



გადაწყვეტილების მიღების
ხელშემწყობი სისტემის
აგების ზოგიერთი საკითხი
მარშრუტიზაციის ამოცანებში

ივანე ჯავახიშვილის
სახელობის თბილისის
სახელმწიფო უნივერსიტეტი

ზუსტ და საბუნებისმეტყველო
მეცნიერებათა ფაკულტეტი

ავტორები:

ნიკო ძნელაძე
ბექა ჭელიძე
მიხეილ ჭელიძე

ხელმძღვანელი
თსუ ასოცირებული პროფესორი
ბექან ღვაბერიძე

ქ.თბილისი

2014 წელი

სარჩევი

ანოტაცია.....	2
Annotation	3
შესავალი	4
§1 მარშრუტიზაციის სხვადასხვა სახის ამოცანების მიმოხილვა.	5
§2 დასაშვები მარშრუტების აგების ევრისტიული მეთოდები.	7
2.1 კომპიუტერის ამოცანის დასმა და მისი ამოხსნის მეთოდების მიმოხილვა	8
2.1.1 ევრისტიული ალგორითმები კომპიუტერის ამოცანისთვის	9
2.1.2 მეორე ევრისტიული ალგორითმი, აირჩიე უმოკლესი რკალი.....	10
2.2 შტოების და საზღვრების მეთოდი კომპიუტერის ამოცანისთვის	11
§3 უმცირესი დაყოფის ამოცანა და ამოხსნის ალგორითმი	14
3.1 ამოცანის ფორმულირება.....	14
3.2 პრობლემის შემცირება.....	15
3.3 ხის ძებნის ალგორითმი დაყოფის ამოცანისთვის	17
3.3.1 ალგორითმი	17
მაგალითი	19
დასკვნა.....	22
ლიტერატურა.....	23

ანოტაცია

ავტორები: ნიკო ძნელაძე, ბექა ჭელიძე, მიხეილ ჭელიძე.

სათაური: გადაწყვეტილების მიღების ხელშემწყობი სისტემის აგების ზოგიერთ საკითხი მარშრუტიზაციის ამოცანებში.

განხილულია მარშრუტიზაციის ერთი კლასის ამოცანების ამოხსნის ალგორითმის აგების საკითხები. რაც სატრანსპორტო მარშრუტიზაციის პრაქტიკულ ამოცანებში გადაწყვეტილების მიღების ხელშემწყობი სისტემის აგების ერთ-ერთ მთავარ შემადგენელ ნაწილს წარმოადგენს.

ალგორითმი პირობითად სამ მთავარ ნაწილად იყოფა: დასაშვები მარშრუტების აგება, დასაშვები მარშრუტებისთვის კომივოიაჟერის ამოცანის ამოხსნა და დაყოფის ამოცანის ამოხსნის გზით ოპტიმალური მარშრუტების ამორჩევა.

კომივოიაჟერის ცნობილი 25 პუნქტიანი ამოცანის მოდიფიცირების გზით მიღებულია მარშრუტიზაციის ამოცანა, რომელიც ამოხსნილია აღნიშნული მიმდევრობით.

Annotation

Authors: Niko Dzeladze, Beka Tchelidze, Mikheil Tchelidze.

Title: Some topics of system that is contributing decision making in routing problems.

It's discussed topic of constructing the algorithm of solving one class problem of routing, which is one of the core components of system aiding decision making of practical vehicle routing problem.

Algorithm is divided into three main components: Possible route construction, solving problems for TSP possible routes, and selecting optimal route of set partitioning problem.

Routing problem is derived from well-known 25 object TSP method, which is solved as stated above.

შესავალი

ჩვენი მიზანია მარშრუტიზაციის პრობლემის გადაჭრა, უფრო მარტივად რომ ვთქვათ, გვაქვს საწყობი სადაც ინახება პროდუქტი, ტრანსპორტი (მხოლოდ ერთი სახეობის) რომელსაც შეზღუდული აქვს პროდუქტის ტვირთამწეობა და განარბენი და პუნქტები რომლებიც ითხოვენ რაღაც რაოდენობის პროდუქტს. ჩვენი მიზანია სატრანსპორტო საშუალებები გავუშვათ ისეთი მარშრუტებით, რომ მათი ჯამური განარბენი იყოს მინიმალური და ყველა პუნქტის მოთხოვნა დააკმაყოფილონ. ამასთან, ყველა პუნქტში მიდის მხოლოდ ერთ ტრანსპორტი.

§1-ში განვიხილავთ მარშრუტიზაციის ამოცანას, მის სხვადასხვა სახეს და განსხვავებებს.

§2-ში აღწერილია ევრისტიული ალგორითმი, რომლის მეშვეობით იქმნება დასაშვები მარშრუტების სია, რომელიც უნდა გაიაროს ტრანსპორტმა.

2.1-ში განხილულია კომივოიაჟერის ამოცანა, რაც მარტივად რომ ვთქვათ, არის უმოკლესი გზის პოვნა ქალაქების შემოვლისა, სადაც ყველა ქალაქს გავივლით მხოლოდ ერთხელ და დავბრუნდებით საწყის ქალაქში. ამ პრობლემას ვწყვეტთ ყოველი დასაშვები მარშრუტისათვის და გვადლევს ამ მარშრუტების უმოკლეს მანძილებს. 2.1.1, 2.1.2 და 2.2 -ში განხილულია ამ პრობლემის გადამწყვეტი ალგორითმები.

2.1.1-ში ევრისტიული ალგორითმი „უახლოეს წერტილში გადასვლის ალგორითმი“ არის აღწერილი, რომელიც ზემოთ (§3-ში) აღწერილი ამოცანას ხსნის. თუმცა ეს არ იძლევა ოპტიმალურ ამონახსნს.

2.1.2-ში განხილულია ევრისტიული ალგორითმი „აირჩიე უმოკლესი რკალი“, რომელიც წინა პარაგრაფში განხილულ ალგორითმს უმეტეს პრაქტიკულ შემთხვევებში ბევრად ჯობნის და უფრო მიახლოებულ პასუხსაც იძლევა. მათი უარყოფითი მხარე ის არის, რომ შესაძლოა მოგვცენ ზუსტი ამონახსნისგან საკმაოდ გადახრილი ამონახსნი. დადებით მხარეში შეგვიძლია ჩავთვალოთ ის, რომ საკმაოდ დიდი რაოდენობის პუნქტებისთვის გვადლევს ამონახსნს რეალურ დროში.

