ივანე ჯავახიშვილის სახელობის თბილისის
სახელმწიფო უნივერსიტეტი

გამოჭრის მრავალკრიტერიუმიანი წრფივი ამოცანები

ინფორმაციული სისტემები

ნაშრომი შესრულებულია ინფორმაციული სისტემების მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი

სტუდენტი — ასოც. პროფესორი

დევნოზაშვილი საბა — ბეჟან ღვაზერიძე

თბილისი, 2016

ანოტაცია

განხილულია ოპტიმალური გამოჭრის წრფივი ამოცანა, რომელიც ჩაწერილია მთელრიცხვა პროგრამირების ამოცანის სახით. საწყის ეტაპზე ხდება გამოჭრის ყველა ვარიანტების ფორმირება, ვარიანტების გადარჩევის გზით, რაც პრაქტიკული პრობლემის გადაჭრისას გამართლებულია. მიღებული მთელრიცხვა პროგრამირების ამოცანის ამოხსნა ხდება C# და Microsoft.Solver.Foundation.dll ბიბლიოთეკის საშუალებით.

განხილულია ამოცანის მრავალკრიტერიუმიანი ვარიანტიც. კერძოდ, შემოტანილია გამოჭრის ღირებულების მინიმუმის კრიტერიუმი. მიღებული ამოცანა ამოხსნილია კრიტერიუმთა წრფივი ნახვევის მეთოდით, სკალარული ოპტიმიზაციის ამოცანაზე მიყვანის გზით.

Annotation

The optimal linear cutting task is discusses, which is recorded in the form of integer programming problem. At the initial stage, the cutting of all the options, by selection of options, which is justified by practical problem solving. The integer programming problem is being solved by C# and Microsoft.Solver.Foundation.dll of library.

The option of a multi-task is discussed. In particular, cutting the cost of minimum criteria is brought. The criteria for the task is solved by linear method of dressings, by bringing to the scalar optimization problem.

სარჩევი

შესავალი

ნედლეულის ოპტიმალური გამოყენების(ხარჯვის) პრობლემას ძალიან დიდი მნიშვნელობა აქვს მრეწველობის მრავალი დარგისათვის. მატერიალური რესურსების ეკონომიის ეფექტურობა წარმოადგენს ერთ-ერთ უმთავრეს ფაქტორს წარმოების ეფექტურობის ასამაღლებლად და საერთოდ მთლიანად საწარმოს ფუნქციონირებისათვის. წარმოების ნარჩენები წარმოადგენენ პროდუქციის თვითღირებულების მნიშვნელოვან ნაწილს, ამიტომ ნარჩენების მინიმიზაცია მნიშვნელოვან ამოცანას წარმოადგენს.

გამოჭრის ამოცანების ქვეშ გვესმის მოდელების ფართო კლასი, რომლებიც გაერთიანებულები არიან ზოგადი ლოგიკური სტრუქტურით.

ნაშრომში განხილულია წრფივი ოპტიმალური გამოჭრის ამოცანები. მოკლედ ამოცანა ასე ყალიბდება: ნედლეული წარმოებას მიეწოდება ერთგვაროვანი ნაჭრების სახით, რომელთა სიგრძე ფიქსირებულია. მოცემულია საჭირო ნაკეთობების რაოდენობა და სიგრძე. უნდა შემუშავდეს გამოჭრის გეგმა, რომელიც მოგვცემს მინიმალური ჯამური რაოდენობის(სიგრძის) ნარჩენს. ან(და) გამოყენებული ნაჭრების მინიმიზაციას. როცა მიზანი რამდენიმეა მაშინ ვღებულობთ მრავალკრიტერიუმიან ამოცანას, რომელიც პრინციპულად განსხვავდება სკალარული ოპტიმიზაციის ამოცანისაგან. ნაშრომის ერთ-ერთ მიზანს წარმოადგენს გამოჭრის ორკრიტერიუმიანი ამოცანის ამოხსნა წრფივი ნახვევის კრიტერიუმის მეთოდით.

1 ოპტიმალური გამოჭრის ამოცანის დასმა და ცნობილი მეთოდების მიმოხილვა

ბოლო წლებში კომბინატორული ოპტიმიზაციის რთული ამოცანების ამოხსნის ალგორითმების განვითარების მიმართულებით ორი ტენდენცია გამოიკვეთა. პირველი მდგომარეობს ისეთი ალგორითმების დამუშავებაში, რომლებიც შეიცავენ მრავალ პროცედურას, რომლებიც იყენებენ სხვადასხვა ქვედა შეფასებებს, მაღალეფექტურ ევრისტიკებს, ლაგრანჟის მამარავლთა მეთოდის გამოყენებას, დაყოფისა და განშტოების სხვადასხვა მოდიფიკაციებს. ამ ალგორითმებზე დაფუძნებული პროგრამული კომპლექსები ხშირად იყენებენ გამოთვლითი პროცესის მართვის პროგრამებს, და პროგრამებს რომელიც უზრუნველყოფს ადამიანის მონაწილეობას დიალოგურ რეჟიმში.

სხვა ტენდენციაა ალგორითმების პროგრამული რეალიზაციის დანახარჯების შემცირება. საწყისი ამოცანა ფორმულირდება ისე, რომ მის ამოსახსნელად გამოიყენება მხოლოდ კომერციული პაკეტი ან გამოთვლითი ტექნიკის სტანდარტული მათემატიკური უზრუნველყოფის პაკეტები. უნდა აღინიშნოს, რომ ბოლო წლებში როგორც ერთი, ასევე მეორე მიმართულებით მიღწეულია შთამბეჭდავი შედეგები.

