;
      sacxobi_maRazia[lower_index, pic_index] = 1;
      for (int i = 0; i < 5; i++)
      {

        for (int j = 0; j < 7; j++)
        {
          sacxobi_maRazia_line[i, j] = new LineShape();
          sacxobi_maRazia_line[i, j].X1 = q_sacxobi[i].Location.X +
q_sacxobi[i].Width / 2;
          sacxobi_maRazia_line[i, j].Y1 = q_sacxobi[i].Location.Y +
q_sacxobi[i].Height / 2;
          sacxobi_maRazia_line[i, j].X2 = q_maRazia[j].Location.X +
q_maRazia[j].Width / 2;
          sacxobi_maRazia_line[i, j].Y2 = q_maRazia[j].Location.Y +
q_maRazia[j].Height / 2;
          sacxobi_maRazia_line[i, j].BorderColor = Color.Green;
          sacxobi_maRazia_line[i, j].BorderWidth = 3;
          q_container.Shapes.Add(sacxobi_maRazia_line[i, j]);
          if (sacxobi_maRazia[i, j] == 0)
          {
            sacxobi_maRazia_line[i, j].Visible = false;
          }
        }
      }
      t = new transponate1();
      qarxana_sacxobi_trans = t.transponate(sacxobi_maRazia);
      q = new multiple1();
      sacxobi_maRazia_q = q.multiple(sacxobi_maRazia, sacxobi_maRazia_trans);
      sacxobi_maRazia_lbl.Text = "არსებული Q ვექტორი საცხობსა და მაღაზიას
შორის\n { " + sacxobi_maRazia_q + " }";
```

```
}

else if (pic_type == "sacxobi")
{

    //lower = sacxobi_maRazia[0, 0];
    int sum = 0;
    bool lower_set = false;
    for (int i = 0; i < qarxana_sacxobi.GetLength(0); i++)
    {
        for (int j = 0; j < qarxana_sacxobi.GetLength(1); j++)
        {
            sum += qarxana_sacxobi[i, j];
        }
        if (qarxana_sacxobi[i, pic_index] == 0)
        {
            if (!lower_set)
            {
                lower_set = true;
                lower = sum;
            }
            else if (lower > sum)
            {
                lower = sum;
                lower_index = i;
```

```
            }
        }
        sum = 0;

    }
    qarxana_sacxobi_line[lower_index, pic_index].Visible = true;
    qarxana_sacxobi[lower_index, pic_index] = 1;

    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            qarxana_sacxobi_line[i, j] = new LineShape();
            qarxana_sacxobi_line[i, j].X1 = q_sacxobi[j].Location.X +
q_sacxobi[j].Width / 2;
            qarxana_sacxobi_line[i, j].Y1 = q_sacxobi[j].Location.Y +
q_sacxobi[j].Height / 2;
            qarxana_sacxobi_line[i, j].X2 = q_qarxana[i].Location.X +
q_qarxana[i].Width / 2;
            qarxana_sacxobi_line[i, j].Y2 = q_qarxana[i].Location.Y +
q_qarxana[i].Height / 2;
            qarxana_sacxobi_line[i, j].BorderColor = Color.Green;
            qarxana_sacxobi_line[i, j].BorderWidth = 3;
            q_container.Shapes.Add(qarxana_sacxobi_line[i, j]);
            if (qarxana_sacxobi[i, j] == 0)
            {
                qarxana_sacxobi_line[i, j].Visible = false;
            }
        }
    }
    t = new transponate1();
    qarxana_sacxobi_trans = t.transponate(qarxana_sacxobi);
    q = new multiple1();
    qarxana_sacxobi_q = q.multiple(qarxana_sacxobi_trans, qarxana_sacxobi);
```

```
                qarxana_sacxobi_lbl.Text = "არსებული Q ვექტორი ქარხანასა და საცხობს
შორის\n { " + qarxana_sacxobi_q + " }";