2.2-ში აღწერილია კომივოიაჟერის ამოცანის ამოხსნის ზუსტი ალგორითმი, რომელიც წინა ალგორითმებს (მიუხედავად მათი კარგი შედეგებისა) ჯობნის და საკმარისად კარგ პასუხსაც იძლევა. ზუსტი ალგორითმის უარყოფითი მხარე ის არის, რომ დიდი რაოდენობის პუნქტებისთვის ამონახსნს რეალურ დროში ვერ იძლევა.

§3-ში ჩამოყალიბებულია ამოცანა, რომელიც დასაშვები მარშრუტებიდან ახდენს იმ მინიმალური რაოდენობის მარშრუტების ამორჩევას, რომელთა შორის მანძილების ჯამი მინიმალურია.

§1 მარშრუტიზაციის სხვადასხვა სახის ამოცანების მიმოხილვა.

ტრანსპორტის მარშრუტიზაციის სხვადასხვა სახის ამოცანების (Vehicle Routing Problem, VRP) ამოხსნის მიზანია სატრანსპორტო საშუალებების (სს) ისეთი მარშრუტების შედგენა, რომლებიც უზრუნველყოფენ ტვირთის მომხმარებლებთან გადაზიდვის ხარჯების მინიმიზაციას. მარშრუტიზაციის კლასიკური ამოცანა შეიძლება წარმოვიდგინოთ გრაფის $G = (N, A)$ სახით, სადაც N წვეროების სიმრავლეა, რომლებიც შეესაბამებიან კლიენტებს $1, 2, \dots, n$ და საწყობს $n + 1$, რომელშიც იწყება და მთავრდება მარშრუტები, A რკალების სიმრავლეა, რომელიც შეესაბამება კლიენტებს შორის დამაკავშირებელ გზებს. ყოველი კლიენტი ხასიათდება ტვირთზე მოთხოვნით P_i .

$I = \{1, 2, \dots, n + 1\}$, $C = (C_{ij})$, $i, j \in I$ -მანძილების მატრიცა;

n ცალი პუნქტი, $n + 1$ - საწყობის ნომერი;

D -სატრანსპორტო საშუალების მაქსიმალური განარბენი;

Q -სატრანსპორტო საშუალების ტვირთამწეობა;

$1 \leq P_i < Q$, $i = 1, \dots, n$, პუნქტის მოთხოვნა პროდუქტზე;

$\{M_k\}$, $k = 1, \dots, m$, დასაშვები მარშრუტების სიმრავლე, სადაც

$M_k = \{n + 1, i_1^k, \dots, i_{l_k}^k, n + 1\}$, $i_j^k \in \{1, \dots, n\}$, $j = 1, \dots, l_k$,

$1 \leq l_k \leq n$ (m და l_k წინასწარ ფიქსირებული არ არის)

$$\bigcup_{k=1}^n M_k = I;$$

$$M_k \cap M_q = \{n + 1\}, \quad k, q \in \{1, \dots, m\}, \quad k \neq q$$

$$\sum_{j=1}^{l_k} P_{i_j^k} \leq Q;$$

$$C_k = C_{n+1, i_1^k} + C_{i_{l_k}^k, n+1} + \sum_{j=2}^n C_{i_j^k, i_{j-1}^k} \leq D,$$

$$\sum_{k=1}^m C_k \rightarrow \min.$$

სხვადასხვა შეზღუდვები წარმოშობს სატრანსპორტო მარშრუტიზაციის ამოცანების სხვადასხვა სახეობას [1;2]:

1. CVRP(Capaciated VRP)-მარშრუტიზაცია შეზღუდვებით ტვირთამწეობაზე (კლასიკური ამოცანა)

2. VRPTW(VRP With Time Windows)-მარშრუტიზაცია ტვირთის ჩაბარებაზე დროის შეზღუდვით (დროითი ფანჯრები)
3. MDVRP (Multiple Depot VRP)-მარშრუტიზაცია რამდენიმე საწყობით.
4. HVRP(Heterogenous VRP)- მარშრუტიზაცია სხვადასხვა ტიპის ტრანსპორტით.
5. SVRP(Stochastic VRP)-მარშრუტიზაცია შემთხვევითი ხასიათის მონაცემებით.
6. DVRP(Dynamic VRP)-დინამიური მარშრუტიზაცია.
7. FVRP(Fuzzy VRP)-მარშრუტიზაცია არამკაფიო ხასიათის (ექსპერიმენტულ ცოდნაზე დაფუძნებული) მონაცემებით.

1. CVRP-ცნობილია სატრანსპორტო საშუალების მაქსიმალური ტვირთამწეობა, მიზანია დავალების შესრულება მინიმალური დანახარჯებით (მაგალითად, მინიმალური ჯამური განარბენი).
2. VRPTW- ყოველი კლიენტისთვის ცნობილია დროითი შუალედი $[t_0; t_1]$ —რომელშიც უნდა მოხდეს მომსახურება (ტვირთის ჩაბარება) მიზანია მომსახურების დროის მინიმიზაცია.
3. MDVRP-ტვირთი განთავსებულია რამდენიმე საწყობში, სატრანსპორტო საშუალებები ბრუნდებიან იმავე საწყობში საიდანაც იწყებენ მოძრაობას. მიზანი-დანახარჯების მინიმიზაცია.
4. HVRP-გვაქვს სხვადასხვა ტვირთამწეობის (ტიპის) სს. მიზანია ჯამური დანახარჯების მინიმიზაცია.
5. SVRP-თითოეული კლიენტისთვის მოთხოვნა არის შემთხვევითი სიდიდე, მანძილები (გადაადგილების დრო) შემთხვევითი სიდიდეებია. მიზანი-დანახარჯების მინიმიზაცია.
6. DVRP-ამონახსნის აგება ხდება სრული ინფორმაციის მიღებამდე. ახალი ინფორმაციის მიღებისთანავე ხდება ამონახსნის მოდიფიცირება. ფაქტიურად, მარშრუტების დაგეგმვა ხდება სამუშაოს შესრულების პროცესის პარალელურად. მიზანი-ხარჯების მინიმიზაცია.
7. FRVP-მონაცემების (გადაადგილების დრო და შესაძლებლობა) აქვთ ექსპერიმენტული ხასიათი. მიზანი-ჯამური ხარჯების მინიმიზაცია.

§2 დასაშვები მარშრუტების აგების ევრისტიული მეთოდები.

მარშრუტიზაციის ამოცანების ამოხსნის ერთერთ მისაღებ მიდგომად ითვლება მიდგომა, რომელიც ემყარება მარშრუტიზაციის ამოცანის მიყვანას უმცირესი დაყოფის ამოცანაზე [2;4].

P - n სიგრძის პუნქტების სიმრავლე (p_0 ნულოვანი პუნქტი არის საწყისი პუნქტი ანუ საწყობი).