ხშირად სხვადასხვა პროდუქტის წარმოებისას ნედლეული შემოდის საწარმოში გარკვეული რულონების, მართკუთხა ფორმის ფურცლების ან გარკვეული სიგრძის ერთგვაროვანი მასალის (მაგ. არმატურა) სახით. შემოსული მასალა უნდა დაიჭრას სხვადასხვა კონფიგურაციის და ზომების ნაწილებად, რომლებიც ზოგ შემთხვევაში გარკვეულ ნაკეთობებს წარმოადგენენ, ზოგ შემთხვევაში კი გარკვეულ დეტალებს უფრო რთულ კონსტრუქციის ნაკეთობების დასამზადებლად.

რაციონალური გამოჭრის ამოცანები მსგავსი მათემატიკური მოდელებით აღიწერება. მათ შორის არსებითი სხვაობა ძირითადად ორი ფაქტორითაა განპირობებული:

1) გამოჭრის შედეგად მიღებული ნაკეთობის კონფიგურაცია
2) გამოსაშვები პროდუქციის მოცულობა

პირველი ფაქტორით განსაზღვრული გამოჭრის ამოცანები ორ ქვეკლასად იყოფა - **ფიგურული გამოჭრის ამოცანები** და **არაფიგურული გამოჭრის ამოცანები**. ფიგურული გამოჭრის ამოცანებში მოითხოვება სხვადასხვა კონფიგურაციის ნაკეთობების მიღება, ასეთი ამოცანები შედარებით რთულ კატეგორია მიეკუთვნება, ჩვენ ძირითადად განვიხილავთ მეორე ტიპის წრფივი გამოჭრის ამოცანებს.

ჩვენს შემთხვევაში ოპტიმალური გამოჭრის ამოცანა გულისხმობს რაიმე ტიპის (სახის) მასალისაგან ჩვენთვის სასურველი ზომის და ფორმის დეტალების მოჭრას ან გამოჭრას ისე, რომ რაიმე კონკრეტული კრიტერიუმით იყოს ოპტიმალური. ეს ამოცანა შეიძლება განვიხილოთ სხვადასხვა განზომილებებისათვის.

ერთი განზომილებისათვის შეგვიძლია წარმოვიდგინოთ, რომ გვაქვს რაიმე ფიქსირებული სიგრძის მონაკვეთები და გვინდა რომ მათგან გამოვჭრათ სხვადასხვა ზომისა და რაოდენობის მონაკვეთები. მასალის მინიმალურად გახარჯვისათვის კი შეგვიძლია სხვადასხვა პირობები წარმოვადგინოთ. მაგ. ნედლეულის გამოყენება მინიმალური რაოდენობით (ცალობით), ან ნედლეულის გამოყენება ისე, რომ ნარჩენის ჯამური სიგრძე იყოს მინიმალური (მეტრობით).

ორგანზომილებიანი ამოცანისათვის კი წარმოიშვება შემდეგი პრობლემა; ამ შემთხვევაში ჩვენი მასალა არა მონაკვეთებად არამედ ფიგურებად უნდა წარმოვიდგინოთ. პრაქტიკაში უმეტესად მართკუთხედის ფორმები გვხვდება მაგ. მინის დაჭრისათვის, კარადების აწყობისათვის... მაგრამ შეიძლება სხვა ფორმებიც შეგვხვდეს და ამ ფორმებიდან გამოიჭრას სხვადასხვა მომცრო ზომის იგივე ან განსხვავებული ფორმები. ამოცანის ოპტიმალური ამონახსნის ვარიანტები კი შეიძლება იგივენაირად განვიხილოთ როგორც ერთგანზომილებიანის შემთხვევაში.

სამგანზომილებიანი ამოცანაც შეიძლება მსგავსად წარმოვიდგინოთ სივრცეში, თუმცა მისი პრაქტიკულ ამოცანებში გამოყენება უფრო ნაკლებად გვხვდება დაბალგანზომილებებთან შედარებით.

1.1 ოპტიმალური გამოჭრის ამოცანის ფორმულირება მთელრიცხვა პროგრამირების ამოცანის სახით

ჭრფივი გამოჭრის ერთგანზომილებიანი კლასიკური ამოცანა შემდეგში მდგომარეობს: L სიგრძის ერთგანზომილებიანი მასალა უნდა დავანაწილოთ l_1, l_2, ..., l_m შედარებით მცირე ზომის ნაკეთობებად შესაბამისად b_1, b_2, ..., b_m რაოდენობით. მიზანი შეიძლება იყოს გამოყენებული მასალის რაოდენობის მინიმიზაცია. დიკხოის კლასიფიკაციით [1] ამ ამოცანას მოკლედ P=(L, n, l, b) სახით აღნიშნავენ, სადაც L, n მოცემული რიცხვებია. N გამოჭრის ყველა ვარიანტის რაოდენობა, l=(l_1, l_2, ..., l_m) და b=(b_1, b_2, ..., b_m) კი მოცემული ვექტორები. ცნობილია, რომ ამოცანა N-რთული ამოცანების კლასს ეკუთვნის[5].

