MOKSLAS – LIETUVOS ATEITIS
SCIENCE – FUTURE OF LITHUANIA
ISSN 2029-2341 / eISSN 2029-2252
http://www.mla.vgtu.lt

STATYBA, TRANSPORTAS, AVIACINĖS TECHNOLOGIJOS
CIVIL AND TRANSPORT ENGINEERING, AVIATION TECHNOLOGIES

2014 6(5): 546–551

Transportas
Transport Engineering

http://dx.doi.org/10.3846/mla.2014.698

OPTIMALAUS MARŠRUTO NUSTATYMAS TAIKANT BELMANO IR FORDO METODĄ

Laima GREIČIŪNĖ

Vilniaus Gedimino technikos universiteto Intermodalinio transporto ir logistikos kompetencijos centras
El. paštas laima.greiciune@vgtu.lt

Santrauka. Šio straipsnio tikslas – aprašyti, kaip dinaminio programavimo metodas pritaikomas optimaliam maršrutui nustatyti naudojant Lietuvos žaliojo koridoriaus, kuris išsiskiria mažesne aplinkos tarša vežant įvairiarūšius krovinius, topologinio žemėlapio ir intelektinių transporto sistemų (ITS) įrenginių duomenis realiu laiku. Taikant *Visual basic application* (VBA) programavimo kalbos kodą sprendžiant dinaminio programavimo Belmano ir Fordo metodu, kodas skirtas optimaliam maršrutui nustatyti, atsižvelgiama į tris optimalumo kriterijus (laiką, atstumą bei teršalų emisijos kiekį).

Reikšminiai žodžiai: optimalus maršrutas, Belmano ir Fordo lygtis, žaliųjų transporto koridorių tinklo žemėlapis, ITS, VBA kodas.

Įvadas

Šalies ekonomikos integravimosi į ES rinką sąlygomis, vykstant esminiams ekonomikos struktūriniams ir užsienio prekybos pokyčiams, reikia įvertinti laiko įtaką, plėtojant transportą ir tranzitą. Užtikrinti nepertraukiamą krovinių vežimą, palaikant dinamišką šalies ekonomikos plėtotės raidą.

Transporto koridorių sąvoką apibūdina krovininio transporto koncentracija tarp pagrindinių transporto mazgų ir važiuojant palyginti tolimais atstumais. Sprendžiant didėjančio eismo problemą ir kartu užtikrinant aplinkos tvarumą bei efektyvų energijos naudojimą, reikia skatinti šiuose koridoriuose vežant krovinius įvairių rūšių transportu optimalų naudojimą ir pažangių technologijų taikymą. Išsprendus problemas, kurias sukelia spūstys, tarša ir triukšmas, CO_2 išmetamosios dujos (logistikos sektoriaus efektyvumas sumažėtų, jei šios problemos nebūtų išspręstos), padidėtų krovininio transporto logistikos sektoriaus ilgalaikis efektyvumas ir augimas.

Tam, kad įvairių rūšių transportas būtų naudojamas optimaliai, būtina užtikrinti naudojimąsi transporto koridoriais, tačiau koridoriai turi turėti pakankamą koridoriaus pajėgumą, kad vyktų sklandus judėjimas įvairiais maršrutais.

Optimaliam maršrutui nustatyti naudojamas įvairių klasikinių nelygybių variacijų formulavimas, pagrįstas dinamine vartotojo kelionės pusiausvyra, lygiaverte matematine išraiška, remiantis keliautojo sprendimų pasirinkimu ir pagrįstas Belmano ir Fordo optimalumo principu, taikant Belmano ir Fordo lygtis (Meimbresse, Lipinski 2007).

Maršruto pasirinkimo modeliavimą nagrinėjantis Prato (2009) rėmėsi mokslininkų Bellman (1958) ir Dijkstra (1959) dinaminio programavimo metodais, nes nuo 1958 m. Bellman (1958), Dijkstra (1959) bei Dantzig (1960) sprendė uždavinius, pasitelkdami optimalaus maršruto nustatymo metodą.

