

*Szolnoki Tudományos Közlemények XI.
Szolnok, 2007.*

OLÁH BÉLA¹

A KÖRUTAZÁSI PROBLÉMÁK MEGOLDÁSÁRA SZOLGÁLÓ DACEY-, ÉS DACEY-VOGEL MÓDSZEREK ÖSSZEHASONLÍTÁSA

I. BEVEZETÉS

Dolgozatom célja, a körutazási probléma megoldására szolgáló matematikai módszerek ismertetése, ezen algoritmusokra épülő egy számítógépes program bemutatása, valamint a heurisztikus módszerek összevetése logisztikai peremfeltételek és célfüggvények figyelembevételével. A dolgozat első részében ismertetésre kerülnek a járatrendezés logisztikai vonatkozásai, összefoglalásra kerülnek az utazó ügynök probléma alapvető megoldási módszerei. Mivel a lehetséges megoldási változatok összehasonlítása igen körülményes lehet, ezért a következő tartalmi részben bemutatásra kerül egy számítógépes szoftver, amely dinamikus programozáson alapulva értékeli ki a kapott eredményeket.

II. AZ UTAZÓ ÜGYNÖK PROBLÉMA ÉS MEGOLDÁSI MÓDSZEREI

Az utazó ügynök problémát a nemzetközi szakirodalom traveling salesman problémának (TSP) nevezi (Evans 1993; Cormen et al. 1997). Az elnevezés eredete azokra a kereskedelmi ügynökökre vezethető vissza, akik azt a feladatot kapták, hogy meghatározott számú ügyfelet keressenek fel, majd térjenek vissza kiinduló állomásukra. Az ügynök nyilvánvalóan arra törekszik, hogy lehetőség szerint a legrövidebb út megtételével hajtsa végre a feladatot. Később a szállítási és anyagmozgatási területen is alkalmazták a problémát és megoldását (Galántai et al. 1997).

A körutazási probléma matematikai megfogalmazása (Lipi et al. 1990): Legyen adott n számú pont (P_1, P_2, \dots, P_n) , valamint ezek egymástól való távolsága (c_{ij}) . A feladat egy adott városból (P_i) kiindulva a legrövidebb úton bejárni valamennyit úgy, hogy mindegyiket csak egyszer szabad érinteni, és az út a kezdőpontban legyen befejezve (minimális Hamilton körök keresése). A körút

¹ Szolnoki Főiskola, Műszaki és Mezőgazdasági Fakultás, Mezőtúr, Gépészeti Tanszék

hossza (bejárási ideje) különbözően alakul attól függően, hogy milyen sorrendben követik egymást a felkeresendő pontok. Matematikai formában a célfüggvény:

$$Q = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \longrightarrow \min. \quad (1)$$

ahol

- c_{ij} az i -edik és j -edik pont közötti távolság. (ha az út kétirányú forgalom lebonyolítására alkalmas, akkor $c_{ij} = c_{ji}$)
- x_{ij} döntési változó, azt fejezi ki, hogy az i - j viszonylat eleme-e a körútnak vagy sem.

A célfüggvény korlátozó feltételei:

$$x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \quad (i, j = 1, 2, \dots, n) \quad (2)$$

Adott i - j viszonylat költség értéke csak akkor szerepelhet, ha a viszonylat eleme a körútnak.

$$\sum_{i=0}^n x_{ij} = 1 \quad (j = 1, 2, \dots, n) \quad (3)$$

Egy pontból csak egyetlen másik pontba vezethet a körút.

$$\sum_{j=0}^n x_{ij} = 1 \quad (i = 1, 2, \dots, n) \quad (4)$$

Egy pontba csak egyetlen másik pontból vezethet a körút.

$$I = \{i_1 i_2, i_2 i_3, \dots, i_{n-1} i_n, i_n i_1\} \quad (5)$$

A körút elemei n elemű index láncot kell, hogy alkossanak.

Az utazó ügynök probléma megoldására számos módszert ismertet a (Prezenszki 1999) szakirodalom (Oszlopösszegző és Croes módszer, stb.). Ezek közül a közelítő megoldást adó Dacey- és Dacey-Vogel módszerek kerülnek bemutatásra. Optimális megoldást az enumerációs (leszámlálós) és a Branch and Bound módszer ad.