გამოჭრის ყოველი დასაშვები გეგმა შეიძლება წარმოვადგინოთ A=(a_{ij}), i=1,…,m, j=1,…,n, მატრიცის სახით, სადაც a_{ij} რიცხვები არაუარყოფითი მთელი რიცხვებია და აღნიშნავს j-ურ გამოჭრის ვარიანტში i-ური სახეობის ნაკეთობის რაოდენობას, ცხადია, რომ

$$\sum_{i=1}^{m} \boldsymbol{l_i * a_{ij} \leq L}.$$

თუ x_j არის ჯ-ური გამოჭრის ვარიანტების რაოდენობა, მაშინ ამოცანა შეიძლება ჩაიწეროს წრფივი მთელრიცხვა პროგრამირების ამოცანის სახით:

$$\boldsymbol{f = \sum_{i=0}^{M} x_i \rightarrow min},$$

$$\boldsymbol{\sum_{j=1}^{M} a_{ij} * x_j \geq b_i, \quad i = 1, \dots, m} \text{ და } \boldsymbol{x_i \in Z_0^+}.$$

შევნიშნოთ, რომ ჩვენი ამოცანის ამონახსნი უნდა იყოს მთელრიცხვა, რაც გარკვეულწილად ართულებს პრობლემის გადაწყვეტას.

პრაქტიკაში ხშირად გვხვდება დაჭრის ამოცანები, როცა მიზნის ფუნქცია გამოხატავს არა რაოდენობის მინიმიზაციას არამედ ნარჩენი ნედლეულის მინიმიზაციას, გამოჭრის ორკრიტერიუმიან ამოცანას შემდგომში განვიხილავთ.

1.2 ამოხსნის ალგორითმების მიმოხილვა

როგორც აღვნიშნეთ გამოჭრის ზევრნაირი ვარიანტი არსებობს, რომლებიც უნდა აკმაყოფილებდნენ პირობას:

$$\sum_{i=1}^{m} l_i * a_{ij} \leq L.$$

თუ L და m-ის მნიშვნელობები საკმაოდ დიდია გამოჭრის ვარიანტების ხელით გადათვლა შეუძლებელი ხდება, ამისათვის საჭირო იქნება ავტომატური გამოთვლითი პროგრამის შექმნა, რომელიც დაითვლის გამოჭრის ყველა შესაძლო ვარიანტს. მოვიყვანოთ ფსევდოკოდი და კონკრეტული მაგალითი, სადაც უფრო თვალნათლივ გამოჩნდება გენერაციის პრინციპები.

ქვემოთ მოყვანილი Generate Variants (GV) პროცედურა მუშაობს შემდეგნაირად: საწყის ეტაპზე ხდება გამოსაყენებელი ცვლადების ინიციალიზაცია, გამოსაჭრელი ზომების დალაგება კლებადობით, შემდგომ კი პირველი ვარიანტის გენერირება უდიდესი გამოსაჭრელი ზომების მეშვეობით (ერთი ზომა განსხვავებული იქნება თუ L - $l_1 * x \geq l_{max}$ სადაც l_1 უდიდესი გამოსაჭრელი ზომის მნიშვნელობის მქონეა, x ამ ზომის რაოდენობა, რამდენიც შეიძლება მოთავსდეს L ზომის მონაკვეთში, ხოლო $\boldsymbol{l_{max}}$ ყველაზე მინიმალური გამოსაჭრელი ზომის მქონე მნიშვნელობა.

PROGRAM GV(**arraySizes[]**)

```
REVERSE sort arraySizes[]
position = 1;
minElement = arraySizes[vectorSizes.SIZE]
size = 0

FOR j = 1 to arraySizes.SIZE
     WHILE size + arraySizes[j] ≤ maxSize
          arrayRoad ADD arraySizes[j]
          size = size + arraySizes[j]
     ENDWHILE
     CONTINUE
END FOR
arrayRoads[][] ADD arrayRoad[];
```

შემდგომ ეტაპზე კი ხდება პირველი ვარიანტის ცვლილებით დანარჩენი ყველა ვარიანტების გენერირება, რომლისთვისაც ვიყენებთ სხვადასხვა დამხმარე ფუნქციებს, ფუნქციის **isAllEqualMin()** მეშვეობით ვხვდებით მოძებნილ ვარიანტში ყველა ზომა მინიმალურია თუ არა. თუ ყველა ზომა მინიმალურია ეს ნიშნავს რომ მოვძებნეთ ბოლო ვარიანტი, რასაც გამოვიყენებთ ციკლის გაჩერების პირობაში. **findindex()** ფუნქცია გვიბრუნებს ბოლო დაგენერირებული ვარიანტის იმ ინდექსს რომელზე მდგომი ზომაც უნდა შეიცვალოს ხოლო **findindexElement()** პოულობს შესაბამისი ზომის ელემენტის ინდექსს ზომების ვექტორში.

```
WHILE !isAllEqualMin()
     changeindex = findindex()
     position = findindexElement()
     FOR i = changeindex TO vectorRoad.SIZE
          size = size - vectorRoad[i];
     END FOR
     arrayRoad[changeindex].ERASE
     FOR j = position + 1 to arraySizes.SIZE
          WHILE size + arraySizes[j] ≤ maxSize
               arrayRoad[changeindex] INSERT arraySizes[j]
               size = size + arraySizes[j]
```

```
                changeindex = changeindex + 1
            ENDWHILE
            CONTINUE
        END FOR
        arrayRoads[][] ADD arrayRoad[];
    END WHILE
END PROGRAM
```

arrayRoads[][] მასივში გვექზება ჩვენი დაგენერირებული ვარიანტები.

1.3 გამოჭრის ვარიანტების გენერირება კონკრეტულ მაგალითზე

წარმოვადგინოთ კონკრეტული ამოცანა და მოცემული ფსევდოკოდის საშუალებით მოვახდინოთ გამოჭრის ვარიანტების გენერირება. ზემოთ მოყვანილი აღნიშვნებიდან შემოვიტანოთ მნიშვნელობები:

L = 22მ, m = 3,

l_1 = 5მ, $b_1 = 150$,

l_2 = 7მ, $b_2 = 200$,

l_3 = 9მ, $b_1 = 300$.