Dinaminio programavimo modelį galima taikyti tiekimo grandinėje, kurioje laiko užimtumas kontroliuojamas. Nors kontrolės sistemos projektuotojas negali kontroliuoti užimto laiko, svarbu modeliuoti vėlavimus (Sarimveisa *et al.* 2008).

Idėją, kad optimalų valdymą kiekviename žingsnyje reikia parinkti, atsižvelgiant į jo įtaką tolesnei proceso eigai, yra iškėlęs Bellmanas (1958).

Daugelis maršrutų sudarymo metodų peremti pasikartojančių trumpiausių kelių tinkle paieška (Huang, Gao 2012), kur optimalūs maršrutai apskaičiuojami pagal vieną ar daugiau įvesties kintamųjų, tokių kaip jungties nepriklausomumas, maršrutų apribojimai ir paieškos kriterijai.

Optimalaus maršruto nustatymas, taikant dinaminio programavimo metodą

Aprašomuosiuose moksliniuose leidiniuose dinaminio programavimo metodai dažniausiai taikomi skaičiuojant finansines išlaidas. Tačiau naudojant Belmano ir Fordo metodą optimizuojami kriterijai gali būti nustatomi pagal sprendžiamo uždavinio sąlygas.

Optimalusis valdymas kiekviename žingsnyje nusakomas sistemos būviu to žingsnio pradžioje ir valdymo tikslu, t. y. optimalusis valdymas turi tokią savybę, kad neatsižvelgus į pradinį būvį bet kuriame žingsnyje ir į tame žingsnyje parinktą valdymą, tolesnis valdymas turi būti parenkamas optimalumo atžvilgiu, prie kurio sistema prieis to žingsnio pabaigoje. Taikant šį principą, gaunama, kad kiekviename žingsnyje parinktas valdymas bus viso proceso geriausias valdymas. Dalijant uždavinį į žingsnius ir taikant optimalumo kriterijų, galima sukurti tam tikras funkcines lygtis.

Matematinė optimalumo principo išraiška F_k, kai k-tajame žingsnyje valdymas $\bar{u}_k$ būvį $\bar{u}_{k-1}$ pakeičia būviu $\bar{x}_k = Y_k\left(\bar{x}_{k-1},\ \bar{u}_k\right)$. Tolesni valdymai $\bar{u}_{k+1}, ..., \bar{u}_n$ turi būti parenkami optimalaus būvio $\bar{x}_k$ atžvilgiu, nes efektyvumo rodiklis minimizuojamas tolesniuose žingsniuose iki proceso galo, pažymint rodiklį, kuris nusako suminį efektyvumą Z_k nuo k-tojo iki n-tojo žingsnio:

$$Z_k = \sum_{i=k}^{n} f_i\left(\bar{x}_{i-1},\ \bar{u}_i\right). \tag{1}$$

Išrinkę optimalųjį valdymą $u_k^* = \left(\bar{u}_k^*, ..., \bar{u}_n^*\right)$ paskutiniuose žingsniuose, pradėdami k-tuoju, gauname optimalią Z_k reikšmę, t. y. dydį:

$$F_k = \min Z_k\ , \tag{2}$$

kuris priklausys tik nuo $\bar{x}_{k-1}$, t. y. $F_k = F_k\left(\bar{x}_{k-1}\right)$.

Pagal optimalumo principą, neatsižvelgiant į valdymo funkciją $\bar{u}_k$, norint gauti $\bar{x}_k$, kiti valdymai $\bar{u}_{k+1},...,\bar{u}_n$ turi būti tokie, kad efektyvumo rodiklis Z_{k+1} būtų minimalus, t. y. lygus $F_{k+1}\left(\bar{x}_k\right)$. Kadangi ši reikšmė priklauso nuo $\bar{x}_k$, o $\bar{x}_k$ priklauso nuo $\bar{u}_k$, tai $\bar{u}_k$ reikia parinkti taip, kad, naudojant jį kartu su optimaliaisiais valdymais kituose žingsniuose, pradedant $(k+1)$, visų žingsnių, pradedant k-tuoju, bendras efektyvumo rodiklis būtų minimalus. Tai išreiškiama formule:

$$F_k\left(\bar{x}_{k-1}\right) = \min_{\bar{u}_k \in \Omega_k} \left\{ f_k\left(\bar{x}_{k-1}, \bar{u}_k\right) + F_{k+1}\left(\bar{x}_k\right) \right\},$$
$$k = 1,\ 2,\ ...,\ n-1\ , \tag{3}$$

$$F_n\left(\bar{x}_{n-1}\right) = \min_{\bar{u}_n \in \Omega_n} f_n\left(\bar{x}_{n-1}, \bar{u}_n\right). \tag{4}$$