Q - ტრანსპორტის მაქსიმალური ტევადობა.

D - ტრანსპორტის მაქსიმალური განარბენი.

Z - დასაშვები მარშრუტების სიმრავლე. Z_k არის ერთ-ერთი დასაშვები მარშრუტი.

b_i - i -ური პუნქტის მოთხოვნა.

c_{ij} - მანძილი i -დან j -მდე.

R_{ij} - უმოკლესი მარშრუტი i -დან j -მდე (არ შეიცავს i -ს და j -ს).

q - Z_k -ში არსებული პუნქტების მოთხოვნების ჯამი.

d - Z_k მარშრუტის მანძილი.

განვსაზღვროთ $F(i, j, q, Z_k)$ ფუნქცია რომელიც ჩაითრევს ყველა იმ პუნქტს, რომელსაც i -დან j -მდე გზაში გაივლის. Z_k არის იმ პუნქტების სია, რომლებიც უკვე ჩათრეულია.

$F(i, j, q, Z_k)$:

$$\forall r \in R_{ij} \text{ თუ } q + b_r \leq Q \text{ და } Z_k \cap \{r\} = \emptyset \text{ მაშინ } Z_k = Z_k \cup \{r\} \quad q = q + b_r$$

განვსაზღვროთ $G(s, q, d, Z_k, i)$ რომელიც ჩაითრევს ახალ პუნქტებს Z_k მარშრუტში, დაწყებული s პუნქტიდან. i კი არის ბოლო პუნქტი Z_k -ში.

$G(s, q, d, Z_k, i)$:

$$\begin{aligned} &\forall p_i \in P \quad j = s, s+1, \dots, n, 1, \dots, s-1 \text{ თუ } Z_k \cap \{p_i\} = \emptyset \text{ და } b_j + q \leq Q \text{ და} \\ &d + c_{ij} + c_{j0} \leq D \text{ მაშინ} \\ &d = d + c_{ij}; \quad q = q + b_j; \quad Z_k = Z_k \cup \{p_j\}; \quad F(i, j, q, Z_k); \quad i = j; \end{aligned}$$

ალგორითმი:

- ბიჯი 1: $k_1 = 1$
- ბიჯი 2: $\forall p_i \in P \ i = 1, \dots, n$
- ბიჯი 2.1: $q = b_i; d = c_{0i}; Z_{k_i} = \emptyset$
- ბიჯი 2.2: $F(0, j, q, Z_{k_i})$
- ბიჯი 2.3: $Z_{k_i+j} = Z_{k_i} \ j = 1, \dots, 4$
- ბიჯი 2.4: $G\left(\left(\frac{n}{5}\right) * j, q, d, Z_{k_i+j}, i\right) \ j = 1, \dots, 4$
- ბიჯი 2.5: $K_{i+1} = K_i + 5$

2.1 კომპიუტაციური ამოცანის დასმა და მისი ამოხსნის მეთოდების მიმოხილვა

ზოგად შემთხვევაში კომპიუტაციური ამოცანა შეიძლება ჩამოყალიბდეს როგორც გრაფში უმოკლესი სიგრძის ჰამილტონის ციკლის მოძებნის ამოცანა.

ვთქვათ გვაქვს $n + 1$ ქალაქი $0, 1, \dots, n$ და ქალაქების ყოველი i, j წყვილისთვის მოცემულია მათ შორის "მანძილი" – $C_{ij} \geq 0$. კომპიუტაციური იმყოფება 0 ქალაქში, შემოვიღოს ყველა ქალაქს და ბრუნდება 0 ქალაქში. უნდა ვიპოვოთ ქალაქების შემოვიღოს ისეთი მიმდევრობა, რომლისთვისაც ჯამური მანძილი იქნება მინიმალური. კომპიუტაციური სიმეტრიული ამოცანისთვის $C_{ij} = C_{ji}$.

განვიხილოთ ამოცანის კომბინატორული დასმა. მარშრუტის ქვეშ გვესმის $z = (0, i_1, i_2, \dots, i_n, 0)$ სადაც $i_1, i_2, \dots, i_n$ არის $1, 2, \dots, n$ რიცხვების გადანაცვლება. Z მარშრუტის სიგრძე

$$L(z) = \sum_{k=0}^n C_{i_k, i_{k+1}}$$

ვთქვათ Z არის ყველა მარშრუტის სიმრავლე მაშინ $|Z| = n!$. უნდა ვიპოვოთ ისეთი Z_0 რომ $L(z_0) = \min(L(z))$.

განვიხილოთ ამოცანის დასმა მთელრიცხვა პროგრამირების ტერმინებში. შემოვიღოთ ბულის ცვლადები :

$$x_{ij} = \begin{cases} 1, & \text{თუ კომპიუტაციური გადის } i\text{-ურ ქალაქს} \\ 0, & \text{წინააღმდეგ შემთხვევაში} \end{cases}$$

მაშინ მარშრუტის სიგრძე იქნება

$$\sum_{i=0}^n \sum_{j=0}^n C_{ij} x_{ij} \rightarrow \min$$

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

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

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

ასევე

$$x_{ij} \in \{0,1\}, \quad i, j = 0, \dots, n$$

ეს შეზღუდვები არ არის სრული და საჭიროებს ისეთი პირობების დამატებას, რომლებიც არ დაუშვებენ ამონახსნში რამდენიმე ციკლის არსებობას.

$$u_i - u_j + n x_{ij} \leq n - 1, i, j = 1, 2, \dots, n \quad i \neq j$$

2.1.1 ევრისტიული ალგორითმები კომივოიაჟერის ამოცანისთვის

ევრისტიული ალგორითმების ქვეშ გვესმის ალგორითმები რომლებიც ემყარებიან პრაქტიკულ მოსაზრებებს, ანუ ალგორითმები, რომელთათვისაც ფორმალური დასაბუთებები არ გვაქვს და რომლებიც ემყარებიან ამოცანის სტრუქტურის ანალიზს [3;5].

უახლოეს წერტილში გადასვლის ალგორითმი. ხშირად ამ ალგორითმს "უახლოესი მეზობლის ალგორითმსაც უწოდებენ". ვირჩევთ საწყის წერტილს და ვპოულობთ ამ წერტილის უახლოეს წერტილს, შემდეგ დარჩენილებიდან უახლოესს და ა.შ.

ვთქვათ $P = \{1, 2, \dots, n\}$ $C_{ij} \geq 0$ მანძილებია და $i \in P$ და $j \in P$ წერტილებს შორის.

i_1 იყოს საწყისი წერტილი და ვიპოვოთ i_2 წერტილი ისეთი რომ $C_{i_1, i_2} = \min C_{i_1, i}, i \neq i_1$.

