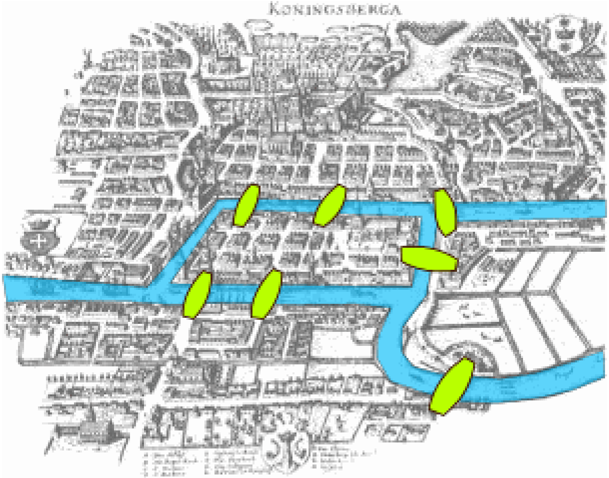
# Gráfalgoritmusok

Gaskó Noémi

2023. május 9.

# Tartalomjegyzék


KONINGSBERGA

## Euler vonal

Euler vonal - a gráf minden élét tartalmazza.

## Euler vonal

Euler vonal - a gráf minden élét tartalmazza.

## Euler gráf

Egy gráfot Euler gráfnak nevezünk, ha van benne zárt Euler-vonal.

### Euler vonal

Euler vonal - a gráf minden élét tartalmazza.

### Euler gráf

Egy gráfot Euler gráfnak nevezünk, ha van benne zárt Euler-vonal.

### Tétel

Egy izolált csúcsokat nem tartalmazó gráf akkor és csakis akkor Euler gráf, ha összefüggõ és minden csúcsának fokszáma páros.

## Bizonyítás

a) Ha a gráf Euler gráf, akkor van benne zárt Euler vonal, tehát összefüggõ. A zárt vonal minden csúcs érintésekor pontosan két élt használ, tehát minden fokszám páros.
b) Legyen G egy összefüggõ gráf, amelyben minden csúcs fokszáma páros. Az élek számára vonatkozó indukcióval bizonyítjuk, hogy ekkor létezik zárt Euler-vonal.

**Tétel**

Egy izolált csúcsokat nem tartalmazó gráf akkor és csakis akkor tartalmaz Euler vonalat, ha két páratlan fokszámú csúcsán kivül minden más csúcs fokszáma páros.

# Fleury algoritmusa

-Henry Fleury - 1883

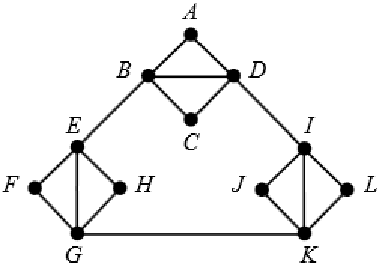
**Fleury algoritmusa**

- válasszunk ki egy tetszõleges v csomópontot (legyen ennek a neve w)
- ismételd ameddig van él a gráfban
  válasszunk ki a w valamely élét, hidat csak akkor válasszunk, ha nincs más lehetõség
  a kiválasztott élt adjuk hozzá az útvonalhoz, míg a w legyen az él másik vége
  töröljük le az élt

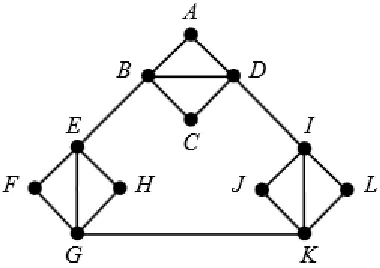
Egy élt **hídnak** nevezünk, ha elhagyásával eggyel nõ a a gráf összefüggõ komponenseinek a száma

Az algoritmus bonyolultsága: $O(m(n+m))$, ha Tarjan algoritmusával határozuk meg a hidakat.