2.1. Dacey-módszer

Az algoritmus szerint lehetőleg a legnagyobb értékű költség elemeket jelentő viszonylatokat törölni kell a mátrixból, hogy ne kerülhessenek beépítésre a körútba (Tóth 1990). A módszer főbb lépései:

1. Kiinduló adatok (távolságmátrix) meghatározása.
2. A távolságmátrix legnagyobb elemének megkeresése és törlése. Ha több ilyen elem van, akkor a törlés sorrendje tetszőleges lehet. A lépést addig kell ismételni, míg elő nem áll egy olyan helyzet, hogy a mátrix valamely sorában vagy oszlopában már csak egyetlen elem marad. Ha több olyan sor vagy oszlop maradt, melyben egy elem van, akkor a kisebb értékűt célszerű választani.
3. A megmaradt elem által meghatározott viszonylat része kell, hogy legyen a körútnak, mert különben nem teljesül az előző fejezetben megfogalmazott 2. és 3. korlátozó feltétel. Mivel az elem a viszonylat része lett, ezért törölni kell:
 - az elemhez tartozó sor és oszlop összes többi (eddig nem törölt) elemét;
 - a kiválasztott elemnek a mátrix főátlójára szimmetrikus párját;
 - minden olyan elemet, amely a mátrixban maradván a 4. korlátozó feltétel kielégítését akadályozza;
 - ha az utoljára kiválasztott viszonylattal nem sikerült olyan körutat létrehozni, amely minden pontot tartalmaz, akkor a 2. lépéshez vissza kell térni.

Példa. Egy tehergépkocsival árugyűjtést végeznek, az **A** pontból indulva kell bejárni a hat felrakóhelyet (**B, C, D, E, F, G**). A feladat: meghatározni a gépkocsi legrövidebb útvonalát Dacey–módszerrel. A megoldásban szürkével a törölt elemek, sárgával a körút részei vannak jelölve.

A 16, 15, 14, 13, 12 értékű cellák törlése után az **E** jelű oszlopban 1 elem marad, tehát a **G-E** szakasz eleme a körútnak.

A	9	10	9	12	13	15
9	B	5	12	14	7	10
10	5	C	7	15	8	12
8	12	7	D	16	8	15
12	14	15	16	E	11	10
13	7	6	8	12	F	9
14	10	12	15	11	9	G

Az **E** oszlopban és a **G** sorban lévő minden más elem, és a főátlóra szimmetrikus párja kiesik A következő egyedülálló elem az **E** sorban van.

A	9	10	9	12	13	15
9	B	5	12	14	7	10
10	5	C	7	15	8	12
8	12	7	D	16	8	15
12	14	15	16	E	11	10
13	7	6	8	12	F	9
14	10	12	15	11	9	G

Kiesik az **F** oszlop, és az **F-G** szakasz (mert rövidebbre zárja a **G-E-F** szakaszt - 4. korlátozó feltétel -). A **G** oszlopban **B-G** szintén része a körútnak.

A	9	10	9	12	13	15
9	B	5	12	14	7	10
10	5	C	7	15	8	12
8	12	7	D	16	8	15
12	14	15	16	E	11	10
13	7	6	8	12	F	9
14	10	12	15	11	9	G

Kiesik a **B** sor, és az **F-B** szakasz. Mivel minden sorban egynél több elem van, a 10-es elemeket törölni kell, az **A** oszlopban egy elem marad.

A	9	10	9	12	13	15
9	B	5	12	14	7	10
10	5	C	7	15	8	12
8	12	7	D	16	8	15
12	14	15	16	E	11	10
13	7	6	8	12	F	9
14	10	12	15	11	9	G

D-A szakasz szintén része a körútnak, és kiesik a **D** sor és az **A-D** szakasz, így **A-B** a körút része.

A	9	10	9	12	13	15
9	B	5	12	14	7	10
10	5	C	7	15	8	12
8	12	7	D	16	8	15
12	14	15	16	E	11	10
13	7	6	8	12	F	9
14	10	12	15	11	9	G

Kiesnek a **B** oszlop elemei, és a **C** sorban egy elem marad, így a **C-D** is a körbe tartozik, végül az **F-C** zárja a kört.