სიგრძის ერთეული არის პირობითი, რადგან პროგრამა არ ითვალისწინებს ერთეულს. მოცემული მაგალითიდან გვექნება გამოჭრის 6 ვარიანტი. ვარიანტები წარმოვადგინოთ ცხრილის სახით:

	1	2	3	4	5	6	7
l_1	0	1	2	0	1	3	4
l_2	0	1	0	3	2	1	0
l_3	2	1	1	0	0	0	0

ამ მონაცემებისათვის სწორედ ეს გამოჭრის ვარიანტები არსებობს, მნიშვნელობები შეესაბამება a_{ij}-ს, საიდანაც მარტივად შეგვიძლია წარმოვადგინოთ ამ ამოცანისათვის მათემატიკური მოდელი. მიზნის ფუნქციას მინიმალური რაოდენობა L სიგრძის მონაკვეთების გამოსაყენებლად ექნება შემდეგი სახე:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \rightarrow \min$$

ხოლო შეზღუდვებს დაგენერირებული ვარიანტების და დასაჭრელი რაოდენობის მიხედვით ექნება შემდეგი სახე:

$$0 * x_1 + 1 * x_2 + 2 * x_3 + 0 * x_4 + 1 * x_5 + 3 * x_6 + 4 * x_7 \geq 150$$

$$0 * x_1 + 1 * x_2 + 0 * x_3 + 3 * x_4 + 2 * x_5 + 1 * x_6 + 0 * x_7 \geq 200$$

$$2 * x_1 + 1 * x_2 + 1 * x_3 + 0 * x_4 + 0 * x_5 + 0 * x_6 + 0 * x_7 \geq 300$$

$$x_i \in Z_0^+, \text{სადაც } i = \{1,2 \dots m\}$$

ამ მოდელს პროგრამა ავტომატურად აგენერირებს სადაც შეზღუდვების კოეფიციენტები ცხრილში წარმოდგენილი მნიშვნელობების შესაბამისი კოეფიციენტებია, რომელიც იწერება lp გაფართოების ფაილში LPSolve IDE პროგრამის სინტაქსით. პროგრამა გვაძლევს ოპტიმალურ ამონახსნს.

2 გამოჭრის მრავალკრიტერიუმიანი ამოცანის დასმა

ადამიანის მოღვაწეობის თითქმის ყველა სფეროში წარმოიშობა მიზნის მიღწევის არსებული საშუალებებიდან საუკეთესოს ანუ ოპტიმალურის არჩევის პრობლემა. მეცნიერებისა და ტექნიკის სხვადასხვა დარგების სპეციალისტები ახალი მოწყობილობების დაგეგმარებისას ცდილობენ მიიღონ საუკეთესო საინჟინრო, კონსტრუქციული და საპროექტო გადაწყვეტილებები. საბანკო სფეროს წარმომადგენლები ოპტიმალურად ირჩევენ ობიექტებს ინვესტირებისათვის, ფირმებისა და საწარმოების ეკონომისტები კი ირჩევენ ოპტიმალურ ეკონომიკურ პოლიტიკას და ა.შ. შევეცადოთ გამოვყოთ ისეთი ზოგადი ელემენტები, რომლებიც დამახასიათებელია ასეთი ამოცანებისათვის.

პირველ რიგში უნდა განისაზღვროს და აღწერილ იქნას სიმრავლე, რომელშიც უნდა განხორციელდეს არჩევა. აღვნიშნოთ ის X-ით და ვუწოდოთ **დასაშვებ ამონახსნთა სიმრავლე.** ხშირად ამონახსნის მაგივრად იხმარება ტერმინები: **ალტერნატივა, ვარიანტი, გეგმა, სტრატეგია.** ეს სიმრავლე შეიძლება იყოს როგორც სასრული, ასევე უსასრულო. თვითონ ამონახსნის ბუნებას გადაწყვეტილების მიღების თეორიისათვის არაფითარი მნიშვნელობა არ აქვს. ეს შეიძლება იყოს საპროექტო გადაწყვეტილება, პოლიტიკური ან ეკონომიკური სტრატეგია, პერსპექტიული გეგმა, განვითარების პროგნოზი და სხვა.

ამორჩევის ამოცანის სირთულე მრავალი კრიტერიუმის შემთხვევაში მდგომარეობს იმაში, რომ შეუძლებელია აპრიორული განსაზღვრება იმისა, თუ რას ვუწოდოთ ოპტიმალური ამონახსნი.

ამონახსნის არჩევა გულისხმობს დასაშვებ ამონახსნთა შორის გარკვეული აზრით საუკეთესოს არჩევას, თუმცა ეს შეიძლება არ იყოს ერთი ამონახსნი და იყოს დასაშვებ ამონახსნთა რაღაც ქვესიმრავლე $C(X)$, რომელსაც ვუწოდოთ **ამორჩეული ამონახსნების სიმრავლე.**

ამორჩევის პროცესი შეუძლებელია განხორციელდეს იმ პირის გარეშე, რომელიც ახორციელებს ამ ამორჩევას. პირს (ან მთელ კოლექტივს) რომელიც ახორციელებს ამორჩევასა და პასუხისმგებელია მის შედეგებზე უწოდებენ **გადაწყვეტილების მიმღებ პირს**(მოკლედ გმპ).

