ივანე ჯავახიშვილის სახელობის თბილისის

სახელმწიფო უნივერსიტეტი

ინფორმაციული სისტემები

მიახლოებითი ალგორითმები კომივოიაჟერის ამოცანაში

სტუდენტი:

გიორგი შავაძე

ხელმძღვანელი:

ასოცირებული პროფესორი,
ბეჟან ღვაბერიძე

თბილისი, 2016

ანოტაცია

ნაშრომში განხილულია მიახლოებითი ამონახსნის აგების საკითხები კომივოიაჟერის ამოცანისთვის. კერძოდ, ცნობილი სატესტო მაგალითებისათვის რეალიზებულია "წადი უახლოეს ქალაქში" , "ამოირჩიე უმცირესი რკალი" და „ლოკალური ძიების 2-opt ამონახსნის" ტიპის ევრისტიკების ზოგიერთი განზოგადება, მოყვანილია რიცხვითი ექსპერიმენტების შედეგების ანალიზი.

abstract

Constructing issues of approximate solution for TSP tasks are discussed in the work. Particularly, “Nearest Neighbor algorithm”, “choose the least arch” and some generalization of local search “2-opt solution” type heuristics are actualized for famous test examples; Analyze of results of number experiments are provided.

სარჩევი:

შესავალი

კომივოიაჟერის ამოცანას განსაკუთრებული ადგილი უჭირავს კომბინატორულ ოპტიმიზაციასა და ოპერაციათა კვლევაში. ისტორიულად ის იყო ერთ-ერთი ამოცანათაგანი, რომელიც სტიმულს აძლევდა ამ დარგის განვითარებას.

ფორმულირების სიმარტივე, დასაშვებ ამონახსნთა სიმრავლის სასრულობა, სიცხადე და ამავე დროს ვარიანტების სრული გადარჩევის კოლოსალური დანახარჯები უბიძგებენ მათემატიკოსებს ახალი რიცხვითი მეთოდების შექმნისაკენ.

კომივოიაჟერის ამოცანას დიდი პრაქტიკული გამოყენება აქვს, მათგან აღვნიშნავთ სატრანსპორტო საშუალებების მოძრაობის ოპტიმალური მარშრუტების დადგენის ამოცანას, ტექნიკური სქემის ოპტიმალური დაგეგმვის ამოცანა და სხვ.

ამოცანა ეკუთვნის NP-რთული ამოცანების კლასს. ცნობილია შემდეგი შედეგი: კომივოიაჟერის ამოცანა NP-რთულია იმ შემთხვევაშიც კი, როცა მანძილების მატრიცა სიმეტრიულია და შედგება 1-იანისა და 2-იანისგან.

ნაშრომის მიზანია კომივოიაჟერის ამოცანისთვის ე.წ. კონსტრუქციული ალგორითმების გამოკვლევა რიცხვითი ექსპერიმენტების საშუალებით, კონსტრუქციული ალგორითმების ქვეშ გვესმის ალგორითმები, რომლებიც თანდათანობით, ნაბიჯ-ნაბიჯ აგებენ დასაშვებ ამონახსნს. ასევე ლოკალური ოპტიმიზაციის ზოგიერთი ალგორითმის მოსინჯვა კომივოიაჟერის ევკლიდური ამოცანისთვის.

1. კომივოიაჟერის ამოცანა და ამოცანის ამოხსნის ალგორითმების მიმოხილვა

ბოლო წლებში კომბინატორული ოპტიმიზაციის რთული ამოცანების და კერძოდ, კომივოიაჟერის ამოცანის ალგორითმების განვითარების მიმართულებით ორი ტენდენცია გამოიკვეთა. პირველი მდგომარეობს ისეთი ალგორითმების დამუშავებაში, რომლებიც შეიცავენ მრავალ პროცედურას, 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.

კომივოიაჟერის ამოცანის ამოხსნის ზუსტი მეთოდები პირობითად შეიძლება შემდეგ ჯგუფებად დავყოთ: დინამიური პროგრამირება, შტოებისა და საზღვრების მეთოდი, ლაგრანჟის მამრავლთა მეთოდი, მკვეთი სიბრტყეების მეთოდი და სხვ.

ზემოთ თქმულიდან გამომდინარე, მაღალეფექტური ევრისტიკების დამუშავება რთული კომბინატორული ოპტიმიზაციის ამოცანების ამოსახსნელად აქტუალურია და დამოუკიდებელ ინტერესს იძენს, მისი პრაქტიკული მნიშვნელობის გამო. კომივოიაჟერის ამოცანას, რთულად ამოსახსნელი კომბინატორული ოპტიმიზაციის ამოცანებს შორის, ცენტრალური ადგილი უჭირავს .

1985 წელს გამოვიდა პირველი მონოგრაფია, რომელიც მთლიანად კომივოიაჟერის ამოცანისადმი იყო მიძღვნილი . მონოგრაფიის მიზანი იყო კომივოიაჟერის ამოცანის მაგალითზე ეჩვენებინა კომბინატორული ოპტიმიზაციის მეთოდების მთელი ძალა და სისუსტე.

1831 წელს გერმანიაში გამოვიდა წიგნი „ვინ არის კომივოიაჟერი და რა უნდა გააკეთოს მან თავისი საწარმოს აყვავებისათვის“. ერთ-ერთი რეკომენდაცია ამბობდა : „საჭიროა ვეწვიოთ რაც შეიძლება მეტ გასაღების პუნქტს ისე, რომ არც ერთში არ მივიდეთ ორჯერ“. როგორც ჩანს, ეს იყო კომივოიაჟერის ამოცანის პირველი ფორმულირება.

კომივოიაჟერის ამოცანის შესწავლის თანამედროვე ეტაპს უკავშირებენ კარლო მენგერს ევროპაში და ჰასლერ უიტნის ამერიკაში. კომივოიაჟერის ამოცანის ყველაზე გავრცელებული ფორმულირება შემდეგია: მოცემულია გარკვეული რეგიონის ქალაქების სია და მათ შორის მანძილების ცხრილი. უნდა ვიპოვოთ ჩაკეტილი მარშრუტი (ანუ მარშრუტი, რომელიც იწყება და მთავრდება ერთი და იგივე ქალაქში), რომელიც გაივლის ყველა ქალაქს ერთხელ და რომელსაც ექნება მინიმალური სიგრძე. თუმცა ხშირად ტერმინებს „ქალაქი“ და „მანძილი“ იყენებენ კომივოიაჟერის ამოცანის ისეთ ინტერპრეტაციებში, რომლებშიც ძნელია მათ მისცე გეოგრაფიული აზრი.