            }

        };
        for (int i = 0; i <5; i++)
        {

            for (int j = 0; j < 7; j++)
            {
                sacxobi_maRazia_line[i, j] = new LineShape();
                sacxobi_maRazia_line[i, j].X1 = q_sacxobi[i].Location.X + q_sacxobi[i].Width / 2;
                sacxobi_maRazia_line[i, j].Y1 = q_sacxobi[i].Location.Y + q_sacxobi[i].Height / 2;
                sacxobi_maRazia_line[i, j].X2 = q_maRazia[j].Location.X + q_maRazia[j].Width /
2;
                sacxobi_maRazia_line[i, j].Y2 = q_maRazia[j].Location.Y + q_maRazia[j].Height /
2;
                sacxobi_maRazia_line[i, j].BorderColor = Color.Green;
                sacxobi_maRazia_line[i, j].BorderWidth = 3;
                q_container.Shapes.Add(sacxobi_maRazia_line[i, j]);
                if (sacxobi_maRazia[i, j] == 0)
                {
                    sacxobi_maRazia_line[i, j].Visible = false;
                }
            }
        }
        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < 5; j++)
            {
                qarxana_sacxobi_line[i, j] = new LineShape();
                qarxana_sacxobi_line[i, j].X1 = q_sacxobi[j].Location.X + q_sacxobi[j].Width / 2;
```

```
                qarxana_sacxobi_line[i, j].Y1 = q_sacxobi[j].Location.Y + q_sacxobi[j].Height / 2;
                qarxana_sacxobi_line[i, j].X2 = q_qarxana[i].Location.X + q_qarxana[i].Width / 2;
                qarxana_sacxobi_line[i, j].Y2 = q_qarxana[i].Location.Y + q_qarxana[i].Height / 2;
                qarxana_sacxobi_line[i, j].BorderColor = Color.Green;
                qarxana_sacxobi_line[i, j].BorderWidth = 3;
                q_container.Shapes.Add(qarxana_sacxobi_line[i, j]);
                if (qarxana_sacxobi[i, j] == 0)
                {
                    qarxana_sacxobi_line[i, j].Visible = false;
                }
            }
        }

    }
    public void add_event(ref PictureBox pb , int index , string type)
    {
        if (type == "sacxobi")
        {
            pb.Click += (s, e) =>
            {
                int sum = 0;
                for (int i = 0; i < qarxana_sacxobi.GetLength(0); i++)
                {
                    if (qarxana_sacxobi[i, index] == 1)
                    {
                        sum++;
                    }
                }

                if (sum == 1)
                {
                    ganawileba.Enabled = true;
                    situacia.Text = "ვინაიდან საცხობი მხოლოდ ერთი ქარხნიდან იღებს
ფქვილს\n ის მცირე დატვირთვასაც ვერ გაუძლებს";
                }
```

```
            else if (sum == 2)
            {
                ganawileba.Enabled = true;
                situacia.Text = "საცხობს 2 ქარხანა აწვდის ფქვილს \n მაგრამ ძლიერ
დატვირთვას ვერ გაუძლებს";
            }
            else
            {
                ganawileba.Enabled = false;
                situacia.Text = "საცხობი დატვირთვას გაუძლებს";
            }
            pic_type = "sacxobi";
            pic_index = index;


        };
    }
    else if (type == "maRazia")
    {
        pb.Click += (s, e) =>
        {
            int sum = 0;
            for (int i = 0; i < sacxobi_maRazia.GetLength(0); i++)
            {
                if (sacxobi_maRazia[i, index] == 1)
                {
                    sum++;
                }
            }

            if (sum == 1)
            {
                ganawileba.Enabled = true;
                situacia.Text = "მაღაზია მცირე დატვირთვასაც ვერ გაუძლებს";
            }
```

```
            else if (sum == 2)
            { უ
                ganawileba.Enabled = true;
                situacia.Text = "მაღაზია ძლიერ დატვირთვას ვერ გაუძლებს";
            }
            else
            {
                ganawileba.Enabled = false;
                situacia.Text = "მაღაზია დატვირთვას გაუძლებს";
            }
            pic_type = "maRazia";
            pic_index = index;
        };


      }

    }

  }

}
```