Šios lygtys yra rekurentinės lygtys, nes norint gauti $F_{n-1}\left(\bar{x}_{n-2}\right)$, reikia žinoti $F_{n-1}\left(\bar{x}_{n-1}\right)$, norint gauti $F_{n-2}\left(\bar{x}_{n-3}\right)$, reikia žinoti $F_{n-1}\left(\bar{x}_{n-2}\right)$ ir t.t., kol gaunama reikšmė $F_1\left(\bar{x}_0\right) = \min_{u \in X} z$, t. y. gaunama optimali tikslo funkcijos Z reikšmė.

Iš (4) lygties randame $\bar{u}_k^*$, kuri dešinėje pusėje esančiam reiškiniui suteikia minimalią reikšmę, priklausančią nuo $\bar{x}_{k-1}$, ir vadinama *sąlyginiu optimaliuoju valdymu* k-tajame žingsnyje.

Belmano ir Fordo lygtis išreiškia dinaminio programavimo esmę. Užuot ieškant minimalumo funkcijos, priklausančios nuo n kintamųjų $\bar{u}_1, \ldots, \bar{u}_n$, sprendžiama n uždavinių. Kiekviename iš tų uždavinių minimizuojama vieno kintamojo $\bar{u}_k$ funkcija. Tačiau šie uždaviniai yra susiję, nes, sprendžiant kitą uždavinį, imama ankstesniame žingsnyje gauta funkcijos reikšmė. Dinaminio programavimo uždavinio esmė: sudėtingas optimizavimo uždavinys skaidomas į seką paprastesnių uždavinių, kuriuos išsprendus, lengvai apskaičiuojamas pradinio uždavinio sprendinys. Nagrinėjant dinaminį uždavinį, turima omenyje koks nors procesas, priklausantis nuo laiko. Tačiau dinaminio programavimo metodu sprendžiami ir uždaviniai, kuriuose laiko sąvokos nėra, bet juos galima nagrinėti kaip daugiažingsnius uždavinius.

Kaip matyti iš optimalumo principo bei iš Belmano ir Fordo lygčių, kiekviename žingsnyje sprendimas (valdymas) priimamas atsižvelgiant į ateityje galimus padarinius. Išimtį sudaro paskutinis žingsnis, kuriuo procesas baigiamas. Todėl valdymą galima planuoti, atsižvelgiant tik į tai, kad šiame etape jis būtų optimalusis. Suplanavus paskutinį žingsnį, planuojamas antras nuo galo žingsnis taip, kad abiejų žingsnių rezultatas būtų optimalus ir t. t. Čia ir pasireiškia viena iš dinaminio metodo ypatybių – sprendžiant uždavinį, einama nuo galo į pradžią.

Sprendžiant įvairiarūšių krovinių srautų optimizavimo užduotis žaliuosiuose transporto koridoriuose, reikia supaprastinti optimalių (trumpiausių) maršrutų nustatymą įtraukiant realaus laiko duomenis keliuose iš intelektinių transporto sistemų (ITS) įrenginių, t. y. kelių oro sąlygų stotelių ir eismo intensyvumo skaitiklių (informacija apie kelių oro sąlygas, apribojimus keliuose, eismo intensyvumo duomenis), tai leistų greitai apskaičiuoti optimalų maršrutą (atsižvelgiant į nuolatos atnaujinamą informaciją ir pateikiant ją vartotojui).