ვთქვათ $G=(V,A)$ ($G=(V,E)$) ორიენტირებული (არაორიენტირებული) გრაფია $G=(|V|=n)$ წვეროების სიმრავლით და $A(E)$ რკალების (წიბოების) სიმრავლით, $|A|=|E|=m$. ორიენტირებული გრაფისთვის იხმარება ტერმინები - რკალი, გზა, კონტური, გრაფებისათვის - წიბო,ჯაჭვი,ციკლი. ვგულისხმობთ რომ გრაფს არ აქვს მარყუჟი და ჯერადი რკალი (წიბო).

კომივოიაჟერის ამოცანა ისმის შემდეგნაირად: მოცემულ G გრაფში არსებობს თუ არა ჰამილტონის ციკლი? ჰამილტონის ჯაჭვის, ჰამილტონის გზის და ჰამილტონის კონტურის ამოცანა ფორმულირდება ანალოგიურად.

ვთქვათ $C=\|c_{ij}\|$ რკალების სიგრძეების $n\times n$ მატრიცაა. მაშინ კომივოიაჟერის არასიმეტრიული ამოცანა შემდეგნაირად ფორმულირდება: G გრაფში ვიპოვოთ მინიმალური სიგრძის ჰამილტონის ციკლი. თუ G გრაფში (i,j) წიბო არ არსებობს, შესაბამის c_{ij} ელემენტს C მატრიცაში ჩავთვლით უსასრულოდ დიდ რიცხვად.

თუ გრაფის წვეროები სიბრტყის წერტილებს წარმოადგენენ x_i და y_i კოორდინატებით, კომივოიაჟერის ამოცანის ძირითად ფორმულირებაში, c მატრიცის ელემენტები ამოცანის პრაქტიკული ხასიათიდან გამომდინარე შემდეგი ფორმულებით გამოითვლება

$c_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$ (ევკლიდური მატრიცა)

$c_{ij}=\sqrt{(x_i-x_j)^2+(y_i+y_j)^2+0,5(x_i-x_j)(y_i-y_j)}$ (აფინური მატრიცა)

$c_{ij}=|x_i-x_j|+|y_i+y_j|$ (მანჰეტენის მატრიცა)

$c_{ij} = \max\{|x_i - x_j|, |y_i + y_j|\}$ (ჩებიშევის მატრიცა)

$c_{ii} = \infty$ ყველა შემთხვევაში.

1.1 კომბინატორული ოპტიმიზაციის ამოცანები

ოპტიმიზაციის ამოცანებს შორის დიდ როლს თამაშობს დისკრეტული (სხვაგვარად კომბინატორული ოპტიმიზაციის) ამოცანები. მათი გადაჭრის პრობლემა უამრავ სიტუაციაში წარმოიშობა. ერთის მხრივ, რადგან ამ ამოცანებში დასაშვებ ამონახსნთა სიმრავლე სასრულია, შეიძლება ვიფიქროთ, რომ მათი ამოხსნა რაღაც გაგებით უფრო იოლი უნდა იყოს, თუმცა რიგ შემთხვევაში ეს პირიქითაა. მოვიყვანოთ ერთკრიტერიუმიანი ამოცანის მაგალითები და მათ საფუძველზე რამდენიმე განმარტება.

ვთქვათ X არის ამოცანის დასაშვებ ამონახსნთა სასრული სიმრავლე.

გვაინტერესებს $f : X \to Z$ ფუნქციის მინიმიზაცია.

$$min_{x \in X} \ f(x).$$

1.2 განმარტებები სირთულის თეორიიდან

ალგორითმულ სირთულეზე სასაუბროდ მოვიყვანოთ მოკლე და არაფორმალური განმარტებები სირთულის თეორიიდან.

სირთულის თეორია შეისწავლის რამდენად ადვილია გავცეთ პასუხი **ამოცნობის ამოცანას (Decision problem)**, რომელიც განსაზღვრულია რაღაც ფორმალურ სისტემაში და მათი პასუხია **"კი"** ან **"არა"**. სიადვილე თავისთავად დამოკიდებულია იმაზე, თუ რამდენად სწრაფად შეგვიძლია გავცეთ ამოცანას პასუხი, ანუ ყველაზე ცუდ შემთხვევაში (მონაცემის მიმართ) ბიჯების რა რაოდენობა დაჭირდება ალგორითმს.

სისწრაფის შესაფასებლად შემოღებულია აღნიშვნა O. შეგვიძლია ვთქვათ, რომ ალგორითმი "მუშაობს" $O\left(g\left(n\right)\right)$ დროში, თუ მას სჭირდება მაქსიმუმ $cg(n)$ (c რაღაც მუდმივაა) ბიჯი ამ ამოცანის ამოსახსნელად ნებისმიერი მონაცემისთვის.

ოპტიმიზაციის ამოცანები მჭიდრო კავშირშია გამოცნობის ამოცანებთან, საკმარისია შემოვიტანოთ რაიმე რიცხვი $b \in Z$ და ამოცანა დავსვათ შემდეგნაირად: არსებობს თუ არა ისეთი $x \in X$, რომ $f\left(x\right) \leq b.$

ამოცნობის ამოცანა ეკუთვნის P კლასს თუ არსებობს დეტერმინისტული ალგორითმი, რომელიც ხსნის ამ ამოცანას პოლინომიალურ დროში(მონაცემის მიმართ).