ვთქვათ $Z_k = (i_1, i_2, \dots, i_k)$ ასაგები მარშრუტის აგებული ნაწილი. მაშინ i_{k+1} წერტილი განისაზღვრება შემდეგი პირობიდან

$$C_{i_k, i_{k+1}} = \min(C_{i_k, i}) \quad i \in P \setminus \{Z_k\} \quad \text{თუ } P \setminus \{Z_k + 1\} \neq \emptyset$$

მაშინ ვიპოვით i_{k+2} და ა.შ. მიღებული მარშრუტის სიგრძე განისაზღვრება

$$L(Z) = \sum_{k=1}^{n-1} C_{i_k, i_{k+1}} + C_{i_n, i_1}$$

გამოსახულებით. 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 მანძილების C_{ij} მატრიცას აქვს შემდეგი სახე:

$$\begin{cases} q, & \text{თუ } j = i + 1, i - 1, 2, \dots, n - 1; \\ A, & \text{თუ } j = i + 1, i - 1, 2, \dots, n - 1; \\ p, & \text{დანარჩენ შემთხვევაში} \end{cases}$$

სადაც $A > p > q$. C მატრიცა არის სიმეტრიული და დავუშვათ რომ მანძილებისთვის სრულდება სამკუთხედის უტოლობა. აღწერილი ალგორითმი იძლევა მარშრუტს $Z = (1, 2, \dots, n)$, $L(Z) = (n - 1)q + A$. A -ს ცვლილების ხარჯზე შესაძლებელია მარშრუტის სიგრძე გავხადოთ რაგინდ დიდი. Z_0 ოპტიმალურ ამონახსნს კი აქვს სახე $(1, 2, \dots, n - 2, n, n - 1)$ $Z_0 = (n - 3)q + 3p$, მაშინ

$$\frac{L(Z)}{L(Z_0)} = \frac{(n-1)q+A}{(n-3)q+3p} = \frac{(n-1)q+A}{nq+3(p-q)} < \frac{(n-1)q+A}{nq} \text{ სადაც } p - q > 0$$

$$\frac{L(Z)-L(Z_0)}{L(Z_0)} < \frac{(n-1)q+A}{nq} - 1 = \frac{A-q}{nq} \quad \text{ეს არის ზედა შეფასება.}$$

ქვედა შეფასებისთვის გვაქვს:

$$\frac{L(Z)}{L(Z_0)} = \frac{(n-1)q+A}{(n-3)q+3p} > \frac{(n-1)q+A}{(n-3)p+3p} = \frac{(n-1)q+A}{np} > \frac{nq-p+A}{np} \text{ საიდანაც გვაქვს}$$

$$\frac{L(Z)-L(Z_0)}{L(Z_0)} > \frac{nq-p+A}{np} - 1 = \frac{A-p}{np} - \frac{p-q}{p} = \frac{A-p}{np} - 1 + \frac{p}{q} > \frac{A-p}{np} - 1$$

$$\text{მივიღეთ რომ: } \frac{A-p}{np} - 1 < \frac{L(Z)-L(Z_0)}{L(Z_0)} < \frac{A-q}{np}$$

საიდანაც ჩანს, რომ გადახრა ოპტიმალური ამონახსნიდან უსასრულოდ იზრდება როცა $A \rightarrow \infty$ ფიქსირებული p და q რიცხვებისთვის. ალგორითმის სირთულე შეიძლება შეფასდეს როგორც $O(n^3)$.

2.1.2 მეორე ევრისტიული ალგორითმი, აირჩიე უმოკლესი რკალი.

ეს ალგორითმი მუშაობას იწყებს მინიმალური სიგრძის რკალის არჩევით

$$C_{i_1, i_2} = \min c_{ij} \text{ თუ ასეთი რამდენიმეა ვირჩევთ ნებისმიერს. ვთქვათ } Z_k = (i_1, i_2, \dots, i_k)$$

უკვე აგებული ნაწილია, მას ემატება ან (i_k, i_{k+1}) რკალი ან (i_{k+1}, i_1) რკალი.

ვთქვათ,

$$C_{i_k, i_{k+1}} = \min(C_{i_k, i})$$

$$C_{i_{k+1}, i_1} = \min(C_i, i_k), \quad i \in P \setminus \{Z_k\} \text{ თუ } C_{i_k, i_{k+1}} < C_{i_{k+1}, i_1}$$

მაშინ $Z_{k+1} = (i_1, i_2, \dots, i_k, i_{k+1})$. თუ არა და $Z_{k+1} = (i_{k+1}, i_1, \dots, i_k)$ ხელახლა

აღვნიშნოთ აგებული მარშრუტის ნაწილი $Z_{k+1} = (i_1, i_2, \dots, i_k, i_{k+1})$. თუ

$P \setminus \{Z_{k+1}\} \neq \emptyset$ გადავდივართ შემდეგ ბიჯზე. ალგორითმის სირთულე შეიძლება შეფასდეს ასე: $\min C_{ij}$ გამოთვლა საჭიროებს n^2 ოპერაციას, ერთი გადასვლის დამატება ნაწილობრივ აგებულ მარშრუტზე ითხოვს $2(n - k)$ ოპერაციას, ამიტომ ალგორითმის სირთულეა:

$$n^2 + 2 \sum_{k=1}^{n-1} (n - k) = O(n^2)$$

2.2 შტოების და საზღვრების მეთოდი კომპიოიაჟერის ამოცანისთვის

შტოების და საზღვრების მეთოდი ერთ-ერთი ყველაზე ეფექტური მეთოდია კომპიოიაჟერის ამოცანისთვის. მეთოდი შეიქმნა 1963 წელს ლიტლისა და თანაავტორების მიერ.