Topologinės struktūros sintezė

Susiejant kelių oro sąlygų stotelių ir eismo intensyvumo skaitiklių duomenis, vartotojai gauna informaciją tiek prieš kelionę, tiek kelionės metu. Sistemos vartotojai daug lengviau gali nuspręsti, kokiu maršrutu vežti krovinį, kokią transporto rūšį pasirinkti. Duomenų surinkimas iš kelių oro sąlygų stotelių ir eismo intensyvumo skaitiklių vykdomas reguliariai kas 15 min., dėl to, atnaujinant duomenis optimalaus maršruto parinkimo uždavinyje, vartotojai gali būti įspėti apie galimas kliūtis, apribojimus maršrute ir gali būti pasiūlomas alternatyvus maršrutas iki paskirties taško.

Tyrimui taikomas geografinės informacinės sistemos (GIS) ir dinaminio programavimo metodas, sudarytas Lietuvą kertančių žaliųjų transporto koridorių tinklo pagrindu (žr. pav.), t. y. tarptautiniai transporto koridorių keliai, kurie kerta Lietuvos Respublikos teritoriją.

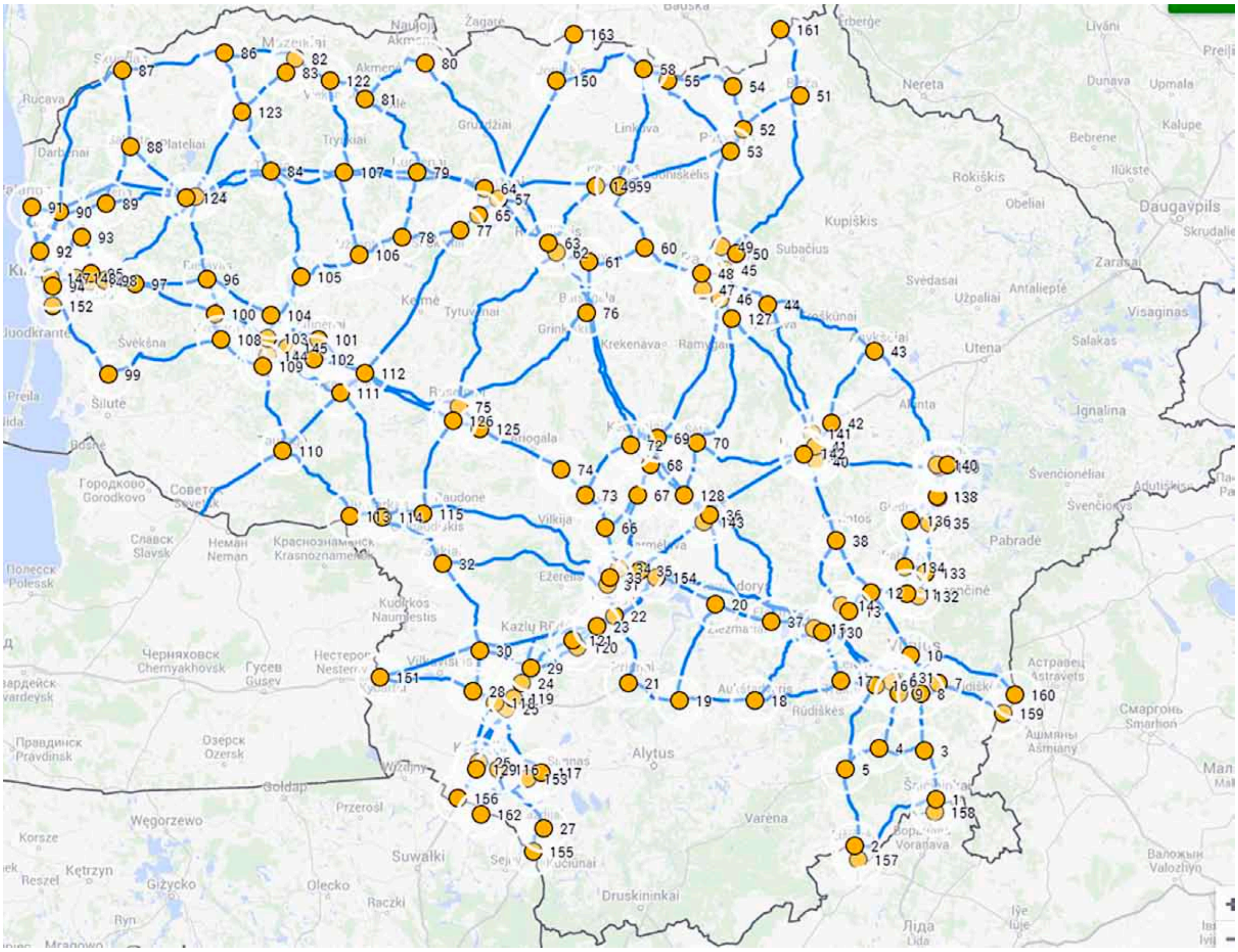
Sudarytas transporto tinklas $G(V, A)$, kuriame yra viršūnių aibė V ir briaunų aibė A. Terminalai transporto tinkle yra atitinkami geografiniai punktai, įtraukti į jį kaip viršūnės, į kuriuos reikia atvežti vienus krovinius ir išvežti kitus. Būtina sutapatinti tinklo $G(V, A)$ terminalus, teigiant, kad kroviniai bet kuria briauna $(i, j) \in A$ gali būti vežami visomis arba kai kuriomis tp (transporto rūšimis).