მაგალითად: მოცემული გვაქვს გრაფი $G = \left(V, E\right)$. წიბოებზე განსაზღვრული c წონის ფუნქციით. $c : E \to Z$. არსებობს თუ არა მარშრუტი T s წვეროდან u წვერომდე ისეთი, რომ

$$\sum_{t \in T} c(t) \leq b, b \in Z$$

უმოკლესი გზის ამოცანა P კლასს ეუთვნის. მოიძებნება ალგორითმი, მაგალითად დეიქსტრა **(Djkstra 1959)**, რომელიც უმოკლეს გზას პოულობს $O\left(\left|S\right|^2\right)$ დროში. (შემდეგ უბრალოდ ამ მანძილს შევადარებთ b-ს და ვიტყვით პასუხს "კი" ან "არა")

ამოცნობის ამოცანა ეკუთვნის NP კლასს თუ მოიძებნება არადეტერმინისტული ალგორითმი რომელიც ხსნის მას პოლინომიალურ დროში. სხვანაირად ეს ნიშნავს, რომ თუ გვაქვს რაიმე ამონახსნი x, შესაძლებელია შევამოწმოთ სამართლიანია თუ არა $x \in X$ და $f\left(x\right) \leq b$ პოლინომიალურ დროში. ცხადია $P \subset NP.$

A ამოცანა არის NP-რთული თუ ნებისმიერი ამოცანა NP კლასიდან შეგვიძლია დავიყვანოთ ამ ამოცანაზე პოლინომიალურ დროში. თუ ამავდროულად $A \in NP$ A ამოცანას ეწოდება NP-სრული.

ყოველი ამოცნობის ამოცანისათვის შეგვიძლია მოვიყვანოთ შესაბამისი დათვლის პრობლემა. ანუ მოცემული $b \in Z$ რამდენი $x \in X$ არსებობს ისეთი, რომ $f\left(x\right) \leq b$? ეს ამოცანები გაერთიანებული $\#P$ კლასში. დათვლის პრობლემა ეკუთვნის $\#P$ სრულ კლასს მაშინ და მხოლოდ მაშინ თუ ის არის $\#P$-ში, და არსებობს ისეთი

პოლინომიალური დაყვანა $\#P$ კლასის ნებისმიერი პრობლემიდან, რომელიც ინარჩუნებს **"კი"**- ების რაოდენობას.

1.3 დანიშვნისა და კომივოიაჟერის ამოცანების შესახებ

გავიხსენოთ დანიშვნის ამოცანის მათემატიკური მოდელი.

$$\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ij} \to \min,$$
$$\sum_{i=0}^{n} x_{ij} = 1, j = 0,\dots,n,$$
$$\sum_{j=0}^{n} x_{ij} = 1, i = 0,\dots,n,$$
$$x_{ij} \in \{0;1\}, i = 0,\dots,n;\ j = 0,\dots,n$$

აქ იგულისხმება, რომ გვაქვს n სამუშაო და ამ სამუშაოების შემსრულებელი n კანდიდატი, c_{ij} წარმოადგენს დანახარჯებს, როცა i -ურ სამუშაოს ასრულებს j -ური კანდიდატი. თუ შემოვიღებთ ბულის ცვლადებს x_{ij}, რომლებიც მიიღებენ 1-ის ტოლ მნიშვნელობას, როცა i -ურ სამუშაოს შეასრულებს j -ური მომხმარებელი, ხოლო 0-ს საწინააღმდეგო შემთხვევაში, მაშინ მივიღებთ ბულის პროგრამირების, ზემოთ მოყვანილ ამოცანას.

თუ $x_{ij} \in \{0;1\}$ პირობის ნაცვლად ჩავწერთ $x_{ij} \geq 0$ პირობას, მივიღებთ სატრანსპორტო ამოცანას $m = n, a_i = b_j = 1$ პირობებით. ამ ამოცანას ექნება ოპტიმალური მთელრიცხვა ამონახსნი, რადგან a_i და b_j რიცხვები მთელებია.

$X = (x_{ij})$ მატრიცას ეწოდება გადანაცვლებადი, თუ ის ყოველ სტრიქონში და ყოველ სვეტში შეიცავს ერთ 1-იანს. ბისტოქასტური ეწოდება მატრიცას რომელიც აკმაყოფილებს პირობებს:

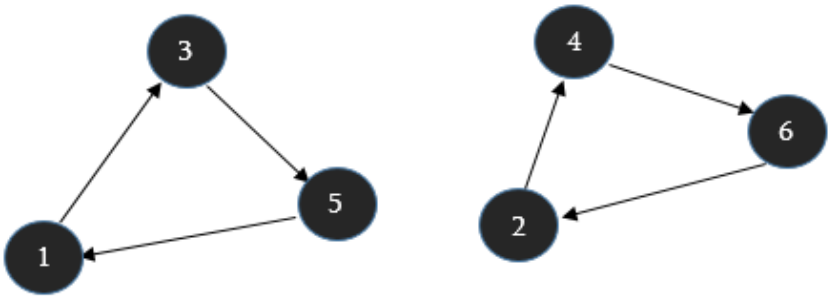
$$\sum_{i=0}^{n} x_{ij} = 1, j = 0,\dots,n; \sum_{j=0}^{n} x_{ij} = 1, i = 0,\dots,n; x_{ij} \geq 0$$

ბისტოქასტურ მატრიცათა სიმრავლე ამოზნექილია. მართლაც თუ $Y=\left(y_{ij}\right)$ და $Z=\left(z_{ij}\right)$ ორი ბისტოქასტური მატრიცაა, უშუალო შემოწმებით ვრწმუნდებით, რომ $x_{ij}=\alpha y_{ij}+\left(1-\alpha\right)z_{ij}$, სადაც $0\le\alpha\le1$, ელემენტების მქონე მატრიცა ასევე ბისტოქასტური იქნება.

ბირგკოფის თეორემა ამტკიცებს, რომ ბისტოქასტური მატრიცების სიმრავლის ამოზნექილ გარსს n^z განზომილებიან სივრცეში კუთხის წერტილებად გადანაცვლებადი მატრიცები აქვს. დანიშვნის ამოცანა $x_{ij}\ge0$ პირობით, წრფივი პროგრამირების ამოცანაა, ამიტომ მინიმალური მნიშვნელობა მიიღწევა კუთხის წერტილებში, ანუ გადანაცვლებადი მატრიცებისთვის.