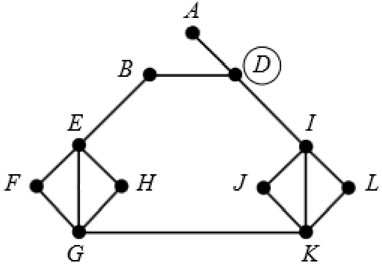
# Egy példa



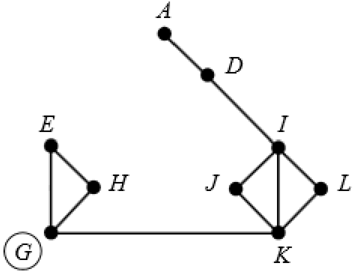
# Egy példa



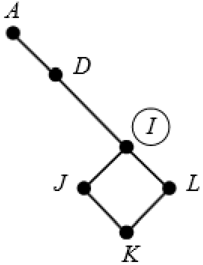

A-ból kindulva AB, BC, CD



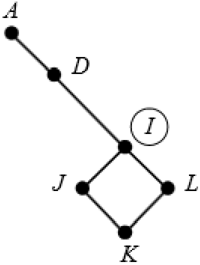

DA egy híd, kiválasztjuk DB, BE, EF, FG




GK híd, GE, EH, HG, GK, KI




ID híd, IJ, JK, KL, LI, ID, DA.




ID híd, IJ, JK, KL, LI, ID, DA.
Az Euler út: ABCDBEFGEHGKIJKLIDA

# Hierholzer algoritmusa

-Hierholzer - 1873

**Hierholzer algoritmus**

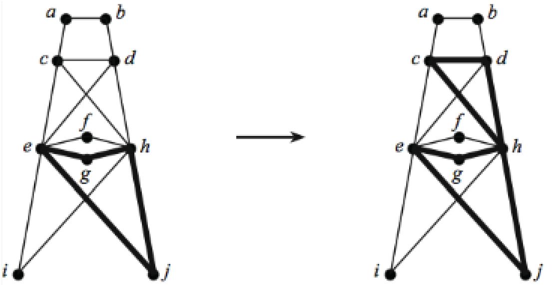
- keressünk egy $R_1$ kört a gráfban, jelöljük meg a kör éleit
- ha az $R_1$ G minden élét tartalmazza az algoritmus befejezõdik
- ellenkezõ esetben legyen $v_i$ egy olyan csomópont az $R_i$-bõl, ami tartalmaz egy nem bejárt élt.
- ebbõl építsünk fel egy $Q_i$ kört
- ismételjük a lépéseket addig, amíg minden élt felhasználtunk

# Hierholzer algoritmusa

Rekurzivan:

```
Hierholzer(u)
   Amig u-nak van szomszédja
      torol(u,v)
      Hierholzer(v)
   Euler_vonal.add(u)
```

# Egy példa



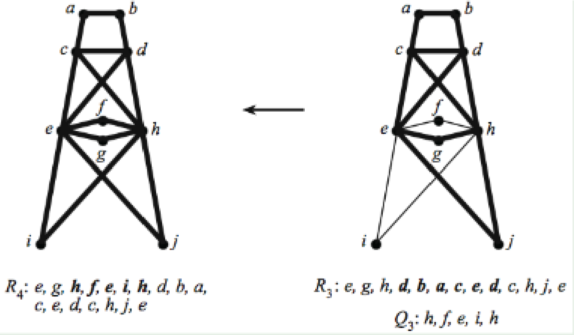

$R_1$: $e, g, h, j, e$

$Q_1$: $h, d, c, h$

$R_2$: $e, g, \mathbf{h}, \mathbf{d}, \mathbf{c}, \mathbf{h}, j, e$

$Q_2$: $d, b, a, c, e, d$

# Egy példa (2)


a
b
c
d
f
e
h
g
i
j
$R_4$: e, g, h, f, e, i, h, d, b, a, c, e, d, c, h, j, e
$R_3$: e, g, h, d, b, a, c, e, d, c, h, j, e
$Q_3$: h, f, e, i, h

# Irányított gráfok esetén

**Tétel**

Egy összefüggõ irányított gráfban akkor és csakis akkor létezik irányított zárt Euler-vonal, ha minden v csúcsra igaz, hogy $\varphi^{ki}(v) = \varphi^{be}(v)$.

# Irányított gráfok esetén

**Tétel**

Egy összefüggõ irányított gráfban akkor és csakis akkor létezik irányított zárt Euler-vonal, ha minden v csúcsra igaz, hogy $\varphi^{ki}(v) = \varphi^{be}(v)$.