ჩვეულებრივ მიღებულია, რომ უკვე ამორჩეული (და ამიტომ შეიძლება ვთქვათ მისაღები, ხელსაყრელი, საუკეთესო) ალტერნატივა არის ისეთი დასაშვები ალტერნატივა, რომელიც ყველაზე უფრო შეესაბამება გმპ-ს ინტერესებს და მიზნებს. გმპ-ს სურვილი მიაღწიოს გარკვეულ მიზანს მათემატიკურ ტერმინებში გარკვეული რიცხვითი ფუნქციების X სიმრავლეზე მაქსიმიზაციის(ან მინიმიზაციის) სახით

ჩაიწერება. მაგრამ რთულ სიტუაციებში საქმე გვაქვს არა ერთ, არამედ რამდენიმე ასეთ ფუნქციასთან. ამ დროს ამბობენ, რომ გვაქვს ე.წ. ვექტორული კრიტერიუმი, რომელიც შედგება რამდენიმე სკალარული ფუნქციისაგან:

$$f_1, f_2, \dots f_m \ \text{სადაც} \ m \ \geq \ 2, \qquad f_i: X \ \rightarrow \ R, \qquad i \ = \ 1, 2 \dots . m.$$

ცხადია, რომ ვექტორული კრიტერიუმი f = (f_1 , f_2 , . . . f_m) ღებულობს მნიშვნელობებს R^m- m განზომილებიან ვექტორთა სივრცეში. ამ სივრცეს უწოდებენ **კრიტერიალურ სივრცეს ან შეფასებათა სივრცეს**, ხოლო ვექტორული f კრიტერიუმის ყოველ $f\,(x)\ =\ (f_1\,(x), f_2\,(x), \dots, f_m\,(x)) \in R^m$ მნიშვნელობას განსაზღვრული $x \in X$ ამონახსნისათვის კი x ამონახსნის ვექტორულ შეფასებას.

$$Y \ = \ f\,(X) \ = \{y \ \in \ R^{\ m} |\ y \ = \ f\,(x), \text{სადაც}\ x \ \in \ X\}$$

სიმრავლეს ეწოდება **შესაძლო შეფასებათა სიმრავლე**. ასევე შემოვიტანოთ ამორჩეული ვექტორების სიმრავლე:

$$C(Y\,) \ = \ f\,(C(X)) \ = \{y \ \in \ Y \,|\, y \ = \ f\,(x), \text{სადაც}\ x \ \in \ C(X)\}$$

2.1 გამოჭრის ორკრიტერიუმიანი ამოცანა

ჩვენს შემთხვევაში $f_1 = \sum_i^M x_i$ მიზნის ფუნქციასთან ერთად პრაქტიკული სიტუაციებიდან გამომდინარე ხშირად საჭირო ხდება $f_2 = \sum_{i=1}^M d_i x_i \rightarrow min$ სადაც $d_i = L$ - $\sum_{i=1}^m l_i \ * \ a_{ij}$ სიდიდის მინიმიზაციაც რაც ნიშნავს ნარჩენი მასალის ჯამური სიგრძის მინიმიზაციას, ამიტომ გამოჭრის ორკრიტერიუმიანი ამოცანა მიიღებს შემდეგ სახეს:

$$f_1 = \sum_{i=1}^{M} x_i \ \rightarrow min.$$

$$f_2 = \sum_{i=1}^{M} d_i x_i \rightarrow min.$$

შეზღუდვები გვექნება შემდეგი სახით:

$$\sum_{j=1}^{M} a_{ij} * \ x_j \ \geq \ b_i, \ \ i = 1, \dots, m,$$

$$x_i \ \in Z_0^+.$$

როგორც წესი X და Y სიმრავლეებს შორის შეიძლება დავამყაროთ ურთიერთცალსახა შესაბამისობა. ამიტომ მათემატიკურად ამორჩევა X სიმრავლეში ტოლფასია ამორჩევისა Y სიმრავლეში, და ყველა განსაზღვრება თუ შედეგი შეიძლება ფორმულირებულ იქნას, როგორც დასაშვებ ამონახსნთა ტერმინებში, ასევე ვექტორების (კრიტერიალური სივრცის) ტერმინებში.

ამორჩევის ამოცანას, რომელშიც დაფიქსირებულია დასაშვებ ამონახსნთა X სიმრავლე და f ვექტორული კრიტერიუმი უწოდებენ **მრავალკრიტერიუმიან ამოცანას ან ოპტიმიზაციის მრავალკრიტერიუმიან ამოცანას.** უნდა აღინიშნოს, რომ ვექტორული კრიტერიუმით წარმოდგენილი "მიზნები" ხშირად წინააღმდეგობრივია და მათი მიღწევა ერთდროულად შეუძლებელია, ამიტომ ვექტორული კრიტერიუმის გარდა საჭიროა დამატებითი ინფორმაცია კომპრომისის განსახორციელებლად, ანუ სხვანაირად რომ ვთქვათ, საჭიროა დამატებითი ინფორმაცია გმპ-ის პრიორიტეტების შესახებ. საჭიროა მრავალკრიტერიუმიანი ამოცანის დასმაში შევიტანოთ კიდევ ერთი ელემენტი, რომელიც აღწერს ამ პრიორიტეტებს (უპირატესობას).