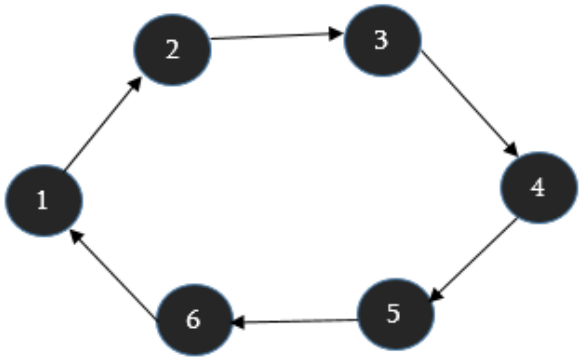
განვიხილოთ დანიშვნის ამოცანის ამონახსნის გეომეტრიული ინტერპრეტაციის საკითხი. ყოველ $X=\left(x_{ij}\right)$ მატრიცას, რომელიც დანიშვნის ამოცანის შეზღუდვებს აკმაყოფილებს, შევუსაბამოთ n წვეროიანი ორიენტირებული გრაფი. ამ გრაფში არსებობს $\left(i;j\right)$ რკალი, თუ $x_{ij}=1$. ამ დროს მიიღება ერთი ან რამდენიმე ციკლი. $n=6$ შემთხვევისთვის განვიხილოთ $\left(x_{ij}\right)$ მატრიცის მაგალითი, რომელიც აკმაყოფილებს დანიშვნის ამოცანის შეზღუდვებს.

	1	2	3	4	5	6
1			1			
2				1		
3					1	
4						1
5	1					
6		1				



ნახაზზე გამოსახულია ამ მატრიცის შესაბამისი გრაფი, რომელიც ორი ციკლისაგან შედგება. ციკლების მაქსიმალური რაოდენობა შეიძლება იყოს n , ეს მაშინ მოხდება, როცა i -ური კანდიდატი j -ურ სამუშაოზე დაინიშნება., ამ შემთხვევას ერთეულოვანი მატრიცა შეესაბამება. ციკლების მინიმალური რაოდენობა კი შეიძლება 1 იყოს. მოვიყვანოთ $\left(x_{ij}\right)$ მატრიცის მაგალითი ამ შემთხვევისთვის.

	1	2	3	4	5	6
1		1				
2			1			
3				1		
4					1	
5						1
6	1					



შესაბამისი გრაფი შეესაბამება კომივოიაჟერის ამოცანას. აქვე აღვნიშნოთ, რომ კომივოიაჟერისა და დანიშვნის ამოცანების მათემატიკური მოდელები მიზნის ფუნქციითა და შეზღუდვების ნაწილით ერთმანეთს ემთხვევა. კერძოდ, თუ ერთი და იგივე $C=\left(c_{ij}\right)$ მატრიცისთვის ვიპოვით ოპტიმალურ ამონახსნს, როგორც დანიშვნის ასევე კომივოიაჟერის ამოცანისათვის, ვნახავთ, რომ კომივოიაჟერის ამოცანის ამონახსნი ყოველთვის იქნება დანიშვნის ამოცანის ამონახსნი, მაგრამ პირიქით ეს ხშირად ასე არ არის. n -ზე ნაკლებგანზომილებიანი ციკლების არსებობა დანიშვნის ამოცანის ამონახსნში ეს არის მთავარი პრობლემა, რის გამოც დანიშვნის ამოცანა რჩება „ადვილად ამოხსნადი“ ამოცანების კლასში, კომივოიაჟერის ამოცანა კი ე.წ. „რთულად ამოხსნად“ ამოცანათა კატეგორიას მიეკუთვნება. იმისთვის რომ მივიღოთ კომივოიაჟერის ამოცანის სრულყოფილი მოდელი მთელრიცხვა პროგრამირების ამოცანის სახით , საჭიროა დამატებითი პირობები, რომლებიც უზრუნველყოფენ n განზომილებიანი ციკლის არსებობას. ამ პირობებს აქვს სახე:

$$u_i - u_j + nx_{ij} \le n-1; i, j = 1,2,\ldots,n; i \ne j$$

სადაც $u_i \ge 0$ დამატებითი (ახალი) ცვლადებია, შემდეგ მიიღება ნაწილობრივ მთელრიცხვა წრფივი პროგრამირების ამოცანა.

ვაჩვენოთ, რომ მოყვანილი პირობები უზრუნველყოფენ ზუსტად ერთი ციკლის არსებობას . დავუშვათ რომ არსებობს ორი ციკლი მაინც, მაშინ მოიძებნება ციკლი τ შედგენილი k რკალით, რომელიც არ გადის ქალაქზე 0;

$\tau = (i_1, i_2, ..., i_k, i_1)$ ამოვწეროტ ამ ციკლის ყოველი რკალისათვის ზემოთ მოყვანილი უტოლობები

$$u_{i_1} - u_{i_2} + n \le n-1,$$
$$u_{i_2} - u_{i_3} + n \le n-1,$$
$$\ldots\ldots\ldots\ldots\ldots,$$
$$u_{i_k} - u_{i_1} + n \le n-1$$

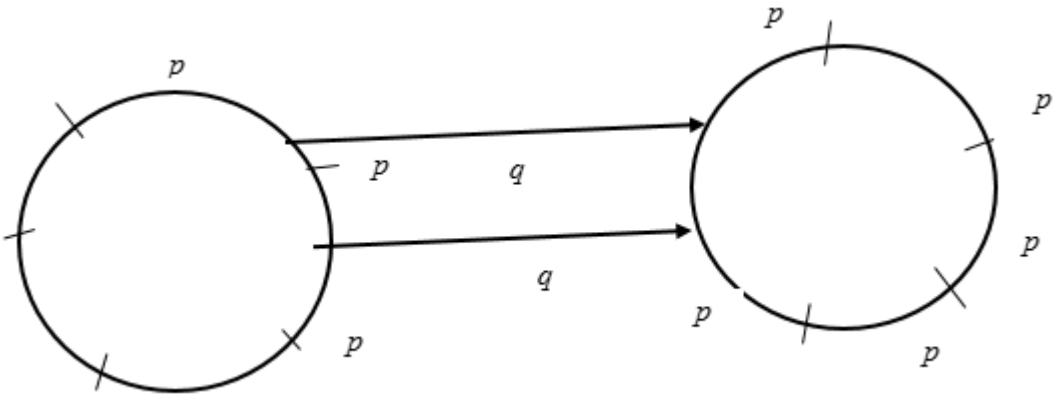
თუ ამ უტოლობების შეკრებით მივიღებთ $nk \le k(n-1)$, რაც შეუძლებელია, თუ $k \neq 0$. ე.ი, თუ არსებობს ციკლი, რომელიც არ გადის 0 ქალაქზე, მაშინ აღნიშნული პირობა არ სრულდება.