A	9	10	9	12	13	15
9	B	5	12	14	7	10
10	5	C	7	15	8	12
8	12	7	D	16	8	15
12	14	15	16	E	11	10
13	7	6	8	12	F	9
14	10	12	15	11	9	G

Tehát a sorrend: **A-B-G-E-F-C-D** és **A**.

Körút hossza: $9 + 10 + 7 + 8 + 11 + 6 + 11 = 62$ km.

2.2. Dacey–Vogel módszer

A módszer a szállítási probléma egyik induló megoldást nyújtó módszerének (Vogel-Korda módszer) és az előzőekben ismertetett (Dacey módszer) algoritmusnak az ötvözetéből született. A módszer főbb lépései (Nagy 1998):

1. Kiinduló adatok (távolságmátrix) meghatározása.
2. Valamennyi sorra és oszlopra meghatározzuk a két legkisebb elem értékét, és különbségüket a mátrix jobb oldalán illetve alul tüntetjük fel.
3. A maximális differenciájú sorban vagy oszlopban található legkisebb elem lesz a körút viszonylata.
4. A kiválasztott elem miatt törölni kell Dacey-módszernél ismertetett feltételeknek megfelelő elemeket.
5. A törölt elemeket figyelmen kívül hagyva a 2. pont szerint újra meg kell határozni a sor és oszlop differenciák értékét, majd a legnagyobb differenciájú sorban vagy oszlopban található legkisebb eleme lesz a viszonylat része. A ciklust addig kell folytatni, amíg a körút teljes nem lesz.

Példa. Az előző példa megoldása Dacey-Vogel módszerrel.

N _o		1.	2.	3.	4.	5.	Sordifferenciák	
3.	A	9	10	<u>9</u>	12	13	15	0, 1, 3
5.	9	B	5	12	14	7	<u>10</u>	2, 2, 1, 1, 4
1.	10	<u>5</u>	C	7	15	8	12	2
4.	8	12	7	D	16	<u>8</u>	15	1, 1, 0, 7
	<u>12</u>	14	15	16	E	11	10	1, 1, 1, 1, 2
2.	13	7	<u>6</u>	8	12	F	9	1, 2
	14	10	12	15	<u>11</u>	9	G	1, 2, 2, 2, 3
Oszlop-differenciák	1, 1, 1, 3, 2	2	1, 1	1, 1, 3	1, 1, 1, 3, 3	1, 1, 1, 1	1, 1, 0, 0, 0	

Tehát a sorrend: **A-D-F-C-B-G-E** és **A**.

Körút hossza: $9 + 10 + 5 + 8 + 12 + 6 + 11 = 61$ km.