მეთოდის იდეა შემდეგში მდგომარეობს: კომპიოიაჟერის ამოცანის დასაშვებ ამონახსნთა მთელი სიმრავლისთვის (M_0) განისაზღვრება შეფასება ქვემოდან $\varphi(M_0)$. შემდეგ (M_0) დაიყოფა ორ თანაუკვეთ ქვესიმრავლედ, პირველი ქვესიმრავლე შედგება ისეთი ამონახსნებისგან რომლებიც შეიცავენ (i, j) რკალს. აღვნიშნოთ ეს სიმრავლე $\{(i, j)\}$ -ით. ხოლო, მეორე სიმრავლე შედგება ამონახსნებისგან, რომლებიც არ შეიცავენ (i, j) რკალს. აღვნიშნოთ ეს სიმრავლე $\{(i^*, j^*)\}$ -ით. თითოეული ამ სიმრავლისთვის დგინდება ქვედა საზღვრები $\varphi(i, j)$ და $\varphi(i^*, j^*)$. ყოველი ახალი შეფასება არ არის ნაკლები $\varphi(M_0)$ -ზე. ამ ორი სიმრავლიდან ირჩევენ იმას რომლის შეფასებაც ნაკლებია. ეს სიმრავლეც დაიყოფა ორ ქვესიმრავლედ და ახლად მიღებული ქვესიმრავლეებისთვისაც გამოითვლება ქვედა შეფასებები. დაყოფის პროცესი გრძელდება მანამ, სანამ არ მივიღებთ ერთადერთ ჰამილტონის ციკლს. ეს პროცესი შეიძლება წარმოვიდგინოთ დიხოტომიური ხის სახით, რომლის წვეროებსაც მიწერილი აქვთ შესაბამისი ქვესიმრავლის ქვედა შეფასება. ჰამილტონის ციკლის მიღების შემდეგ განიხილება გაწყვეტილი შტოები და ამ გაწყვეტილი შტოების შესაბამისი ქვესიმრავლეების შეფასებებს ვადარებთ მიღებული ჰამილტონის ციკლის სიგრძეს (იგივე რეკორდს Rec). თუ შეფასება ნაკლებია რეკორდზე მაშინ ხდება შტოების განვითარება ზემოთ მოყვანილი წესების შესაბამისად, პროცესი გრძელდება მანამ, სანამ ახლადწარმოქმნილი ქვესიმრავლეების შეფასებები ნაკლებია რეკორდზე. შტოების ასეთი განვითარების შედეგად შეიძლება გამოჩნდეს ჰამილტონის ახალი ციკლები და ხდება რეკორდის კორექტირება თუ რომელიმე ციკლის სიგრძე ნაკლები აღმოჩნდება რეკორდზე. ამოცანის ამოხსნა ითვლება დასრულებულად თუ გაწყვეტილი შტოების ქვედა საზღვრები არ

არის ნაკლები რეკორდზე. კომპიოთაჟერის ამოცანის ამოსახსნელად კი ვღებულობთ მარშრუტს, რომელსაც აქვს მინიმალური სიგრძე.

ქვედა შეფასებების გამოთვლა შეიძლება შემდეგ ფაქტზე დაყრდნობით: ნებისმიერ დასაშვებ მარშრუტში შედის ერთი ელემენტი მანძილების მატრიცის ყოველი სტრიქონიდან და ერთი ელემენტი მანძილების მატრიცის ყოველი სვეტიდან, ამიტომ თუ რომელიმე სტრიქონის ყოველ ელემენტს გამოვაკლებთ ერთსა და იმავე რიცხვს, ახალ მატრიცაში ყველა დასაშვები მარშრუტის სიგრძე შემცირდება ამ რიცხვით ძველ მატრიცასთან შედარებით. იგივე შეიძლება ითქვას სვეტების შესახებაც. ამიტომ, თუ გამოთვლებს ჩავატარებთ ყველა სტრიქონისა და ყველა სვეტის მიმართ, ოპტიმალური მარშრუტის სიგრძე გარდაქმნილი მატრიცისთვის განსხვავდება იგივე მარშრუტის სიგრძისგან საწყისი მატრიცისთვის იმ რიცხვების ჯამით, რომლებიც გამოვაკელით საწყისი მატრიცის სტრიქონებს და სვეტებს.

თუ აღვნიშნავთ i -ურ სტრიქონში მინიმალურ ელემენტს α_i -თი და j -ურ სვეტში მინიმალურ ელემენტს β_j -თი და ავიღებთ $C_{ij}^1 = C_{ij} - \alpha_i$ და შემდეგ $C_{ij}^2 = C_{ij}^1 - \beta_j$ გვექნება $C_{ij} = C_{ij}^2 + \alpha_i + \beta_j$.

α_i და β_j რიცხვებს ეწოდებათ დაყვანის მუდმივები. ხოლო C_{ij}^2 -ს დაყვანილი მატრიცის ელემენტები.

თუ L არის ოპტიმალური მარშრუტის სიგრძე, L_1 არის ოპტიმალური მარშრუტის სიგრძე დაყვანილ მატრიცაში. მაშინ $L = L_1 + \gamma$ სადაც $\gamma = \sum_i \alpha_i + \sum_j \beta_j$ დაყვანილ მატრიცაში ყველა ელემენტი არაუარყოფითია ამიტომ $L_1 \geq 0$ ყოველ სტრიქონში და ყოველ სვეტში არის ერთი ნულოვანი ელემენტი მაინც. γ რიცხვი შეიძლება ავიღოთ ქვედა შეფასებად, ანუ $\varphi(M0) = \gamma$.

განვიხილოთ იმ (i, j) რკალის ამორჩევის საკითხი, რომლის აპრიორული ჩასმა ან არ ჩასმა მარშრუტში დასაშვებ ამონახსნთან სიმრავლეს დაყოფს ორ ქვესიმრავლედ. (i, j) რკალის არ ჩასმა მარშრუტში ხორციელდება C_{ij} -ის მაგივრად დიდი რიცხვის ჩასმით (პირობითად ∞). შედეგად გვეძლევა საშუალება შევასრულოთ დამატებითი დაყვანა და გავაუმჯობესოთ ქვედა საზღვარი.

(i, j) რკალის ჩართვა მარშრუტში კი იძლევა მატრიცის განზომილებების შემცირების საშუალებას (i -ური სტრიქონისა და j -ური სვეტის ამოშლის ხარჯზე) ამ დროს არის საშიშროება n ზე ნაკლებგანზომილებიანი ციკლის შექმნისა, ამიტომ როგორც წესი იღებენ $a_{ji} = \infty$. ამის შემდეგ, ხორციელდება ახალი მატრიცის დაყვანა და უმჯობესდება $\{(i, j)\}$ სიმრავლის მარშრუტების ქვედა შეფასება.