განვიხილოთ ორი დასაშვები x' და x'' ამონახსნი. ვთქვათ, გმპ ირჩევს (უპირატესობას ანიჭებს) პირველს. ამ შემთხვევაში წერენ

$$x' \succ_X x'',$$

შევნიშნოთ, რომ ყოველი დასაშვები x' და x'' ამონახსნისათვის შეიძლება არც $x' \succ_X x''$ თანაფარდობა და არც $x' \prec_X x''$ თანაფარდობა არ შესრულდეს. ეს სიტუაცია რეალურად ასახავს საქმის არსს.

შეგვიძლია ვთქვათ, რომ

$$f(x') \succ_Y f(x'') \Longleftrightarrow x' \prec_X x'',\ x',\ x'' \in X,$$

ანუ $y' = f(x')$ ვექტორი უპირატესია $y'' = f(x'')$ ვექტორთან მაშინ და მხოლოდ მაშინ, როცა x' უპირატესია x'' -თან.

2.2 ოპტიმალური ამონახსნი პარეტოს აზრით (პარეტოს ოპტიმალური ამონახსნი)

ვთქვათ, x' და x'' ორი ნებისმიერი დასაშვები ამონახსნია, მაშინ შესაძლებელია შემდეგი სამი შემთხვევიდან მხოლოდ ერთი:

1) სამართლიანია თანაფარდობა $x' \succ_X x''$,

2) სამართლიანია თანაფარდობა $x'' \succ_X x'$,

3) არ სრულდება არც 1) და არც 2).

1) შემთხვევაში ამბობენ, რომ x' **დომინირებს** x'' -ზე,

2) შემთხვევაში x'' **დომინირებს** x' -ზე.

3) შემთხვევაში კი x' და x'' **არ არიან სადარნი.**

განსაზღვრება: $x^* \in$ X ამონახსნს ეწოდება **პარეტოს აზრით ოპტიმალური (პარეტოოპტიმალური),** თუ არ მოიძებნა ისეთი $x \in X$ სამართლიანია უტოლობა f (x) ≤ f (x^*). დასაშვები ამონახსენი, რომ პარეტოს აზრით ოპტიმალური ყველა ამონახსნი ქმნის **პარეტოს სიმრალეს,** რომელსაც P_f(X)-ით აღნიშნავენ.

ვთქვათ, x^* პარეტო-ოპტიმალური ამონახსნია, ხოლო f (x^*)-მისი შესაბამისი პარეტო-ოპტიმალური ვექტორი, განმარტების ამონახსნისათვის, რომელიც განსხვავებულია

x^* -სგან, შესრულებულია უტოლობა f_i (x) < f_i (x^*), მაშინ უნდა მოიძებნოს ერთი მაინც ისეთი i, რომ f_i (x^*) < f_i (x). სხვანაირად: **პარეტო-ოპტიმალური ამონახსენი-ისეთი დასაშვები ამონახსნია, რომელიც არ შეიძლება გაუმჯობესდეს არც ერთი კრიტერიუმის მიმართ, სხვა კრიტერიუმის მიმართ გაუარესების გარეშე.**

ხშირად პარეტო-ოპტიმალურ ამონახსნებს ეფექტურ ამონახსნებსაც უწოდებენ.

პარეტო-ოპტიმალური x^* ამონახსნისათვის f(x^*) ვექტორს უწოდებენ **პარეტო ოპტიმალურ** ვექტორს. ასეთი ვექტორების სიმრავლისათვის იყენებენ აღნიშვნას:

$$P\,(Y\,) \;=\; f\,(P_f(X)) \;=\; \{f\,(x^*) \;\in\; Y \,|\, x^* \;\in\; P_f(x)\}\; სადაც\, Y \;=\; f\,(X).$$

ადვილი შესამჩნევია, რომ პარეტო-ოპტიმალური ვექტორების სიმრავლე შეიძლება ასეც განიმარტოს:

$$P\,(Y\,) \;=\; \{y^{\,*} \;\in\; Y \,|\; არ\, არსებობს\, ისეთი\, y \;\in\; Y\,,\, რომ\, y \;\leq\; y^{\,*}\,\}.$$

2.3 ზოგიერთი ცნობილი შედეგი

მრავალკრიტერიალური ოპტიმიზაციის დარგში პირველი შედეგები უკავშირდება პარეტოს სიმრავლის ან მისი ქვესიმრავლის საპოვნელად წრფივი ნახვევის კრიტერიუმის გამოყენების საკითხის გამოკვლევას.

კარგადაა ცნობილი [1], რომ ნებისმიერი

$$\lambda \in \Lambda = \{\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m) \mid \sum_{i=1}^{m} \lambda_i = 1, \lambda_i > 0, i = 1, 2 \dots, m\}$$

$$\lambda_i = 1, \lambda_i > 0, \quad სადაც\; i = 1, 2 \dots, m$$

ვექტორისათვის

$$\Phi(\lambda, x) = \sum_{i=1}^{m} \lambda_i f_i(x)$$

კრიტერიუმების წრფივი ნახვევის X სიმრავლეზე მამინიზირებელი x^* ელემენტი არის ეფექტური წერტილი. შევნიშნოთ, რომ ეს ფაქტი სამართლიანია ნებისმიერი ბუნების (როგორც წარმოადგენს უწყვეტი, ასევე დისკრეტული) X სიმრავლისათვის, და ის წარმოადგენს საფუძველს უმრავლესობა ალგორითმებისათვის, რომლებიც გამოიყენება დღესდღეობით მათემატიკური პროგრამირების მრავალკრიტერიუმიანი ამოცანებისათვის. მათემატიკური პროგრამირების ზოგიერთი ამოცანისათვის სამართლიანია შებრუნებული ფაქტიც [1], კერძოდ, თუ X ამოზნექილი სიმრავლეა, ხოლო f (x) ვექტორული კრიტერიუმის ფუნქციები ჩაზნექილი ფუნქციებია, მაშინ $x^* \in P_f(X)$ მაშინ და მხოლოდ მაშინ, როცა არსებობს λ ∈ Λ, რომლისთვისაც $\Phi(\lambda, x^*) = min\{\Phi(\lambda, x), x \in X\}$, საიდანაც გამომდინარეობს, რომ ამოზნექილი და კერძოდ წრფივი მრავალკრიტერიალური ამოცანებისათვის წრფივი ნახვევის კრიტერიუმის გამოყენებით თეორიულად შესაძლებელია X სიმრავლიდან P_f (X) სიმრავლის გამოყოფა. ამ შემთხვევაში ამბობენ, რომ ამოცანა ამოხსნადია წრფივი ნახვევის