Maršruto grandinė išreiškiama briaunų seka (V_0, V_1), (V_1, V_2), ..., (V_{n-1}, V_n). Norint apibrėžti grandinę, pakanka eilės tvarka nurodyti viršūnes, per kurias eina grandinė. Maršrutai, kurių pirma viršūnė sutampa su paskutine, vadinami ciklais.

Suformavus koridorių tinklą, buvo priimta sąlyga uždaviniui, kad kelių tinklas, esantis aplink koridorius, turi būti sujungtas kaip potenciali galimybė aplinkkeliui dėl susidariusios kliūties pagrindiniame koridoriuje. Šiame tinkle identifikuotos $V = 163$ (viršūnės) ir $A = 488$ (briaunos) ir jose veikia $tp = 2$ (transporto rūšys).

Formuojant transporto koridorių tinklą buvo panaudotos GPS VIZUALIZER galimybės, kur tyrimo metu visa žemėlapio duomenų bazė buvo sudaryta iš koordinačių sistemų. Tai leido, pritaikius koordinačių sistemą, apskaičiuoti atstumus tarp identifikuotų susikirtimo taškų, t. y. briaunų ilgius. Šie duomenys naudojami, taikant dinaminio programavimo metodą, kaip rodikliai, kurie leidžia apskaičiuoti optimalų maršrutą transporto tinkle, o atsiradus poreikiui, pakeisti maršrutą ir pasinaudoti aplinkkeliais dėl susidariusios kliūties pagrindiniame transporto koridoriuje.

Tyrime taip pat atsižvelgta į kelių transporto priemonės greičio priklausomybę nuo kelių oro sąlygų. Šie kintamieji naudojami vidutiniam greičiui v_{vid} briaunoje įvardinti. Gauti optimalaus maršruto duomenys buvo per-



Pav. GIS pagrindu sudarytas Lietuvos žaliųjų transporto koridorių tinklas $G(V, A)$ su galimais aplinkkeliais (sudaryta autorės)

Fig. GIS-based green transport corridor network $G(V, A)$ with possible detours in Lithuania kelti į optimalaus maršruto modelį, kuris yra sudarytas naudojant VBA kodą. Norint nustatyti optimalų maršrutą žaliajame transporto koridoriuje, reikia atsižvelgti į rodiklius, kurie optimizuojami, šiuo atveju pasirinktas optimalus maršrutas pagal trumpiausią laiką t_{ij}, atstumą L_{ij} ir mažiausią emisijos E_{ij} kiekį. Dėl to briaunų svoris buvo jos ilgis (L), km, laikas (t), h, per kurį įveikiama briauna L priklausomai nuo realaus laiko informacijos apie esamas kelių oro ir eismo sąlygas. Emisijos kiekis briaunoje nustatytas pagal briaunos ilgio ir vieno kilometro g/TEU emisijos kiekį.