ვაჩვენოთ, რომ ყველა ციკლისთვის, რომელიც გადის 0 ქალაქზე და რომელიც მოიცავს ყველა ქალაქს, ეს პირობა შესრულებულია, ანუ შესაძლებელია u_i ცვლადების შესაბამისი მნიშვნელობების პოვნა. ავიღოთ $u_i = p$, თუ i-ურ ქალაქში შესვლა ხდება p ბიჯზე; მაშინ $u_i - u_j \le n-1$ ე.ი. როცა $x_{ij} = 0$ პირობა შესრულებულია. ვთქვათ $x_{ij} = 1$, მაშინ $u_i = p$ და $u_j = p+1$, $p-(p+1)+n \le n-1$, რადგან $n-1 = n-1$.

განვიხილოთ დანიშვნისა და კომივოიაჟერის ამოცანებში ერთი და იგივე C მატრიცისთვის მიზნის ფუნქციების მნიშვნელობის თანაფარდობის საკითხი ოპტიმალური ამონახსნებისათვის. ვთქვათ p_0 არის დანიშვნის ამოცანის ოპტიმალური ამონახსნი, $f(p_0)$ კი მიზნის ფუნქციის შესაბამისი მნიშვნელობა, z_0 კომივოიაჟერის ამოცანის ოპტიმალური ამონახსნი, $l(z_0)$ კი მიზნის ფუნქციის შესაბამისი მნიშვნელობა. ადვილი შესამჩნევია, რომ კომივოიაჟერის დასაშვებ ამონახსნთა სიმრავლე წარმოადგენს დანიშვნის ამოცანის დასაშვებ ამონახსნთა სიმრავლის ქვესიმრავლეს, ამიტომ $f(p_0) \le l(z_0)$. დანიშვნის ამოცანის ოპტიმალური ამონახსნი წარმოადგენს კომივოიაჟერი ამოცანის ამონახსნის ქვედა შეფასებას. ეს ფაქტი ხშირად გამოიყენება კომივოიაჟერის ამოცანის ამოხსნის ალგორითმებში. გავიხსენოთ რომ ეს შეფასება შეიძლება იყოს საკმაოდ ცუდი. ანუ

არსებობს ისეთი C მატრიცები, რომელთათვისაც $l(z_0) - f(p_0)$ არის საკმაოდ დიდი.

დავუშვათ, რომ დანიშვნის ამოცანის ოპტიმალური ამონახსნი შედგება ორი ციკლისაგან, C მატრიცის ყველა ელემენტი, რომელიც შედის ციკლში p -ს ტოლია, ხოლო C მატრიცის ყველა დანარჩენი ელემენტი q -ს ტოლია და $q > p$. გრაფიკულად ეს სიტუაცია ასე წარმოდგება



რომ გადავიდეთ კომივოიაჟერის ამოცანის ამონახსნზე, ანუ ერთ ციკლზე , საჭიროა პირველი და მეორე ციკლები გავწყვიტოთ აღნიშნულ ადგილებზე, ამოვაგდოთ შესაბამისი წიბოები $2p$ სიგრძით და დავამატოთ წიბოები $2q$ სიგრძით, მაშინ $l(z_0) - l(p_0) = 2q - 2p = 2(q - p)$, საიდანაც ჩანს, რომ ფიქსირებული p და q -სთვის $l(z_0) - l(p_0)$ სხვაობა შეიძლება იყოს ძალიან დიდი.

1.4 კომივოიაჟერის ამოცანის დასმის კომბინატორული ვარიანტი

კომივოიჟერმა(მოხეტიალე ვაჭარმა) უნდა შემოიაროს ქალაქების ფიქსირებული ერთობლიობა. მოძრაობა უნდა დაიწყოს იმ ქალაქიდან რომელშიც იმყოფება და უნდა დაამთავროს იგივე ქალაქში. ამასთან ყველა ქალაქში უნდა შევიდეს ერთხელ. ეს ისე უნდა გააკეთოს, რომ მთელი მოგზაურობის ჯამური

დანახარჯები იყოს მინიმალური.დანახარჯებში იგულისხმება ან თანხა, ან დრო, ან ჯამური გავლილი მანძილი.

კომივოიაჟერის ამოცანა (TSP – travelling salesman problem) კომბინატორული ოპტიმიზაციის ტიპური ამოცანაა, რომლის ჩამოყალიბება(სიტყვიერი ფორმულირება) მარტივი ამოცანის შთაბეჭდილებას ტოვებს, თუმცა ზუსტი ამონახსნის პოვნა სერიოზულ ძალისხმევას ითხოვს და დიდი განზომილებისთვის გადაუჭრელ პრობლემად რჩება. ბიზნეს გამოყენებაში გვხვდება კომივოიაჟერის ამოცანის სხვა ვარიანტი-ხელსაწყოთა ოპტიმალური გადაწყობის ამოცანის სახით.

ზოგად შემთხვევაში კომივოიაჟერის ამოცანა შეიძლება ჩამოყალიბდეს, როგორც გრაფში უმოკლესი სიგრძის ჰამილტონის ციკლის ამოცანა.

ვთქვათ, გვაქვს $n+1$ ქალაქი და $0;1;...,n$ ქალაქების ყოველი i და j წყვილისათვის მოცემულია მათ შორის „მანძილი“- $c_{ij} \geq 0$ კომივოიაჟერი იმყოფება 0 ქალაქში, უნდა ვიპოვოთ ქალაქების შემოვლის ისეთი მიმდევრობა, რომლისთვის ჯამური მანძილი იქნება მინიმალური. კომივოიაჟერის სიმეტრიული ამოცანისთვის $c_{ij}=c_{ji}$.