კრიტერიუმის გამოყენებით. [4] შრომაში ნაჩვენებია, რომ არსებობს დისკრეტული მრავალკრიტერიალური ამოცანების ისეთი ეფექტური ამონახსნები, რომელთა პოვნაც წრფივი ნახვევის კრიტერიუმით შეუძლებელია. ამდენად, შესაძლებელია საუბარი ასეთი ამოცანების წრფივი ნახვევის კრიტერიუმით ამოუხსნადობაზე.

2.4 მრავალკრიტერიალური ამოცანების ამოხსნის ძირითადი სქემების შესახებ.

არსებობს ძირითადად ორი ტიპის მიდგომა მრავალკრიტერიალური ოპტიმიზაციის ამოცანების ამოხსნისა : 1) კომპრომისების დაშვების მეთოდები, 2) ეფექტურ ამოხსნათა სრული სიმრავლის გულისხმობს გარკვეული აზრით აგების მეთოდები. პრიველი ტიპის მეთოდები ფორმულირებადი ერთკრიტერიუმიანი ამოცანის ან ამოცანათა სასრული მიმდევრობის ამოხსნას. ასეთ მეთოდებს მიეკუთვნება:

კრიტერიუმის ადიტიური ნახვევის მეთოდი, "იდეალური წერტილის" მეთოდი. ასეთი მეთოდების ნაკლია: ყოველთვის არ არის ნათელი კომპრომისის რომელი სქემა არის მიზანშეწონილი, ყოველთვის არ მიიღება პარეტოს ამონახსნი.

უფრო დაწვრილებით შევჩერდეთ ე.წ. "იდეალური წერტილის" მეთოდზე, რომელიც წრფივი პროგრამირების ამოცანებისათვის პირველად გამოიყენა მ.სალუქვაძემ [3]. ვექტორული შეფასებების Y სივრცეში განვიხილოთ ე.წ. იდეალური წერტილი f^*= f_1^*, $f_2^*, \dots, f_m^*$), სადაც $f_i^* = min_{x\in X} f_i(x), i = 1, 2, \dots m$. რომ არსებობდეს ალტერნატივა x^* $\in$ X, ისეთი, რომ $f_i(x^*) = f_i^*$, i=1, 2, . . . m მაშინ x^* იქნებოდა საუკეთესო ალტერნატივა. მაგრამ როგორც წესი, ასე არ ხდება. ამიტომ ბუნებრივია საუკეთესო ამონახსნად გამოვაცხადოთ წერტილი, რომლის ვექტორული შეფასება ყველაზე ახლოსაა იდეალურ f ∗ წერტილთან. ინტუიციურად ასეთი მიდგომა სავსებით მისაღებია, მარამ მას აქვს სერიოზული ნაკლი. როცა ჩვენ ვსაუბრობთ "იდეალურთან ახლოს" მდგომ წერტილზე, ვგულისხმობთ, რომ ვექტორული შეფასების სივრცეში გვაქვს მეტრიკა, რომელიც შეიძლება მრავალნაირად შემოვიღოთ. ცხადია, რომ სხვადასხვა მეტრიკისათვის საუკეთესო იქნება სხვადასხვა ალტერნატივა. შეიძლება ვაჩვენოთ, რომ ყოველი პარეტო ოპტიმალური წერტილისათვის მოიძებნება მეტრიკა, რომელშიც ამ წერტილის ვექტორული შეფასება იქნება "იდეალურ წერტილთან" ყველაზე ახლოს. ამრიგად, ალტერნატივის არჩევის ამოცანიდან მივედით მეტრიკის არჩევის ამოცანამდე, რაც არანაკლებ რთული პრობლემაა.

მეორე ტიპის მეთოდებში, როგორც ზემოთ აღვნიშნეთ ვექტორულ წერტილთა სრული სიმრავლის აგებაა გამიზნული. მიდგომა უნივერსალურია, რადგან გმპ-ს აქვს საშუალება შეარჩიოს მისთვის სასურველი (პრაქტიკულ სიტუაციაზე გარკვეული აზრით მორგებული) ამონახსნი. მაგრამ არსებობს ნაკლიც: ალტერნატივათა სრული სიმრავლის სიმძლავრე შეიძლება იყოს ძალიან დიდი და მისი მიმოხილვა გახდეს პრაქტიკულად შეუძლებელი(ამ საკითხს კონკრეტული ამოცანისათვის მე-2ე პარაგრაფშიც შევეხებით) და ასეთი სიმრავლის აგება გამოთვლითი თვალსაზრისითაც ძალიან რთულია.