დიდია ალბათობა იმისა, რომ ოპტიმალურ მარშრუტში აღმოჩნდებიან ის რკალები, რომლებიც დაყვანილ მატრიცაში ნულოვან ელემენტებს შეესაბამებიან, ამიტომ განშტოების რკალი ავირჩიოთ შემდეგი წესით: დაყვანილ მატრიცაში $a_{ij} = 0$ ელემენტები პირობითად

შევცვალოთ ∞ ით, შევკრიბოთ დაყვანის მუდმივები α_i და β_j და აღვნიშნოთ $\gamma(i^*, j^*) = \alpha_i + \beta_j$ და მარშრუტიდან ამოვავდოთ ის რკალი, რომლისთვისაც $\gamma(i^*, j^*)$ იქნება მაქსიმალური. ეს გამოიწვევს "შეფასების მკვეთრ ზრდას". ცხადია, რომ არ გამოყენების დანახარჯი ვერ იქნება ნაკლები i -ურ სტრიქონსა და k -ურ სვეტში მინიმალური ელემენტების ჯამზე. ვთქვათ, $\gamma(r^*, s^*) = \max\{\gamma(i^*, j^*)\}$, მაშინ თუ (r, s) რკალს ამოვავდებთ მარშრუტიდან, მივიღებთ $\{(r, s^*)\}$ ქვესიმრავლეს, ჩართვით კი მივიღებთ $\{(r, s)\}$ ქვესიმრავლეს.

ქვედა საზღვარი $\{(r^*, s^*)\}$ სიმრავლისთვის იქნება:

$$\varphi(r^*, s^*) = \varphi(M_0) + \gamma(r^*, s^*) = \gamma + \gamma(r^*, s^*);$$

ხოლო ქვედა საზღვარი $\{(r, s)\}$ სიმრავლისთვის:

$$\varphi(r, s) = \varphi(M_0) + \gamma(r, s) = \gamma + \gamma(r, s);$$

სადაც $\gamma(r, s)$ არის შემცირებული მატრიცის დაყვანის მუდმივების ჯამი.

პრაქტიკული გამოთვლებისას მარშრუტების სიმრავლის დაყოფის პროცესი გაგრძელდება 2×2 განზომილების მატრიცის მიღებამდე,

ჩამოვყალიბოთ ზემოთ განხილული ალგორითმის სახით:

ბიჯი1: მანძილების საწყისი მატრიცა დავიყვანოთ სტრიქონებისა და სვეტების მიმართ, ვიპოვოთ მარშრუტების სრული სიმრავლის (M_0) ქვედა შეფასება.

$$\varphi(M_0) = \gamma = \sum_{i=1}^n \alpha_i + \sum_{j=1}^n \beta_j$$

ბიჯი2: დაყვანილ მატრიცაში ყოველ ნულს პირობითად ვცვლით უსასრულოდ და ვპოულობთ დაყვანის მუდმივების ჯამს. $\gamma(i^*, j^*) = \alpha_i + \beta_j$.

ბიჯი3: მარშრუტიდან გამოვრიცხავთ იმ (i, j) რკალს, რომლისთვისაც დაყვანის მუდმივების ჯამი მაქსიმალურია.

ბიჯი4: დაგვყავს მიღებული მატრიცა და ვპოულობთ ქვედა საზღვარს მარშრუტების $\{(i^*, j^*)\}$

სიმრავლისთვის.

ბიჯი5: აპრიორულად ჩავრთოთ (i, j) რკალი მარშრუტში რასაც მივყავართ ბიჯი2 ში მიღებული მატრიციდან i ური სტრიქონისა და j ური სვეტის ამოგდებად a_{ji} ელემენტს ვცვლით უსასრულოდ, რომ თავიდან ავიცილოთ ჰამილტონის n ზე ნაკლებგანზომილებიანი ციკლის წარმოშობა.

ბიჯი6: დაგვყავს შემცირებული მატრიცა და გამოვითვლით $\{(i, j)\}$ სიმრავლის ქვედა $\varphi(i, j)$ შეფასებას.

ბიჯი7: ვამოწმებთ შემცირებული მატრიცის განზომილებას, თუ განზომილება 2×2 -ია გადავდივართ ბიჯი9 ზე, თუ არა და გადავდივართ ბიჯი 8 ზე;

ბიჯი8: ვადარებთ დაყოფის შედეგად მიღებული ქვესიმრავლების ქვედა შეფასებებს $\varphi(i, j)$ და $\varphi(i^*, j^*)$ და გადავდივართ ბიჯი 2 ის შესრულებაზე ამასთან თუ $\varphi(i^*, j^*) < \varphi(i, j)$ დაყოფას ექვემდებარება $\{(i^*, j^*)\}$ ქვესიმრავლე. თუ $\varphi(i, j) < \varphi(i^*, j^*)$ მაშინ შემდგომ დაყოფას ექვემდებარება $\{(i, j)\}$ ქვესიმრავლე.

სიმრავლის დაყოფის პროცედურას სასურველია მივცეთ ორობითი ხის სახე.

ბიჯი9: ვიპოვოთ ჰამილტონის ციკლი და მისი სიგრძე (Rec)

ბიჯი10: შევადაროთ მიღებული ციკლის სიგრძე გაწყვეტილი შტოების ქვედა საზღვრებს. თუ მიღებული ამონახსნის სიგრძე არ აღემატება ქვედა საზღვრებს, ამოცანა ამოხსნილია, თუ მარშრუტის სიგრძე მეტია, რომელიმე შეფასებაზე, ალგორითმის მიხედვით ვახდენთ ამ მიმართულებით დაყოფას, სანამ არ მივიღებთ მარშრუტს ნაკლები სიგრძით ან არ დავრწმუნდებით რომ ასეთი არ არსებობს.

§3 უმცირესი დაყოფის ამოცანა და ამოხსნის ალგორითმი

3.1 ამოცანის ფორმულირება

მოცემულია სიმრავლე $R = \{r_1, \dots, r_M\}$, და ქვესიმრავლე $\varphi = \{S_1, \dots, S_N\}$ სიმრავლისა $S_j \subset R$ ნებისმიერი ქვესიმრავლე $\varphi' = \{S_{j_1}, S_{j_2}, \dots, S_{j_k}\}$ ისეთი, რომ

$$\bigcup_{i=1}^k S_{j_i} = R$$

ამას ეძახიან სიმრავლის დაფარვას, ხოლო S_{j_i} -ს დამფარავ სიმრავლეს. ხოლო თუ φ' აგრეთვე აკმაყოფილებს პირობებს

$$S_{j_h} \cap S_{j_l} = \emptyset, \quad \forall h, l \in \{1, \dots, k\}, \quad h \neq l$$

ამ შემთხვევაში φ' -ს ეწოდება R -ის დამყოფი სიმრავლე.

თუ თითოეული $S_j \in \varphi$ აქვს (დადებითი) წონა c_j , მაშინ სიმრავლის დაფარვის ამოცანა განისაზღვრება შემდეგნაირად: ვიპოვიოთ დამფარავი სიმრავლე მინიმალური წონით.

$\varphi' = \{S_{j_1}, \dots, S_{j_k}\}$ –ს წონა იქნება $\sum_{i=1}^k c_{j_i}$. ხის დაყოფის ამოცანაც განიმარტება შესაბამისად.