Dėl nuolatos didėjančios krovinių transporto paslaugų paklausos didėja ir teršalų emisijos kiekis. Dinaminio programavimo uždavinyje CO_2 rodikliai buvo naudojami iš Winebrake *et al*. (2007), Kindberg (2013), ECOtransIT (EcoTransIT World 2014) pateikiamų duomenų, nes nėra vienos teršalų emisijos kiekių nustatymo metodikos bei dydžių, todėl naudojami anksčiau minėtų šaltinių CO_2 emisijos duomenys (jų kiekis išreikštas g/TEUkm).

3. Belmano ir Fordo uždavinio sprendimas

Tyrimo metu Belmano ir Fordo dinaminio programavimo uždavinio optimaliam sprendimui buvo naudojama VBA kalba MS *Excel* programoje. *Visual Basic Application* (VBA) metodas – tai objektinio programavimo kalba. Kalba valdo objektus, kuriems atliekami įvairūs veiksmai. Pasikeitus programavimo aplinkai, atsirado galimybė stebėti projekto vykdymą, projektuoti priedus ir visas programos kodas paskirstytas į procedūras (paprogrames), kurios redaguojamos ir iškviečiamos atskirai.

Optimalaus maršruto VBA kodas:

```
Option Explicit
Private Type edge
   a As Long `iš
   b As Long `į
   cost As Double `briaunos svoris
End Type

Private Const INF As Double = 1E+100 `begalybės žymėjimas

Function FordBellman(rng As Range, s1&, s2&, Optional edgeCost As Long = 3) `funkcijos išraiška, grįžtamasis masyvas
   `rng – grafas, briaunų aprašymas
   `s1 – pradžios viršūnė
   `s2 – pabaigos viršūnė
   `edgeCost – vertinimo kriterijus (pagal minimalų atstumą, laiką ar emisiją)
   Dim x, i&, j&, k&, n&, m&, flag As Boolean
   x = rng.Value `pervedimas į masyvą
   m = UBound(x) `briaunų skaičius

   ReDim e(1 To m) As edge ` aprašomos briaunos
   For i = 1 To m
      e(i).a = x(i, 1)
      e(i).b = x(i, 2)
      e(i).cost = x(i, edgeCost)
      If n < e(i).a Then n = e(i).a `viršūnių skaičius
      If n < e(i).b Then n = e(i).b
   Next i

   ReDim d#(1 To n), p&(1 To n) `pradžia
   For i = 1 To n
      d(i) = INF        p(i) = -1
   Next i

   d(s1) = 0 `pradžios pozicija
   p(s1) = s1
   Do ` paleidžiamas optimizavimas
      flag = False
      For j = 1 To m
         If d(e(j).a) < INF Then
               If d(e(j).b) > d(e(j).a) + e(j).cost Then
                  d(e(j).b) = d(e(j).a) + e(j).cost
                  p(e(j).b) = e(j).a
                  flag = True
               End If
         End If
      Next j
   Loop While flag

   If d(s2) < INF Then `jei maršrutas rastas
         ReDim tmp(1 To n, 1 To 2) `sukuriamas laikinas masyvas
      j = s2
      i = 0
      Do `maršruto elementai keliami į laikinąjį masyvą
        i = i + 1
        tmp(i, 1) = p(j)
        tmp(i, 2) = j
        j = p(j)
      Loop While j <> s1

        ReDim out(1 To i, 1 To 4) `ieškoma reikšmė grafe pagal vertinimo kriterijų
      For j = 1 To i
         out(i - j + 1, 1) = tmp(j, 1)
         out(i - j + 1, 2) = tmp(j, 2)
         For k = 1 To m
```

```
      If out(i - j + 1, 1) = x(k, 1) And out(i - j + 1, 2) = x(k, 2) Then
        out(i - j + 1, 3) = x(k, 3)
        out(i - j + 1, 4) = x(k, 4)
        Exit For
      End If
    Next k
  Next j
  FordBellman = out
 End If
End Function

Sub test1()
 Dim out
 out = FordBellman(Range(„graph“), [g3].Value, [h3].Value)
 Range(„G6:J30“).ClearContents
 If IsArray(out) Then
  Range(„G6“).Resize(UBound(out), 4) = out
 End If
End Sub

Sub test2()
 Dim out
 out = FordBellman(Range(„graph“), [g3].Value, [h3].Value, 4)
 Range(„G39:J60“).ClearContents
 If IsArray(out) Then
  Range(„G39“).Resize(UBound(out), 4) = out
 End If
End Sub
```

Pavyzdinio optimalaus maršruto pasirinkimo rezultatų suvestinė pagal tris kriterijus pateikiama 1 ir 2 lentelėje.