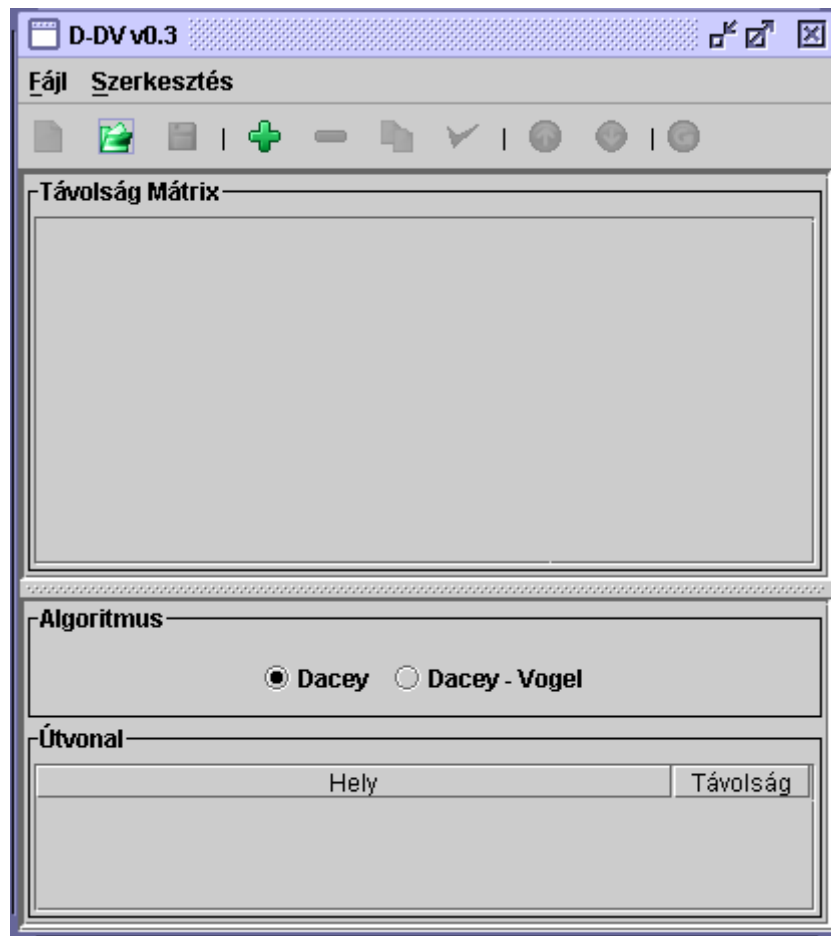
3. A SZÁMÍTÓGÉPES PROGRAM BEMUTATÁSA

A Java programnyelven írt program (1. ábra) egy tetszőleges számú állomás között megvalósítható körutazási feladat közelítő megoldását adja meg Dacey- vagy Dacey-Vogel módszer alkalmazásával. A program futásához szükséges egy mátrix, amely az egyes csomópontok közötti költséget (mely lehet akár távolság vagy idő) adja meg.

A programhoz nem kötődnek szorosan mértékegységek, ezért tetszőleges költség alapú számításra használható. A program felülete 3 egységre tagolódik:










- Távolságmátrix:
Itt tudjuk a táblázat adatait feltölteni. A táblázat elemeiként kizárólag csak számok adhatóak meg (a program nem engedélyezi betűk felvitelét a mátrixba). Ugyanakkor nem feltétel, hogy a mátrix szimmetrikus legyen, hiszen nem minden esetben ugyan azok az adatok (pl.: idő szempontjából nem mindegy, hogy milyen domborzati viszonyok vannak az oda-, illetve a visszaút során).
- Algoritmus:
Kiválaszthatjuk a Dacey- vagy a Dacey-Vogel algoritmust, attól függően, hogy melyik módszerrel szeretnénk meghatározni az útvonalat. Ezáltal lehetőség nyílik arra, hogy azonos bemeneti mátrix esetén összehasonlítsuk a két módszer által meghatározott eredményt.

- Útvonal:
A megfelelő algoritmus kiválasztása után itt tekinthetjük meg a program által meghatározott állomások sorrendjét és a körút teljes hosszát.



1. ábra: A számítógépes szoftver felhasználói felülete.

A program eszköztára: a táblázattal végezhető műveletek parancsikonzit tartalmazza. A program eszköztára a következő funkciókat biztosítja:

-  Új üres mátrix létrehozása: Egy olyan mátrix kerül létrehozásra, mely nem tartalmaz egyetlen helyszínt sem.
-  Adatok betöltése: Előzőleg mentett adatok felvitele a mátrixba.
-  Adatok mentése: Az éppen aktuális mátrix mentése.
-  Új hely hozzáadása: Új helyszínnel bővíthetjük a mátrixot. Az új elem nevét egy felbukkanó ablakban adhatjuk meg.
-  Hely törlése: Törli az aktuális helyet a mátrixból.
-  Hely átnevezése: A kiválasztott hely nevét változtathatjuk meg.
-  Hely tiltása: Ezzel a parancsikonnal figyelmen kívül hagyhatunk bizonyos helyszíneket a számítások során.
-  Hely engedélyezése: Előzőleg tiltott helyszínt újra figyelembe vesz.
-  Mozgatás felfelé: A kiválasztott helyet egy pozícióval feljebb mozgatja (sora eggyel felfelé, oszlopa eggyel balra tolódik).



Mozgatás lefelé: A kiválasztott helyet egy pozícióval lejjebb mozgatja (sora eggyel lefelé, oszlopa eggyel jobbra tolódik).



Útvonal meghatározás: A kiválasztott algoritmus segítségével az adott költség mátrix alapján meghatároz egy lehetséges körútvonalat.