0-1 მატრიცა M სტრიქონით და N სვეტით რომელიც იფარება სვეტებით, შეიძლება წარმოდგინდეს 0-1 $[t_{ij}]$ წრფივი პროგრამირებით.

მინიმიზაცია:

$$z = \sum_{j=1}^N c_j \xi_j$$

იმის გათვალისწინებით რომ:

$$\sum_{j=1}^N t_{ij} \xi_j \geq 1, \quad i = 1, 2, \dots, M.$$

სადაც:

$$\begin{aligned} c_j &\geq 0 \\ \xi_j &= \begin{cases} 1, & S_j \in \varphi' \\ 0, & S_j \notin \varphi' \end{cases} \\ t_{ij} &= \begin{cases} 1, & r_i \in S_j \\ 0, & r_i \notin S_j \end{cases} \end{aligned}$$

დაყოფის ამოცანაში შეზღუდვა შეიცვლება

$$\sum_{j=1}^N t_{ij} \xi_j = 1, \quad i = 1, 2, \dots, M.$$

3.2 ამოცანის გამარტივება

ამოცანის გამარტივების რამდენიმე გზა არსებობს

- 1) თუ რომელიმე $r_i \in R$, $r_i \notin S_j$, $\forall j = 1, \dots, N$ მაშინ r_i ვერ დაიფარება და ამოცანას ამონახსნი არ გააჩნია.

- 2) თუ $\exists r_i \in R$ ისეთი, რომ $r_i \in S_k$ და $r_i \notin S_j, \forall j \neq k$, მაშინ S_k იქნება ყველა ამონახსენში და ამოცანა გამარტივდება შემდეგნაირად $R = R - \{r_i\}$ და $\varphi = \varphi - \{S_k\}$.
- 3) ვთქვათ $V_i = \{j \mid r_i \in S_j\}$. მაშინ, თუ $\exists p, q \in \{1, \dots, M\}$ ისე, რომ $V_p \subseteq V_q$, r_q შეიძლება წაიშალოს R -დან რადგან, ყოველი სვეტი რომელიც დაფარავს r_p , ასევე დაფარავს r_q . ე.ი r_p დომინირებს r_q -ზე.
- 4) თუ რომელიმე ქვესიმრავლე $\bar{\varphi} \subset \varphi$, $\bigcup_{S_j \in \bar{\varphi}} S_j \supseteq S_k$ და $\sum_{S_j \in \bar{\varphi}} C_j \leq C_k$ ნებისმიერი $S_k \in \varphi - \bar{\varphi}$, მაშინ S_k უნდა წაიშალოს φ -დან, რადგან მასზე დომინირებს $\bigcup_{S_j \in \bar{\varphi}} S_j$.

3.3 ხის ძებნის ალგორითმი დაყოფის ამოცანისთვის

თავდაპირველად, მატრიცის სვეტები იყოფა ბლოკებად. K - ური ბლოკი შეიცავს ისეთ სვეტებს φ —დან რომელიც მოიცავს r_k — ურ ელემენტს, მაგრამ არ შეიცავს არცერთ უფრო დაბალი რიგის მქონე ელემენტს $r_1, \dots, r_{k-1}$. თითოეული სვეტი შესაბამისად მოხვდება ერთ ბლოკში და თითოეული ბლოკი დაემსგავსება ცხრილში მოყვანილ მატრიცას.

	ბლოკი 1	ბლოკი 2	ბლოკი 3	ბლოკი 4	ბლოკი 5	ბლოკიN
r_1	11111	0	0	0	0	
r_2	0 ∨ 1	11111				
r_3		0 ∨ 1	0 ∨ 1	11111		
r_4				11111		
r_5				0 ∨ 1	11111	
r_m				0 ∨ 1	ა.შ.	

ცხრილი 1.

საგულისხმოა ის ფაქტი, რომ ბლოკები დალაგებული არის ზრდადობის მიხედვით. არ დავიწყებთ k - ური ბლოკის ელემენტის დაფარვას მანამ სანამ, არ იქნება დაფარული $r_i, 1 \leq i \leq k - 1$

თითოეული სვეტი ბლოკში დალაგებულია წონების ზრდის მიხედვით.

ვთქვათ, რომ:

- B — სავარაუდოდ საუკეთესო ამონახსენი.
- $\bar{B}$ — საუკეთესო ამონახსენი.
- z — სავარაუდოდ საუკეთესო ფასი.
- $\bar{z}$ — საუკეთესო ფასი.
- E — დაფარული სტრიქონები.

3.3.1 ალგორითმი

ბიჯი 1: $B = \emptyset, \bar{B} = \emptyset, E = \emptyset, z = 0, \bar{z} = \infty$

ბიჯი 2: ვიპოვოთ $p = \min[i \mid r_i \notin E]$. მოვნიშნოთ p -ურ ბლოკში სვეტი (მინიმალური წონით).

ბიჯი 3: დავიწყოთ p ბლოკის ელემენტების გამოკვლევა მონიშნული სვეტიდან S_j^P (j -ს ზრდადობით).

I. თუ S_j^p ვიპოვნეთ ისეთი, რომ $S_j^p \cap E = \emptyset$ და $z + c_j^p < \bar{z}$ (სადაც c_j^p არის S_j^p -ს წონა), გადავდივართ ბიჯი 5:-ზე.

II. წინააღმდეგ შემთხვევაში, თუ p ბლოკის ელემენტები ამოვწურეთ, ან მივაღწიეთ ისეთ S_j^p -ს, $z + c_j^p \geq \bar{z}$ რომ მაშინ გადავდივართ ბიჯი 4:-ზე.

ბიჯი 4: B -ს არ შეუძლია მიგვიყვანოს საუკეთესო ამონახსნამდე.

I. თუ (ე.ი. ბლოკი 1 ამოვწურეთ) შევჩერდეთ საუკეთესო ამონახსნს $\bar{B}$ -ზე.

II. წინააღმდეგ შემთხვევაში, B -დან წავშალოთ ბოლოს დამატებული $S_k^l, p = l$. მოვნიშნოთ S_{k+1}^p . გადავიდეთ ბიჯი 3:-ზე.

ბიჯი 5: $B = B \cup \{S_j^p\}; E = E \cup S_j^p; z = z + c_j^p$.