**Tétel**

Egy összefüggõ irányított gráfban akkor és csakis akkor létezik irányított nyitott Euler-vonal, ha van két olyan u és v csúcsa, amelyekre $\varphi^{ki}(v) = \varphi^{be}(v) - 1$, $\varphi^{ki}(u) = \varphi^{be}(u) + 1$ és a gráf minden más $x$ csúcsára $\varphi^{ki}(x) = \varphi^{be}(x)$. A vonal $u$-val kezdõdik és $v$-vel végzõdik.

# Kínai postás problémája

-1962-ben Kuan Mei-Ko kínai matematikus fogalmazta meg

# Kínai postás problémája

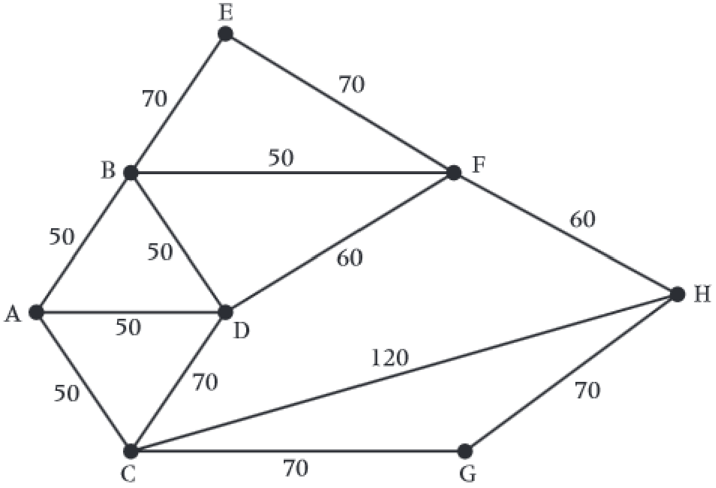
-1962-ben Kuan Mei-Ko kínai matematikus fogalmazta meg
Kínai postás problémája (súlyozott gráfok esetén): egy postás a város minden utcáján kell járjon legalább egyszer. Melyik a legrüvidebb útvonal?

# Kínai postás problémája

-1962-ben Kuan Mei-Ko kínai matematikus fogalmazta meg
Kínai postás problémája (súlyozott gráfok esetén): egy postás a város minden utcáján kell járjon legalább egyszer. Melyik a legrüvidebb útvonal?
Kínai postás problémája (súlyozatlan gráfok esetén): legfeljebb hány élismétléssel lehet bejárni egy gráfot úgy, hogy minden élen áthaladjunk legalább egyszer.

Mikor létezik ideális megoldás? (tehát csak egyszer járjuk be az éleket)

# Egy példa

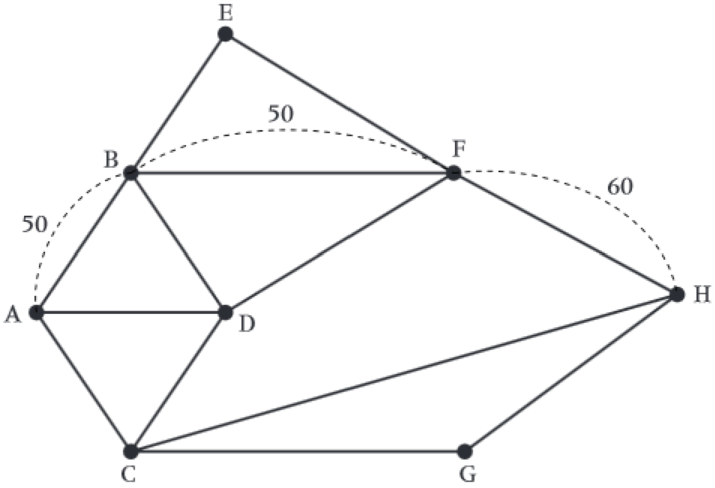


# Az algoritmus

1. határozzuk meg a páratlan fokszámú csúcsokat
2. határozzuk meg az összes lehetséges párosítást
3. minden párosítás esetén határozzuk meg a legrövidebb utat
4. határozzuk meg úgy a párosításokat, hogy az összeg minimális legyen
5. a 4. lépésben talált megoldást adjuk hozzá az eredeti gráfhoz
6. az összhossz az egésznek az összege lesz

Megjegyzés: a 2. és 3. lépés egyben a minimális összegű maximális párosítás meghaározása

Bonyolultság: $O(n^3)$ a minimális összegű maximális párosítás miatt


E
50
B
F
60
50
H
A
D
C
G

Kínai postás problémája irányított gráfok esetén:

- csak akkor létezik megoldás, ha a gráf erősen összefüggő
- meg kell határozni a $\delta(u) = \varphi^{be}(u) - \varphi^{ki}(u)$ értékeket.
- építsük fel egy K teljes páros gráfot, melyben bal oldalon a pozitív, jobb oldalon a negatív $\delta$ értékkel rendelkező csomópontok vannak.
- minden u csomópont $|\delta(u)|$-szor szerepel, az élek hossza a megfelelő csúcsok közötti legrövidebb utak hosszával egyenlő
- határozzuk meg a minimális összegű párosítást K-ban, és utána ugyanazok a lépések mint nem irányított gráfok esetén

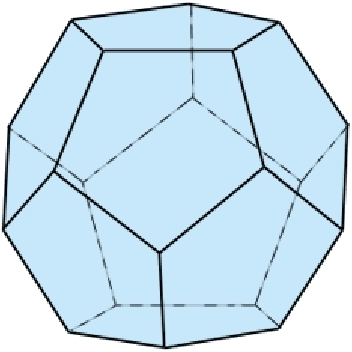
Egy példa: lásd 9_jegyzet.pdf

# Kínai postás problémájának változatai

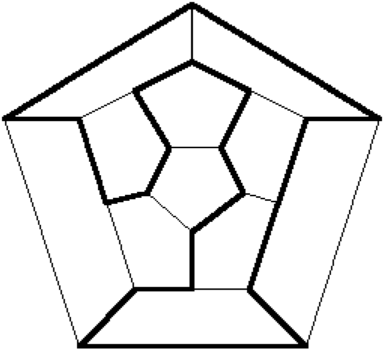
- vidéki postás problémája - csak az élek egy részhalmazát kell bejárni
- megszorításos feladat - egy kitüntetett él nem kell kötelező módon a megoldás része legyen
- ha a postás bárhonnan indulhat és bárhova érkezhet

# Hamilton - dodekaéder játék

-1857-ben Hamilton - dodekaéder játék
-a dodekaéder 20 csúcsú, 30 élû, 12 lapú szabályos ötszög
- a feladat: minden csúcs egy várost jelent, be kell járni minden várost, csak egyszer érintve mindegyiket, és visszajutni a kiindulópontba.



# Megoldás



## Hamilton út

Egy utat Hamilton útnak nevezünk, ha tartalmazza a gráf összes csúcsát.

### Hamilton út

Egy utat Hamilton útnak nevezünk, ha tartalmazza a gráf összes csúcsát.

### Hamilton kör

Egy kört, amely tartalmazza a gráf összes csúcsát, Hamilton körnek nevezünk.

### Hamilton út

Egy utat Hamilton útnak nevezünk, ha tartalmazza a gráf összes csúcsát.

### Hamilton kör

Egy kört, amely tartalmazza a gráf összes csúcsát, Hamilton körnek nevezünk.

### Hamilton gráf

Egy gráf Hamilton gráf, ha tartalmaz Hamilton kört.

# Mikor létezik Hamilton út?

## Ore tétele

Ha egy legalább $n \geq 3$ csúcsú egyszerû gráfban bármelyik két nemszomszédos $u$, $v$ csúcsra igaz, hogy $\varphi(u) + \varphi(v) \geq n$ akkor a gráfban van Hamilton kör.

## Bizonyítás

Reductio ad absurdum.

## Dirac tétele

Ha egy legfeljebb $2k$ csúcsú egyszerû gráfban minden csúcs fokszáma legalább $k$, $(k > 1)$ akkor a gráfban van Hamilton kör.

## Bizonyítás

Alkalmazzuk Ore tételét.

### Dirac tétele

Ha egy legfeljebb $2k$ csúcsú egyszerû gráfban minden csúcs fokszáma legalább $k$, $(k > 1)$ akkor a gráfban van Hamilton kör.

### Bizonyítás

Alkalmazzuk Ore tételét.

### Tétel

Legyen $G$ egy $n \geq 2$ csúcsú egyszerû gráf. Ha $\varphi(v) \geq \frac{n-1}{2}$ bármelyik $v$ csúcsra, akkor $G$ tartalmaz Hamilton utat.

## Rédei tétele

Egy legalább kétcsúcsú teljes gráf éleinek tetszõleges irányításával kapott gráfban mindig van irányított Hamilton út.