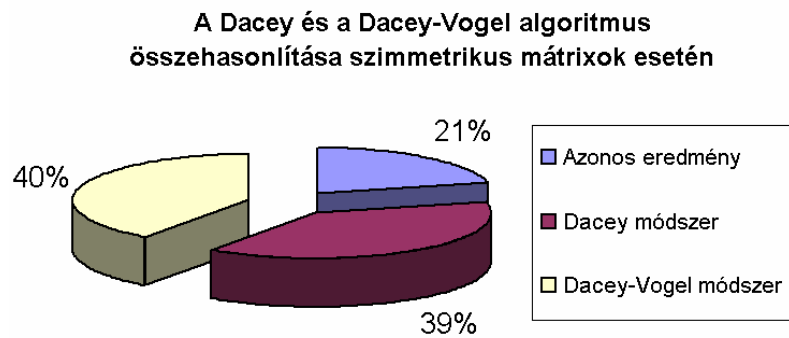
4. A PROGRAM TESZTELÉSE

A két algoritmus vizsgálatára tesztelést végeztem. A tesztprogram először 10x10-es szimmetrikus, majd 10x10-es aszimmetrikus véletlen-szám generátor által létrehozott mátrixokat vizsgált. A létrehozott mátrixok száma 10000 darab mind a két esetben.

4.1 Szimmetrikus mátrixszal végrehajtott tesztelés

A tesztprogram eredményei (1. táblázat):

- 2105 esetben (21,05%) azonos eredményt adott mindkét módszer;
- 3895 esetben (38,95%) a Dacey-módszer adott jobb eredményt;
- 4000 esetben (40%) a Dacey-Vogel-módszer bizonyult jobbnak.

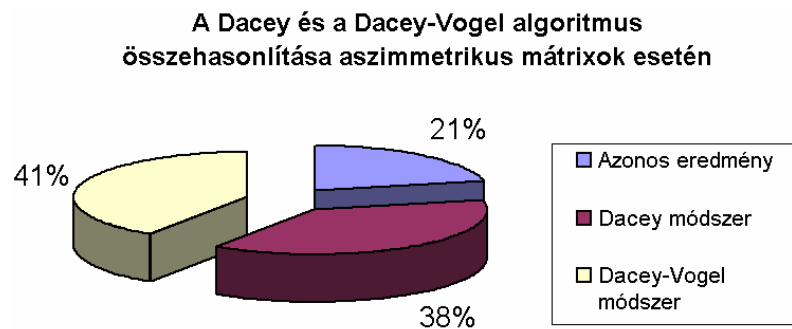


1. táblázat: A Dacey és a Dacey-Vogel algoritmus összehasonlítása szimmetrikus mátrixok esetén

4.2 Aszimmetrikus mátrixszal végrehajtott tesztelés

A tesztprogram eredményei (2. táblázat):

- 2136 esetben (21,36%) azonos eredményt eredményezett mindkét algoritmus;
- 3822 esetben (38,22%) a Dacey-módszer adott jobb eredményt;
- 4042 esetben (40,42%) a Dacey-Vogel-módszer bizonyult jobbnak.



2. táblázat: A Dacey és a Dacey-Vogel algoritmus összehasonlítása aszimmetrikus mátrixok esetén

Mind a szimmetrikus, mind pedig az aszimmetrikus mátrixokon történő tesztelés során megállapítható, hogy a sokak által esetleg oly nagyon várt szignifikáns különbség nem állapítható meg a két módszer hatékonysága között, hiszen ahogy az a táblázatokon is jól látható, a két algoritmus mind a két esetben 20%-ban azonos eredményt adott, míg a maradék 79%-on 40-39, illetve 41-38 arányban osztoztak. Bár mindkétszer a Dacey-Vogel módszer adott nagyobb százalékban jobb eredményt, véleményem szerint mégsem tehetjük le voksunkat ezen algoritmus mellett, hiszen nem realizálható jelentős különbség, igaz tesztelésem során nem vizsgáltam, hogy az eltérések milyen mértékűek voltak a nem azonos eredmények esetén.

5. ÖSSZEFOGLALÁS

A gyakorlati tapasztalatok azt mutatják, hogy a körutazási feladatok megoldására eddig alkalmazott heurisztikus módszerek csak igen kis rendszerek esetén közelítik meg az optimális megoldást. Nagyobb adatstruktúra esetén azonban kifejezetten gyengén szerepelnek.