I. თუ $E = R$, საუკეთესო მარშრუტი მოიძებნა $\bar{B} = B; \bar{z} = z; B = \emptyset$ გადავიდეთ ბიჯი 4:-ზე.

II. წინააღმდეგ შემთხვევაში გადავიდეთ ბიჯი 2:-ზე.

მაგალითი

განვიხილოთ ცნობილი 25 პუნქტიანი ამოცანა, სადაც 1 საწყობია და 24 დანიშნულების პუნქტი რომლებსაც პროდუქტი უნდა მიეწოდოს. შემომავალი ინფორმაციაა:

1. სატრანსპორტო საშუალების ტვირთამწეობა: 100 ერთ.
2. სატრანსპორტო საშუალების მაქსიმალური გარბენი: 1000
3. 24 პუნქტისათვის პროდუქტის მოხოვნები:

17	12	10	13	16	11	21	12	18	11	20	17	18	13	17	17	15	17	24	12	15	12	21	24
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

4. მანძილების მატრიცა:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	182	86	129	211	185	185	41	171	106	18	73	186	71	26	233	288	212	201	183	290	208	93	373	453
1	182	0	107	142	195	22	50	144	85	165	200	109	167	114	208	128	231	36	144	126	231	45	272	305	396
2	86	107	0	167	249	99	99	58	88	151	104	26	100	50	112	208	252	126	167	162	264	122	179	308	417
3	129	142	167	0	82	164	115	163	220	23	146	148	267	124	136	270	341	178	254	236	343	187	158	426	506
4	211	195	249	82	0	217	175	245	280	105	228	230	349	206	218	323	423	231	336	318	425	240	240	500	588
5	185	22	99	164	217	0	72	144	63	187	203	125	145	114	211	119	209	27	122	104	211	23	272	283	374
6	185	50	99	115	175	72	0	144	127	138	203	117	199	114	211	167	271	75	184	166	270	85	272	355	436
7	41	144	58	163	245	144	144	0	146	140	59	46	158	39	67	201	256	171	179	151	258	167	134	341	421
8	171	85	88	220	280	63	127	146	0	237	192	112	82	136	200	120	178	88	79	95	180	65	267	240	343
9	106	165	151	23	105	187	138	140	237	0	123	125	251	113	113	275	330	201	243	225	332	210	135	415	495
10	18	200	104	146	228	203	203	59	192	123	0	91	204	89	22	251	306	230	219	201	308	226	87	391	471
11	73	109	26	148	230	125	117	46	112	125	91	0	126	24	99	186	241	145	154	136	243	148	166	326	406
12	186	167	100	267	349	145	199	158	82	251	204	126	0	150	212	158	189	170	96	114	187	147	279	208	321
13	71	114	50	124	206	114	114	39	136	113	89	24	150	0	97	162	217	141	130	112	219	137	158	302	382
14	26	208	112	136	218	211	211	67	200	113	22	99	212	97	0	259	314	238	227	209	316	234	67	399	479
15	233	128	208	270	323	119	167	201	120	275	251	186	158	162	259	0	114	92	68	50	103	108	320	206	279
16	288	231	252	341	423	209	271	256	178	330	306	241	189	217	314	114	0	203	99	105	36	186	375	94	165
17	212	36	126	178	231	27	75	171	88	201	230	145	170	141	238	92	203	0	116	98	195	23	299	288	368
18	201	144	167	254	336	122	184	179	79	243	219	154	96	130	227	68	99	116	0	18	101	99	288	193	264
19	183	126	162	236	318	104	166	151	95	225	201	136	114	112	209	50	105	98	18	0	107	81	270	190	270
20	290	231	264	343	425	211	270	258	180	332	308	243	187	219	316	103	36	195	101	107	0	188	377	128	195
21	208	45	122	187	240	23	85	167	65	210	226	148	147	137	234	108	186	23	99	81	188	0	295	271	351
22	93	272	179	158	240	272	272	134	267	135	87	166	279	158	67	320	375	299	288	270	377	295	0	460	540
23	373	305	308	426	500	283	355	341	240	415	391	326	208	302	399	206	94	288	193	190	128	271	460	0	113
24	453	396	417	506	588	374	436	421	343	495	471	406	321	382	479	279	165	368	264	270	195	351	540	113	0

გამომავალი ინფორმაცია კი შეიცავს შემდეგ ინფორმაციას:

1. ჯამური განარბენი: 3657
2. მარშრუტები და მათი განარბენი:

განარბენი	პუნქტები								
656	0	3	4	6	1	5	2	7	0
541	0	13	11	8	12	14	10	0	
801	0	9	17	15	16	19	0		
688	0	22	21	18	0				
971	0	20	24	23	0				

დასკვნა

ნაშრომში წარმოდგენილი სატრანსპორტო მარშრუტიზაციის ამოცანის ამოხსნის ეს მიდგომა გამორჩეულია იმით, რომ მარშრუტებზე დადებული ისეთი შეზღუდვები როგორიცაა სატრანსპორტო საშუალების ტვირთამწეობა და განარბენის სიგრძე გათვალისწინებული ხდება ე.წ. „იმედისმომცემი“ მარშრუტების აგების პროცესში. შემდეგ ხდება თითოეული მარშრუტის ოპტიმიზირება და ბოლოს დაყოფის ამოცანის ამოხსნის გზით ოპტიმალური მარშრუტების (დაყოფის ამოცანის თვალსაზრისით) ამორჩევა. მეთოდის ეფექტურობა დამოკიდებულია იმაზე, თუ რამდენად „სრულად“ იქნება აგებული დასაშვებ მარშრუტთა სიმრავლე, რაც თავის მხრივ პირდაპირ კავშირშია დაყოფის ამოცანის ამოხსნასთან. შექმნილია პროგრამათა კომპლექსი, რომელიც ხსნის დასმულ ამოცანას. მეთოდი შეიძლება გამოყენებული იქნას მარშრუტიზაციის პრაქტიკული ამოცანების ამოხსნისას, როცა საქმე გვაქვს ერთგვაროვანი ტვირთის მომხმარებელთათვის მიწოდებისას რაციონალური მარშრუტების აგების პრობლემასთან.

ლიტერატურა

1. N. Christofides. Graph theory / an algorithmic approach. ACADEMIC PRESS INC. London 1975.
2. Toth P., Vigo D. An overview of vehicle routing problems// The Vehicle Routing Problem: SIAM Monograph on Discrete Mathematics and Applications. Philadelphia 2002 p.1-24.
3. Thomas H. Cormen. Introduction to Algorithms. Third Edition. The MIT Press Cambridge, Massachusetts London, England. 2009.
4. Гваберидзе Б, Мачаидзе З, Цинцадзе З, О приближенном методе решения одного класса задач маршрутизации. Проблемы управления макросистемами М 1985 7-11
5. Пападимитриу Х., Стейглиц К. Комбинаторная оптимизация. Алгоритмы и сложность, М. Мир, 1984.