## Rédei tétele

Egy legalább kétcsúcsú teljes gráf éleinek tetszõleges irányításával kapott gráfban mindig van irányított Hamilton út.

## Tétel

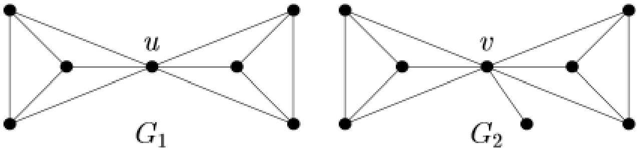
Ha egy G gráf k csúcs kitörlése után több mint k komponensre bomlik, akkor G nem tartalmaz Hamilton kört. Ha k csúcs törlésével G több mint k+1 komponensre bomlik, akkor G Hamilton utat sem tartalmaz.

## Rédei tétele

Egy legalább kétcsúcsú teljes gráf éleinek tetszõleges irányításával kapott gráfban mindig van irányított Hamilton út.

## Tétel

Ha egy G gráf k csúcs kitörlése után több mint k komponensre bomlik, akkor G nem tartalmaz Hamilton kört. Ha k csúcs törlésével G több mint k+1 komponensre bomlik, akkor G Hamilton utat sem tartalmaz.

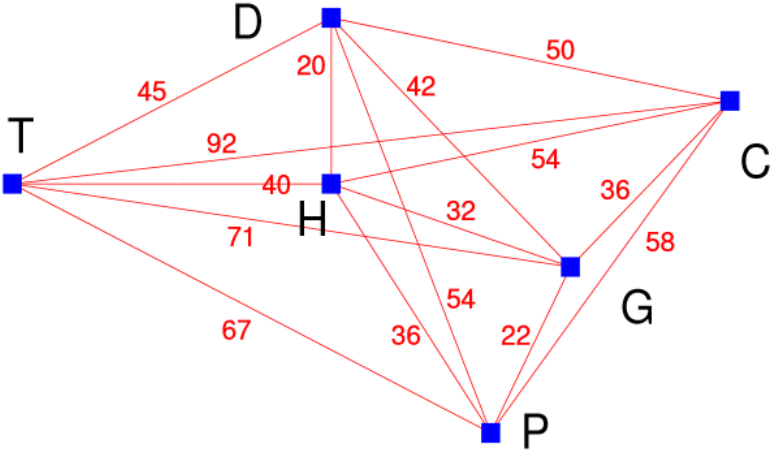


# Feladat

Hamupipőkét elküldi a gonosz mostohája bevásárolni hat különböző helyre. Hogyan tudja a leghamarabb elintézni a bevásárlást Hamupipőke, hogy eljuthasson a bálba?

# Feladat

Hamupipőkét elküldi a gonosz mostohája bevásárolni hat különböző helyre. Hogyan tudja a leghamarabb elintézni a bevásárlást Hamupipőke, hogy eljuthasson a bálba?

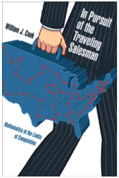


# Az utazó ügynök problémája

- 1930-ban fogalmazták meg

# Az utazó ügynök problémája

- 1930-ban fogalmazták meg
- kombinatorikus optimalizálási problémák körébe tartozik




## A feladat

Adott n város és az út költsége bármely két város között. Az utazóügynök minden várost kell érintsen úgy, hogy minden városba csak egyszer jut el, és az út végén visszaér a kiinduló városba. A feladat olyan útvonal meghatározása, melynek hossza minimális és teljesíti a feltételeket.

# Egy példa


DEN.
Baltic
Sea
North
Sea
Kiel
Rostock
Neuendorf
bei Wilster
Lübeck
Bremerhaven
Hamburg
Emden
Bremen
BERLIN
POL.
Hannover
NETH.
Magdeburg
Duisburg
Essen
Leipzig
Elbe
Düsseldorf
Kassel
Köln
Dresden
BEL.
Bonn
Frankfurt
am Main
Wiesbaden
CZECH
REPUBLIC
Rhine
Mannheim
LUX.
Nürnberg
Karlsruhe
Stuttgart
Danube
FRANCE
München
AUSTRIA
Zugspitze
0 50 100 km
0 50 100 mi
SWITZ.
LIECH.