Az utóbbi évtizedben olyan megközelítési módok terjedtek el, amelyeket összefoglaló néven természetes számításnak (natural computing) hívunk, mert a természettől lestük el, hogy különböző problémákat hogyan old meg (Adleman 1994). A természet évmilliárdok óta nagyon jó eredménnyel számol. A jó eredménnyel azt értjük, hogy folyamatosan fejlődnek a leszármazottak, az élet stabilan létezik, a lények hatékonyan működnek. Kézenfekvő a gondolat, hogy próbáljuk meg imitálni az életet, hátha a természetben évmilliárdok óta jól működő eljárások mesterséges környezetben is hatékonyan alkalmazhatók (Katsányi 2000). Tény az is, hogy a számítógépes világban az evolúciós- és a genetikus algoritmusok egyre nagyobb jelentőségre tesznek szert (Kömlödi 2000).

A genetikus algoritmusok (Holland 1975; Goldberg 1989; Michalewicz 1996;) tervezése során az evolúciót tekinthetjük mintaképnek. Kezdetben nem optimálisan megírt, vagy paraméterezett programok keresztezések során a természetes kiválasztódás elve alapján fejlődnek, és a tapasztalat szerint közelítenek egy jó megoldáshoz. Nagyon jól alkalmazhatók bizonyos optimalizálási problémákhoz. Az is a genetikus algoritmus mellett szól, hogy már néhány száz iteráció után ad egy viszonylag jó eredményt, míg a heurisztikus algoritmusban nincs félkész megoldás. Ha egyszer megvan a jó adatrepresentáció, könnyedén variálhatjuk kiválasztását, mutációit valamint keresztezését az optimális eredmény érdekében. Egyszerűségével mindenestre szót érdemel. Az informatika rohamos fejlődése biztosan előremozdítja ezt az ágát a problémamegoldásnak.

6. KÖSZÖNETNYILVÁNÍTÁS

A tudományos munkát a Főiskolai Publikációs Alapítvány és A Mezőtúri Hallgatókért Alapítvány támogatta.

7. IRODALOMJEGYZÉK

- [1] ADLEMAN L. M.: Molecular computation of solutions to combinatorial problems. *Science*, 266:1021-1024, 1994.
- [2] CORMEN T. H.; LEISERSON C. E.; RIVEST R. L.: *Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1997.
- [3] EVANS J. R.: *Applied Production and Operations Management*, 4th ed, West Publishing Company, 1993.
- [4] GALÁNTAI A.; HUJTER M.: *Optimalizálási módszerek*, Miskolci Egyetemi Kiadó, 1997.

- [5] GOLDBERG D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [6] HOLLAND J. H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology Control, and Artificial Intelligence*, First MIT Press Edition 1992, Cambridge, MA, second edition; First edition 1975, The University of Michigan Press
- [7] KATSÁNYI I.: Molekuláris számítások, 2000.
<http://ludens.elte.hu/~kacs/molszame.html>
- [8] KÖMLŐDI F.: Evolúciós algoritmusok, 2000. <http://www.index.hu/kultur/cyb/evolucio>
- [9] LIPI G.; NEMES Á.; NOVÁK I.: *A kínai hadseregtől az utazó ügynökig*, Novotrade Kiadó, Budapest, 1990.
- [10] MICHALEWICH Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1996.
- [11] NAGY T.: *Operációkutatás*, Miskolci Egyetemi Kiadó, Miskolc, 1998.
- [12] PREZENSZKI J.: *Logisztika II. (Módszerek, eljárások)*, Logisztikai Fejlesztési Központ, Budapest, 1999.
- [13] TÓTH I.: *Operációkutatás I.*, Tankönyvkiadó, Budapest, 1990.

Béla OLÁH

COMPARISON OF DACEY AND DACEY-VOGEL METHODS FOR SOLVING OF TRAVELING SALESMAN PROBLEMS

Abstract: The traveling salesman problem (TSP) is a very important practical problem. This problem is a complex combinatorial optimization problem. TSP is NP-hard. The main goal of this scientific work is to take a survey of the solution methods of routing problem, to describe two simple algorithm (Dacey and Dacey-Vogel methods) which produce reasonably good results very quickly, finally to compare the heuristic methods considering logistic conditions and objective functions such as minimum route or time.