განვიხილოთ ამოცანის კომბინატორული დასმა. მარშრუტის ქვეშ გვესმის $Z=(0,i_1,\ i_2,\ldots,\ i_n,0)$, სადაც $i_1,\ i_2,\ldots,\ i_n$ არის $1,2,\ldots,n$ რიცხვების გადანაცვლება. Z მარშრუტის სიგრძე $l(Z)=\sum_{k=0}^{n} c_{i_k,i_{k+1}}, (i_0=i_{n+1}=0)$.

ვთქვათ Z არის ყველა მარშრუტის სიმრავლე $|Z|=n!$, უნდა ვიპოვოთ ისეთი z_0 მარშრუტი, რომ $l(z_0)=min\ l(z), z\in Z$.

განვიხილოთ ამოცანის დასმა მთელრიცხვა პროგრამირების ტერმინებში , შემოვიღოთ ბულის ცვლადები.

$$X_{ij} = \left\{ \begin{array}{ll} 1, & \text{თუ კომივოიაჟერი გადადის i-ური ქალაქიდან j-ში} \\ 0, & \text{წინააღმდეგ შემთხვევაში} \end{array} \right\}$$

მაშინ მარშრუტის სიგრძე იქნება $\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij} X_{ij} \to \min$.

ყოველ ქალაქში თითოჯერ შესვლისა და თითოჯერ გამოსვლის პირობებს კი უზრუნველყოფს შესაბამისად შემდეგი შეზღუდვები:

$$\sum_{i=0}^{n} X_{ij} = 1, i = 0,1,...,n$$

$$\sum_{j=0}^{n} X_{ij} = 1,\ j = 0,1,...,n$$

ამას ემატება $x_{ij} \{0;1\}$ $i, j = 0,\ 1,\ \ldots,\ n$ პირობები.

ადვილი შესამჩნევია , რომ მიღებული მოდელი არ განსხვავდება დანიშვნის ამოცანის მათემატიკური მოდელისაგან, რომლის ოპტიმალური ამონახსნი შეიძლება არ შეესაბამებოდეს კომივოიაჟერის დასაშვებ მარშრუტს. ამიტომ მიღებული შეზღუდვები არ არის სრული და საჭიროებს ისეთი პირობების დამატებას, რომელიც არ დაუშვებს ამოხსნაში რამდენიმე ციკლის არსებობას.

იმისთვის რომ მივიღოთ კომივოიაჟერის ამოცანის სრულყოფილი მოდელი მთელრიცხვა პროგრამირების ამოცანის სახით , საჭიროა დამატებითი პირობები, რომლებიც უზრუნველყოფენ n განზომილებიანი ციკლის არსებობას. ამ პირობებს აქვს სახე:

$$u_i - u_j + nx_{ij} \le n-1; i, j = 1,2,...,n; i \ne j$$

სადაც $u_i \ge 0$ დამატებითი (ახალი) ცვლადებია.

2. ევრისტიკული ალგორითმები კომივოიაჟერის ამოცანისათვის

ევრისტიკული ალგორითმები, („ევრისტიკები“) წარმოადგენენ პრაქტიკული (დიდ განზომილებიანი) ამოცენების ამოხსნის ძირითად ინსტრუმენტს. ევრისტიკების ძირითადი იდეა მდგომარეობს ამოცანის სპეციფიკის გათვალისწინებაში მარტივი საშუალებით.

ევრისტიკული ალგორითმები შეიძლება (პირობითად) დაიყოს შემდეგ ტიპებად:

1. ტურის აგების ალგორითმები
2. ტურის გაუმჯობესების მონოტონური ალგორითმები
3. კომბინირებული ალგორითმები
4. დიალოგური (ადამიანი-მანქანა) ალგორითმები.

ტურის აგების ალგორითმებში ხდება წვეროების (რკალების) ტურში მიმდევრობის ჩართვა გარკვეული წესების საფუძველზე.

ტურის გაუმჯობესებისალგორითმებში ამცირებენ ტურის სიგრძეს წვეროების გადანაცვლებით საწყის ტურში.

კომბინირებული ალგორითმები წარმოადგენენ პირველი ორი კლასის შერწყმას.

თუ შესაძლებელია ადამიანის ფაქტორის გათვალისწინება (ჩართვა) გამოთვლით პროცესში მისი მართვის თვალსაზრისით, მაშინ ვღებულობთ დიალოგურ ალგორითმს.

ჩვენ ძირითადად პირველი და მეორე ტიპის ალგორითმებს განვიხილავთ.

ევრისტიკული ალგორითმების ქვეშ გვესმის ალგორითმები, რომლებიც ემყარებიან პრაქტიკულ მოსაზრებებს, ანუ ალგორითმები, რომელთათვისაც ფორმალური დასაბუთებები არ გვაქვს და რომლებიც ემყარებიან ამოცანის სტრუქტურის ანალიზს.

2.1 უახლოეს წერტილში გადასვლის ალგორითმი.

ხშირად ამ ალგორითმს „უახლოეს მეზობლის“ ალგორითმს უწოდებენ. ვირჩევთ საწყის წერტილს და ვპოულობთ ამ წერტილის უახლოეს წერტილს, შემდეგ დარჩენილებიდან უახლოესს და ა.შ.

ვთქვათ $P=\{1,2,...,n\}, c_{ij}\geq 0$ მანძილებია $i\in P$ და $j\in P$ წერტილებს შორის i_1 იყოს საწყისი წერტილი. ვიპოვოთ i_2 წერტილი ისეთი, რომ $c_{i_1}c_{i_2}=\min c_{i_1 i}, i\neq i_1$. ვთქვათ $Z_k=\{i_1,...,i_k\}$ არის ასაგები მარშრუტების აგებული ნაწილი. i_{k+1} წერტილი განისაზღვრება პირობიდან

$$c_{i_k i_{k+1}}=\min c_{i_k i}, i\in P\setminus\{Z_k\}, Z_{k+1}=(i_1,...,i_k,i_{k+1})$$

თუ $P\setminus\{Z_k\}\neq\varnothing$, მაშინ ვიპოვით i_{k+2} წერტილს და ა.შ. მიღებული Z მარშრუტის $l(Z)$ სიგრძე განისაზღვრება $l(Z)=\sum_{k=1}^{n-1}c_{i_k,i_{k+1}}+c_{i_n i_1}$ გამოსახულებით. ცხადია რომ აღწერილი ალგორითმის სირთულე შეიძლება შეფასდეს, როგორც $O(n^2)$.