# Jelenlegi algoritmusok

| Év | Kutatók | Városok száma | TSPLIB |
|---|---|---|---|
| 1954 | G. Dantzig, R. Fulkerson, S. Johnson | 49 | dantzig42 |
| 1971 | M. Held, R. M. Karp | 64 | véletlen pontok |
| 1975 | P. M. Camerini, L. Fratta, F. Maffioli | 67 | véletlen pontok |
| 1975 | P. Miliotis | 80 | véletlen pontok |
| 1977 | M. Grötschel | 120 | gr120 |
| 1980 | H. Crowder, M. W. Padberg | 318 | lin318 |
| 1987 | M. Padberg, G. Rinaldi | 532 | att532 |
| 1987 | M. Grötschel, O. Holland | 666 | gr666 |
| 1987 | M. Padberg, G.Rinaldi | 1 002 | pr1002 |
| 1987 | M. Padberg, G. Rinaldi | 2 392 | pr2392 |
| 1994 | D. Applegate, R. Bixby, V. Chvátal, és W. Cook | 7 397 | pla7397 |
| 1998 | D. Applegate, R. Bixby, V. Chvátal, és W. Cook | 13 509 | usa13509 |
| 2001 | D. Applegate, R. Bixby, V. Chvátal, és W. Cook | 15 112 | d15112 |
| 2004 | D. Applegate, R. Bixby, V. Chvátal, W. Cook, és K. Helsgaun | 24 978 | sw24798 |
| 2006 | D. Applegate, R. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea és K. Helsgaun | 85 900 | pla85900 |

# Megoldás

1. Módszer - Brute force
megnézzük az összes lehetséges útvonalat, majd kiválasszuk a legrövidebbet

összesen $n!$ lehetséges kombináció

# Megoldás

1. Módszer - Brute force
megnézzük az összes lehetséges útvonalat, majd kiválasszuk a legrövidebbet

összesen $n!$ lehetséges kombináció
ha a számítógép egy másodperc alatt 1 millió Hamilton utat talál meg, akor N=6,7,8 esetén azonnal, N=11 esetén 4 másodperc, ..., N=14 majdnem két óra, N=20 - több mint 1 millió év

| | | | |
|---|---|---|---|
| H,C,D,G,P,T,H | 275 | H,C,P,D,G,T,H | 319 |
| H,C,D,G,T,P,H | 320 | H,C,P,D,T,G,H | 314 |
| H,C,D,P,G,T,H | 291 | H,C,P,G,D,T,H | 261 |
| H,C,D,P,T,G,H | 328 | H,C,P,G,T,D,H | 270 |
| H,C,D,T,G,P,H | 278 | H,C,P,T,D,G,H | 298 |
| H,C,D,T,P,G,H | 270 | H,C,P,T,G,D,H | 312 |
| H,C,G,D,P,T,H | 293 | H,C,T,D,G,P,H | 291 |
| H,C,G,D,T,P,H | 280 | H,C,T,D,P,G,H | 299 |
| H,C,G,P,D,T,H | 251 | H,C,T,G,D,P,H | 349 |
| H,C,G,P,T,D,H | 244 | H,C,T,G,P,D,H | 313 |
| H,C,G,T,D,P,H | 296 | H,C,T,P,D,G,H | 341 |
| H,C,G,T,P,D,H | 302 | H,C,T,P,G,D,H | 297 |

# Christofides algoritmusa - egy megközelítés

garantáltan maximum 3/2-eddel haladja meg az optimális eredményt
Az algoritmus:

- keressük meg a gráf minimális feszítőfáját (T)
- adjunk T-hez éleket úgy, hogy minden csúcs fokszáma párossá váljon (a kínai postás feladatnál használt algoritmus alapján)
- határozzuk meg az Euler kört az így kapott gráfban
- alakítsuk át a zárt Euler vonalat Hamilton körré, úgy, hogy minden csúcsot amiben már jártunk átugorjuk

Egy példa: lásd 9_jegyzet.pdf

# Heurisztikus módszerek

2. Módszer - egy random módszer

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost
jelöljük meg a kiválasztott várost

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost
jelöljük meg a kiválasztott várost
amíg (létezik nem megjelölt város)

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost
jelöljük meg a kiválasztott várost
amíg (létezik nem megjelölt város)
   válasszunk véletlenszerûen egy várost az eddig megjelöltek közül,