Pirmame etape pasirenkamos briaunos viršūnės, tarp kurių viršūnių norima rasti optimalų maršrutą: taškas 1 – A_1, taškas 2 – A_{152}.

1 lentelė. Briaunų A_1–A_{152} pagal min L_{ij} ir min t_{ij} optimalaus maršruto rezultatai taikant Belmano ir Fordo modelį
Table 1. Optimal route of Edge A_1–A_{152} using Bellman-Ford model by min L_{ij} and min t_{ij}

Maršruto grandinės briaunų seka		Atstumas (*L*), km	Laikas (*t*), h	Emisijos kiekis, g/TEUkm	Transporto rūšis (*tp*)
Viršūnė *i*	Viršūnė *j*				
1	2	5,259	00:04:06	0,0030	Kelių transportas
2	5	28,74	00:23:33	0,0165	Kelių transportas
5	16	10,91	00:08:16	0,0063	Kelių transportas
16	17	10,91	00:08:06	0,0063	Kelių transportas
17	130	17,5	00:14:22	0,0100	Kelių transportas
130	15	2,7	00:01:52	0,0015	Kelių transportas
15	37	12,9	00:07:44	0,0074	Kelių transportas
37	20	17,9	00:10:46	0,0103	Kelių transportas
20	35	20,4	00:12:40	0,0117	Kelių transportas
35	34	6,521	00:04:37	0,0037	Kelių transportas
34	66	12,7	00:08:34	0,0073	Kelių transportas
66	73	12,4	00:07:02	0,0071	Kelių transportas
73	74	10,34	00:05:58	0,0059	Kelių transportas
74	125	27,6	00:16:11	0,0158	Kelių transportas
125	126	8,2	00:04:40	0,0047	Kelių transportas
126	112	30,2	00:17:22	0,0173	Kelių transportas
112	102	16	00:09:50	0,0092	Kelių transportas
102	145	8,8	00:06:05	0,0051	Kelių transportas
102	145	8,8	00:06:05	0,0051	Kelių transportas
145	103	6,6	00:04:33	0,0038	Kelių transportas
103	100	17,43	00:10:06	0,0100	Kelių transportas
100	97	25,8	00:15:06	0,0148	Kelių transportas
97	98	9,2	00:05:20	0,0053	Kelių transportas
98	146	4,15	00:03:18	0,0024	Kelių transportas
146	147	5,1	00:03:58	0,0029	Kelių transportas
147	94	2,4	00:01:57	0,0014	Kelių transportas
94	152	8,97	00:07:56	0,0051	Kelių transportas
Iš viso		**339,63**	**3:44**	**0,195**	

2 lentelė. Briaunų $A1$–$A152$ pagal min E_{ij} optimalaus maršruto rezultatai taikant Belmano ir Fordo modelį
Table 2. Optimal route of Edge $A1$–$A152$ using Bellman–Ford model by min E_{ij}

Maršruto grandinės briaunų seka		Atstumas (L), km	Laikas (t), h	Emisijos kiekis (E), g/TEUkm	Transporto rūšis (tp)
Viršūnė i	Viršūnė j				
1	2	5,259	0:04	0,0030	Kelių transportas
2	5	28,74	0:23	0,016	Kelių transportas
5	16	10,91	0:08	0,006	Kelių transportas
16	6	7,632	0:06	0,004	Kelių transportas
6	152	377,5	10:49	0,06	Geležinkelių transportas
Iš viso		**430,041**	**11:31**	**0,093**	

Antrame etape pagal sukurtą VBA kodą apskaičiuojamas optimalus maršrutas pagal min L_{ij}, min t_{ij} ir min E_{ij} kriterijus maršrute $A_1 - A_{152}$, kai i yra 1 taškas, o j yra 152 taškas.