მოვიყვანოთ მაგალითი, რომლისთვისაც აღწერილი ალგორითმი იძლევა საკმარისად „ცუდ“ ამონახსნს. ვთქვათ საწყის წერტილად ავირჩიეთ პირველი წერტილი, ხოლო მინიმალების c_{ij} მატრიცას აქვს შემდეგი სახე:

$$c_{ij}=\begin{cases} q, \text{ if } \ j=i+1; i=1,2,...n-1; \\ A, \text{ if } \ i=n, j=1; \\ P, \text{ other} \end{cases}$$

სადაც $A>p>q$. C მატრიცა არის სიმეტრიული, $A=C_{n1}\neq C_{1n}=p$. დავუშვათ, რომ მანძილებისათვის სამკუთხედის უტოლობა არ არის შესრულებული.

აღწერილი ალგორითმი იძლევა მარშრუტს $Z=(1,2,...,n)$,

$l(z)=(n-1)q+A.$

A-ს ცვლილების ხარჯზე შესაძლებელია მარშრუტის სიგრძე გავხადოთ რაგინდ დიდი, Z_0 ოპტიმალურ ამოხსნას კი აქვს სახე $(1,2,\ldots,n-2,n,n-1)$,

$l(z_0)=(n-3)q+3p$, მაშინ

$$\frac{l(z)}{l(z_o)}=\frac{(n-1)q+A}{(n-3)q+3p}=\frac{(n-1)q+A}{nq+3(p-q)}<\frac{(n-1)q+A}{nq},$$

რადგან p-1>0,

$\frac{l(z)-l(z_0)}{l(z_o)}<\frac{(n-1)q+A}{nq}-1=\frac{A-q}{nq}$, ეს არის ზედა შეფასება.

ქვედა შეფასებისათვის გვაქვს:

$\frac{l(z)}{l(z_o)}=\frac{(n-1)q+A}{(n-3)q+3p}>\frac{(n-1)q+A}{(n-3)p+3p}=\frac{(n-1)q+A}{np}>\frac{nq-p+A}{np}$, საიდანაც

$$\frac{l(z)-l(z_0)}{l(z_0)}>\frac{nq-p+A}{np}-1=\frac{A-p}{np}-\frac{p-q}{p}=\frac{A-p}{np}-1+\frac{q}{p}>\frac{A-p}{np}-1,$$

ე.ი. $\frac{A-p}{np}-1<\frac{l(z)-l(z_0)}{l(z_0)}<\frac{A-q}{nq}$

საიდანაც ჩანს, რომ გადახრა ოპტიმალური ამონახსნიდან უსასრულოდ იზრდება, როცა $A\to\infty$, ფიქსირებული p და q -სთვის .

აღწერილი ალგორითმის მოდიფიცირებულ ალგორითმში საწყის წერტილად იღებენ $1,2,\ldots,n$ წერტილებს და აირჩევენ საუკეთესო მარშრუტს. ამ ალგორითმის სირთულე შეიძლება შეფასდეს, როგორც $O(n^3)$.

```
1.   V = {1, ..., n-1}
2.   U = {0}
3.   while V not empty
4.      u = most recently added vertex to U
5.      Find vertex v in V closest to u
6.      Add v to U and remove v from V.
7.   end while
8.   Output vertices in the order they were added to U
```

2.2 ევრისტიკული ალგორითმი „აირჩიე უმოკლესი რკალი”

ეს ალგორითმი მუშაობას იწყებს მინიმალური სიგრძის რკალის არჩევით $c_{i_1 i_2} = \min c_{ij}$ (თუ ასეთი რამდენიმეა, ვირჩევთ ნებისმიერს). ვთქვათ, $z_k = (i_1, i_2, ..., i_k)$ უკვე აგებული ნაწილია, მას ემატება ან (i_k, i_{k+1}) რკალი, ან (i_{k+1}, i_1) რკალი. ვთქვათ, $c_{i_k i_{k+1}} = \min c_{i_k i}, c_{i_{k+1} i_1} = \min c_{i i_k}, i \in p \setminus \{z_k\}$, თუ $c_{i_k i_{k+1}} < c_{i_{k+1} i_1}$, მაშინ $Z_{k+1} = (i_1, i_2, ..., i_k, i_{k+1})$, თუ პირიქით, მაშინ $Z_{k+1} = (i_{k+1}, i_1, ..., i_k)$, ხელახლა აღვნიშნოთ აგებული მარშრუტის ნაწილი $Z_{k+1} = (i_1, i_2, ..., i_k, i_{k+1})$, თუ $p / \{Z_{k+1}\} \neq \varnothing$ გადავდივართ შემდეგ ბიჯზე. ამ ალგორითმის სირთულე ასე შეიძლება შეფასდეს, $\min c_{1j}$ გამოთვლა საჭიროებს n^2 ოპერაციას, ერთი გადასვლის დამატება ნაწილობრივ აგებულ მარშრუტზე ითხოვს 2(n-k) ოპერაციას, ამიტომ სირთულე შეიძლება შეფასდეს ასე: $n^2 + 2\sum_{k=1}^{n-1}(n-k) = 0(n^2)$

```
1.    way=empty
2.    counter=0
3.    e= minimal edge in E ={e1,e2,…en}
4.     remove e from E
5.    way= way.add(e)
6.    While  counter<size(E)-1  do
7.    e= minimal edge in E
8.       if   e is Safe Edge
9.             way= way.add(e)
10.            remove e from E
11.            counter++
12.      end if
13.         remove e from E
14.    end while
15.    return way
```

2.3 2-opt ამონახსნის მოძებნის მეთოდი

ალგორითმი მუშაობს შემდეგნაირად, გენერირდება საწყისი ტური და შემდეგ ხდება ამ ტურში ორი არამეზობელი წიბოების გადანაცვლება ისე, რომ ციკლი არ დაირღვეს. ყოველ ეტაპზე ხდება ტურის ღირებულების გამოთვლა და თავდაპირველ ღირებულებასთან შედარება, თუ მეორე შედეგი უფრო პატარა რიცხვია, დავიტოვებთ ამ ტურს და გავაგრძელებთ იგივე პროცედურას სხვა ტურებზე. უნდა აღინიშნოს რომ 2-opt ამონახსნის მეთოდით მიღებული შედეგი ხშირად შესაძლოა დაემთხვეს ზემოთ მოყვანილი ალგორითმებით მიღებულ შედეგს.

```
1.   T = some starting tour
2.   noChange = true
3.   repeat
4.      for all possible edge-pairs in T
5.         T' = tour by swapping end points in edge-pair
6.         if T' < T
7.             T = T'
8.             noChange = false
9.             break
10.        endif
11.     endfor
12.  until noChange
13.  return T
```

სტატისტიკა

ქვემოთ მოცემულია მატრიცები:

როცა $n = 25$.