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost
jelöljük meg a kiválasztott várost
amíg (létezik nem megjelölt város)
    válasszunk véletlenszerûen egy várost az eddig megjelöltek közül,
    jelöljük meg ezt a várost

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost
jelöljük meg a kiválasztott várost
amíg (létezik nem megjelölt város)
  válasszunk véletlenszerûen egy várost az eddig megjelöltek közül,
  jelöljük meg ezt a várost
  kössük össze a várost az elõzõleg választott várossal

# Heurisztikus módszerek

2. Módszer - egy random módszer
-egy véletlen útvonalat építünk

válasszunk véletlenszerûen egy kiinduló várost
jelöljük meg a kiválasztott várost
amíg (létezik nem megjelölt város)
   válasszunk véletlenszerûen egy várost az eddig megjelöltek közül,
   jelöljük meg ezt a várost
   kössük össze a várost az elõzõleg választott várossal
amíg vége

Hogyan tudnánk ezen az algoritmuson javítani?

Hogyan tudnánk ezen az algoritmuson javítani?

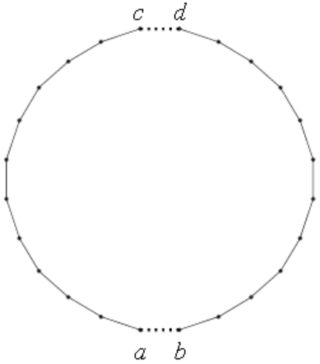
- ismételjük ezt az algoritmust többször, és válasszuk ki a legrövidebb útvonalat

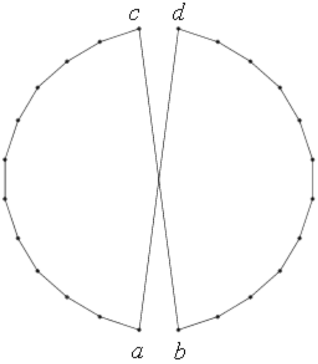
# Legközelebbi szomszédok módszere

legközelebbi szomszédba menjünk tovább - greedy heurisztika

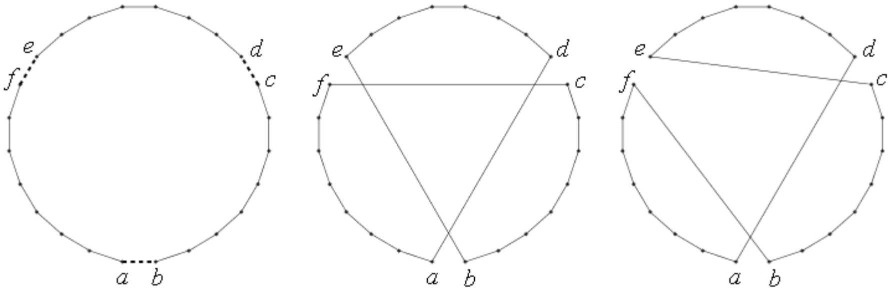
# 2-Opt algoritmus

első lépésben véletlenszerűen generálunk egy útvonalat
A 2-Opt lényege:



# 3-Opt algoritmus

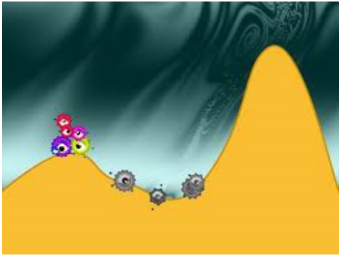



Az általánosítás: Lin-Kerninghen algoritmus

# Metaheurisztikus módszerek

- tabu-keresés
- hegymászó algoritmus
- szimulált hûtés
- genetikus algoritmusok
- ...

# Hegymászó algoritmus



# Hegymászó algoritmus

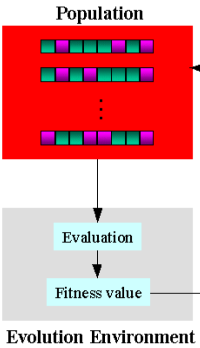
Az alapötlet a következő: kiindulunk egy véletlen megoldásból, majd azon változtatunk egy kicsit, ha a változtatott megoldás jobb, mint az elődje volt, akkor megtartjuk. Addig ismételjük ezeket a lépésket, amíg egy leállási feltétel nem teljesül, ez a leállási feltétel lehet például a következő: maximum lépésszám elérése, nem tapasztalható javulás a megoldásban. Hátránya a hegymászó algoritmusnak, hogy könnyen beragadhat lokális optimumokba, de ennek a kiküszöbölésére léteznek különböző változatai.