პარაგრაფის ბოლოს შევეხოთ კრიტერიუმთა სკალირების პრობლემას. თითოეული $f_i(x)$ კრიტერიუმი წარმოადგენს x ალტერნატივის გარკვეულ ლოკალურ თვისებას (მახასიათებელს), მაგალითად: წონას, ღირებულებას, საიმედოობას, სწრაფმოქმედებას და ა.შ. ეს კრიტერიუმები ერთი და იგივე ერთეულებში არ იზომება, ამიტომ ისინი დაყავთ ე.წ. უგანზომილო სიდიდეებზე. მაგალითად, f_i(x) ფუნქციის ნაცვლად გამოვიყენოთ λ_i(x) = $\frac{f_i(x)}{f_i^*}$ ფუნქცია, სადაც f_i^* = max $f_i(x)$, ცხადია, რომ $0 \le \lambda_i\,(x) \le 1$.

2.5 რიცხვითი შედეგები

ჩვენს მიერ ზემოთ განხილული f_1 და f_2 მიზნის ფუნქციების ამოხსნის შედეგად მიღებული შედეგები არის შემდეგი: f_1 = 242 რაც გულისხმობს რომ გვჭირდება მინიმუმ 242 ცალი L=22მ სიგრძის არმატურა, რომ გამოვჭრათ l_1 = 5მ სიგრძის $b_1 =$ 150 ცალი, l_2 = 7მ სიგრძის $b_2 = 200$ ცალი, l_3 = 9მ სიგრძის $b_1 = 300$ ცალი არმატურები. ამისათვის უნდა გამოვიყენოთ დაჭრის რამდენიმე ვარიანტი, კერძოდ: პირველი ვარიანტი 76-ჯერ, მეორე ვარიანტი 148-ჯერ, მეოთხე ვარიანტი 16-ჯერ, მეხუთე ვარიანტი 2-ჯერ. აღნიშნული ვარიანტების შემთხვევაში ჩვენ გვრჩება ჯამური ნარჩენი არმატურების სიგრძე 474 მეტრი.

f_2 მიზნის ფუნქციის ამოხსნის შემთხვევაში, როცა ჩვენი მიზანია რომ ნარჩენის ჯამური სიგრძე იყოს მინიმალური გვრჩება 300 მეტრი არმატურა ამ შემთხვევაში მხოლოდ გამოჭრის მეორე ვარიანტს ვიყენებთ 300-ჯერ რაც გვაძლევს იმის საშუალებას რომ მხოლოდ 300 ცალი 22მ სიგრძის არმატურა გავხარჯოთ.

როგორც ვნახეთ საკმაოდ განსხვავებული შედეგები მივიღეთ იქიდან გამომდინარე თუ რა იყო ჩვენი მიზანი. ამიტომ მიღებული ორკრიტერიუმიანი ამოცანის ამოსახსნელად გამოვიყენეთ წრფივი ნახვევის მეთოდი, კერძოდ განვიხილეთ სკალარული მიზნის ფუნქცია:

$\alpha f_1 + (1 - \alpha) f_2$, სადაც $\alpha \in [0; 1]$, კერძოდ α=0.1, 0.2,...,0.9 მნიშვნელობებისათვის ამოხსნილია სკალარული ოპტიმიზაციის ამოცანა და მიღებული პარეტო ამონახსნები მოცემულია ცხრილში:

f \ α	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
f_1	300	300	300	300	300	300	400	400	464
f_2	300	300	300	300	300	300	250	250	242

მიღებული შედეგები მიღებულია წინა თავში განხილული სკალირების პრობლემის გადაჭრით. ჩვენს შემთხვევაში სკალირების პრობლემა გადაჭრილ იქნა მიზნის ფუნქციის კოეფიციენტების შეფარდებით მიღებულ ოპტიმალურ შედეგებთან რის შემდეგაც მივიღეთ უგანზომილებო ახალი ტიპის ცხრა მიზნის ფუნქცია $\alpha f_1 + (1 - \alpha) f_2$, α-ს ცვლილებიდან გამომდინარე.

დასკვნა

ნაშრომში განხილულია ამოცანა, რომლის ამოხსნის მიზანს წარმოადგენს გარკვეული ტიპის ნაკეთობების ისეთი წარმოების გეგმის შედგენა, რომელიც უზრუნველყოფს ასეთი ნაკეთობების აუცილებელი რაოდენობის წარმოებას, მინიმალური დანახარჯებით (მაგალითად მინიმალური ნარჩენებით). განხილულია ამ ამოცანის მთელრიცხვა მათემატიკური მოდელი და მისი განზოგადება მრავალკრიტერიუმიანი ამოცანის სახით. ეს უკანასკნელი ამოხსნილია კრიტერიუმთა წრფივი ნახვევის მეთოდით. მცირე განზომილებიანი პრაქტიკული ამოცანებისათვის ჩატარებულია რიცხვითი ექსპერიმენტი. გამოჭრის ვარიანტების გენერირებისათვის გამოყენებულია C++ პროგრამირების ენა ხოლო შედეგების დასათვლელად C# და მისი ბიბლიოთეკა Microsoft.Solver.Foundation.dll.

ლიტერატურა

[1] Dyckhoff H. A typology of cutting and packing problems. F.R. Germany. 1991

[2] Ehrgott M. Multicriteria Optimization, Academic Press, Springer, 2nd Edition, New York, 2005.

[3] Salukvadze M. Vector-Valued Optimization Problems in Control Theory, Academic Press, New York, 2005.

[4] Melamed I. I. Multicriteria combinatorial optimization. Theory and algorithms, Erro VI, Brussels, 1993. p.72

[5] Гэри М. Джонсон Д. Вычислительные машины труднорешаемые задачи.-М.:Мир.1982.