99999 182 86 129 211 185 185 41 174 106 18 73 186 71 26 233 288 212 201 183 290 208 93 373 453
182 99999 107 142 195 22 50 144 85 165 200 109 167 114 208 128 231 36 144 126 231 45 272 305 396
86 107 99999 167 249 99 99 58 88 151 104 26 100 50 112 208 252 126 167 162 264 122 179 308 417
129 142 167 99999 82 164 115 163 220 23 146 148 267 124 136 270 341 178 254 236 343 187 158 426 506
211 195 249 82 99999 217 175 245 280 105 228 230 349 206 218 323 423 231 336 318 425 240 240 500 588
185 22 99 164 217 99999 72 144 63 187 203 125 145 114 211 119 209 27 122 104 211 23 272 283 374
185 50 99 115 175 72 99999 144 127 138 203 117 199 114 211 167 271 75 184 166 270 85 272 355 436
41 144 58 163 245 144 144 99999 146 140 59 46 158 39 67 201 256 171 169 151 258 167 134 341 421
174 85 88 220 280 63 127 146 99999 237 192 112 82 136 200 120 178 88 79 95 180 65 267 240 343
106 165 151 23 105 187 138 140 237 99999 123 125 251 113 113 275 330 201 243 225 332 210 135 415 495
18 200 104 146 228 203 203 59 192 123 99999 91 204 89 22 251 306 230 219 201 308 226 87 391 471
73 109 26 148 230 125 117 46 112 125 91 99999 126 24 99 186 241 145 154 136 243 148 168 326 406
186 167 100 267 349 145 199 158 82 251 204 126 99999 150 212 158 189 170 96 114 187 147 279 208 321
71 114 50 124 206 114 114 39 136 113 89 24 150 99999 97 162 217 141 130 112 219 137 158 302 382
26 208 112 136 218 211 211 67 200 113 22 99 212 97 99999 259 314 238 227 209 316 234 67 399 479
233 128 208 270 323 119 167 201 120 275 251 186 158 162 259 99999 114 92 68 50 103 108 320 206 279
288 231 252 341 423 209 271 256 178 330 306 241 189 217 314 314 99999 203 99 105 36 186 375 94 165
212 36 126 178 231 27 75 171 88 201 230 145 170 141 238 92 203 99999 116 98 195 23 299 288 368
201 144 167 254 336 122 184 169 79 243 219 154 96 130 227 68 99 116 99999 18 101 99 288 193 264
183 126 162 236 318 104 166 151 95 225 201 136 114 112 209 50 105 98 18 99999 107 81 270 190 270
290 231 264 343 425 211 270 258 180 332 308 243 187 219 316 103 36 195 101 107 99999 188 377 128 195
208 45 122 187 240 23 85 167 65 210 226 148 147 137 234 108 186 23 99 81 188 99999 295 271 351
93 272 179 158 240 272 272 134 267 135 87 168 279 158 67 320 375 299 288 270 377 295 99999 460 540
373 305 308 426 500 283 355 341 240 415 391 326 208 302 399 206 94 288 193 190 128 271 460 99999 113
453 396 417 506 588 374 436 421 343 495 471 406 321 382 479 279 165 368 264 270 195 351 540 113 99999

როცა $n = 5$

9999 48 27 31 43
33 9999 28 44 43
41 28 9999 40 36
37 35 29 9999 46
48 48 25 29 9999

სტატისტიკური ცხრილი:

n	წადი უახლოესში	$\frac{L_A - L^*}{L^*} * 100\%$	მინიმალური რკალი	$\frac{L_A - L^*}{L^*} * 100\%$	2-Opt	$\frac{L_A - L^*}{L^*} * 100\%$
5	162	0,0125	163	0,0188	161	0,0006
25	1772	0,0357	1741	0,0175	1733	0,0129

დასკვნა

ევრისტიკული ალგორითმები წარმოადგენენ კომივოიაჟერის პრაქტიკული ამოცანების ამოხსნის ძირითად ინსტრუმენტს. ნაშრომში განხილულია მეთოდები,რომლებიც გულისხმობენ, რომ ამოცანის შესაბამისი გრაფი არის სრული, ხოლო შესაბამისი მატრიცა-მეტრიკული (მაგალითად: ევკლიდური).

რიცხვითი ექსპერიმენტი ჩატარდა ძირითადად სამი ტიპის „წადი უახლოეს ქალაქში“ , „აირჩიე მინიმალური რკალი“ და „ლოკალური ძებნის 2-opt ამონახსნის მოძებნის“ ალგორითმებისთვის, ცნობილი სატესტო $n = 5$ და $n = 25$ ამოცანისათვის. გამოთვლილია სიდიდეები $\frac{L_A - L^*}{L^*} * 100\%$, სადაც L_A ამონახსნის მიახლოებითი მნიშვნელობაა, L^* - კი ოპტიმალური მნიშვნელობაა.

ლიტერატურა

1. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация: Алгоритмы и сложность. М.: Мир, 1985

2. Lawler E.L., Lenstra J.K. , Rinnooy Kan A.H., Shmoys D.B. The travelling salesman problem. A Guided Tour of Combinatorial Optimization. N.Y.; J.Wiley & Sons, 1985.

3. Golden B., Bodin L., Doylet., Stewart W.Ir. Aproximate travelling salesman problem || Oper.Res. 1980. V.28,N3. P.694-711

4. М.Хелд, Р.Карп. Применение динамического программирования к задачам упарядочениюю. 1964 г.

5. Р.Беллман. Применение динамического программирования к задачае о коммивояжере. 1964 г.