$S \leftarrow$ véletlenszerűen kigenerált megoldás
**Amíg** (leállási feltétel nem teljesül)
    $S' \leftarrow S$
    $R \leftarrow$Változtatott$(S')$
    **Ha** Minőség$(R)>$Minőség$(S)$
        $S \leftarrow R$
    **Ha vége**
**Amíg vége**
**visszatérít** $S$

# Hegymászó algoritmus

- Kiindulunk egy kezdeti megoldásból (akár véletlenszerûen megválasztva a városok sorrendjét
- Kiszámoljuk a teljes út hosszát
- Felcseréljük két város sorrendjét, és kiszámoljuk az így kapott hosszt
- Ha jobb eredményre jutunk a 3. lépésben, akkor eltároljuk az így kapott megoldást, mint optimálisabbat
- A megállási kritérium teljesüléséig (pl. n lépéshez értünk) folytatjuk a 3. lépéstõl az eljárást

# Genetikus algoritmuok

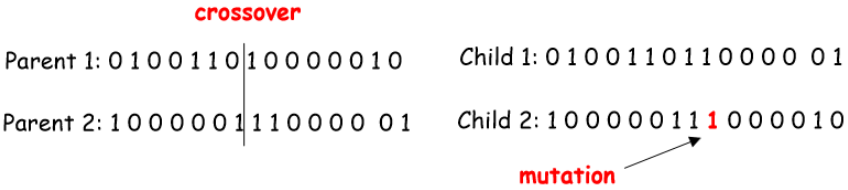



Genetic Algorithm Evolution Flow

# Genetikus algoritmusok

- egy kezdeti populációnk van, ami egyedekbõl áll
- az egyedekbõl új egyedek jönnek létre keresztezõdéssel
- mutáció történhet (valamilyen valószínûséggel)
- kiválasztással létrejön egy új pouláció - evaluáljuk az egyedeket

# Keresztezõdés és mutáció

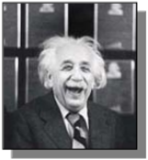


# Egy példa

| Eye color | Hair color | Skin color | height | streng<br>th | Brain size | music<br>ality | coordi<br>nation | maths |
|---|---|---|---|---|---|---|---|---|

Phenotypes of Albert Einstein

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| brown | black | bright | Avg. | Avg. | Avg. | high | low | mediu<br>m |



Phenotypes of Sharon Stone

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| green | blond<br>e | fair | Avg. | Avg. | Avg | high | good | good |

Milyen szempont szerint evaluáljuk az egyedeket? Kik a jobbak?

Milyen szempont szerint evaluáljuk az egyedeket? Kik a jobbak?
Fitnesz függvény

Hogyan alkalmazhatjuk az utazóügynök problémára?

Hogyan alkalmazhatjuk az utazóügynök problémára?
Mik az egyedek?

Hogyan alkalmazhatjuk az utazóügynök problémára?
Mik az egyedek?
Mi a fitnesz függvény?

Hogyan alkalmazhatjuk az utazóügynök problémára?
Mik az egyedek?
Mi a fitnesz függvény?
Hogyan keresztezhetjük az egyedeket?

Hogyan alkalmazhatjuk az utazóügynök problémára?
Mik az egyedek?
Mi a fitnesz függvény?
Hogyan keresztezhetjük az egyedeket?
Hogyan történhet a mutáció?

Részletek: lásd 9_jegyzet.pdf

# Egy példa

Egy példa

# Utazó ügynök más változatai

- Biton Euklideszi utazó ügynök problémája - eleinte csak kelet felé haladhat a legnyugatibb városból, utána pedig csak nyugat felé a legkeletibb városból
- utazó vásárló problémája - mindegyik városban van egy piac, ahol termékek vannak, és egy bevásárló lista, minimalizálni kell az utazás és a vásárlás összköltéségét
- utazó politikus feladata - egy politikus elég ha minden megyéből csak egy városba látogat el
- utazó tolvaj problémája - TSP kombinálva a hátizsák problémával

# Forrásanyag


- Kása jegyzet
- Santanu Saha Ray, Graph Theory with Algorithms and its Applications, Springer, 2013.
- http://web.info.uvt.ro/ mmarin/lectures/GTC/c-09-new.pdf