Rezultatai rodo, kad 1-ojo rezultato optimalus maršrutas atliekamas kelių transportu, pagal min E_{ij} gautas optimalus 2-ojo rezultato maršrutas sudarytas iš geležinkelių ir kelių transporto tinklo. Nors 1-ojo rezultato maršrutas atliekamas greičiau ir trumpesniu atstumu, 2-ojo rezultato maršrutas išskiria mažesnį emisijos kiekį.

Išvados

Tyrimo metu Belmano ir Fordo dinaminio programavimo uždavinio optimaliam sprendimui buvo naudojama VBA programavimo kalbos kodas MS *Excel* programoje. Šiuo metodu išspręstas optimalaus maršruto Lietuvos žaliajame transporto koridoriuje uždavinys pagal tris optimalumo kriterijus (laiką, atstumą bei teršalų emisijos kiekį), žaliasis transporto koridorius ypatingas mažesne aplinkos tarša vežant įvairiarūšius krovinius. Taip supaprastinamas optimalaus maršruto nustatymas įtraukiant realaus laiko keliuose duomenis iš intelektinių transporto sistemų (ITS) įrenginių (kelių oro sąlygų stotelių ir eismo intensyvumo skaitiklių), kurie leidžia greitai apskaičiuoti optimalų maršrutą (prieš kelionę bei kelionės metu atsižvelgiant į nuolatos atnaujinamą informaciją ir pateikiant ją vartotojui).

Literatūra

Bellman, R. E. 1958. Dynamic programming and stochastic control processes, *Information and Control* 1: 228–239. http://dx.doi.org/10.1016/S0019-9958(58)80003-0

Dantzig, G. B. 1960. On the shortest route through a network, *Management Science* 6(2): 187–190. http://dx.doi.org/10.1287/mnsc.6.2.187

Dijkstra, E. W. 1959. A note on two problems in connection with graphs, *Numerische Mathematik* 1(1): 269–271. http://dx.doi.org/10.1007/BF01386390

EcoTransIT World. 2014. *The calculation of energy consumption and emission data* [Energijos vartojimo ir emisijos parametrų skaičiavimas] [inetraktyvus], [žiūrėta 2014 m. birželio 12 d.]. Prieiga per internetą: http://www.ecotransit.org/calculation.en.html

Huang, H.; Gao, S. 2012. Optimal paths in dynamic networks with dependent random link travel times, *Transportation Research Part B: Methodological* 46(5): 579–598. http://dx.doi.org/10.1016/j.trb.2012.01.005

Kindberg, L. 2013. Improving vessel and supply chain fuel efficiency, in *CAAAC (The Clean Air Act Advisory Committee) Full Committee Meeting*, 26–27 February 2013, Alexandria, VA.

Meimbresse, B.; Lipinski, R. 2007. Tools for route planning of intermodal logistic chains and improvement potential as well as adaptation for spatial planning purposes, in *Logistics and Industrial Informatics LINDI 2007, International Symposium*, 13–15 September 2007, Wildau, 105–110.

Prato, C. G. 2009. Route choice modelling: past, present and future research directions, *Choice Modelling* 2(1): 65–100. http://dx.doi.org/10.1016/S1755-5345(13)70005-8

Sarimveisa, H.; Patrinosa, P.; Tarantilis, C. D.; Kiranoudisa, C. T. 2008. Dynamic modelling and control of supply chain systems: a review, *Computers & Operations Research* 35(11): 3530–3561. http://dx.doi.org/10.1016/j.cor.2007.01.017

Winebrake, J. J.; Hatcher, J.; Farrell, A. E. 2007. *Emissions Analysis of Freight Transport Comparing Land-Side and Water-Side Short-Sea Routes: Development and Demonstration of a Freight Routing and Emissions Analysis Tool (FREAT)*. Final report. The United States Department of Transportation Research and Special Programs Administration.

BELLMAN – FORD METHOD FOR SOLVING THE OPTIMAL ROUTE PROBLEM

L. Greičiūnė

Abstract

The article aims to adapt the dynamic programming method for optimal route determination using real-time data on ITS equipment. For this purpose, VBA code has been applied for solving the Bellman – Ford method for an optimal route considering optimality criteria for time, distance and the amount of emissions.

Keywords: dynamic programming, optimal route, VBA code, intermodal transport.