A gráfszínezés bonyolultságának és a gráf statisztikus tulajdonságainak összefüggései

Szép Tamás, II. éves mérnök informatikus, BME
Mann Zoltán, egyetemi adjunktus, BME-SZIT

2009.11.20.

Intelligens Rendszerek 2009 – Fiatal Kutatók 4. Szimpóziuma (IRFIX'09)

szep.tamas@mail.datanet.hu, zoltan.mann@gmail.com

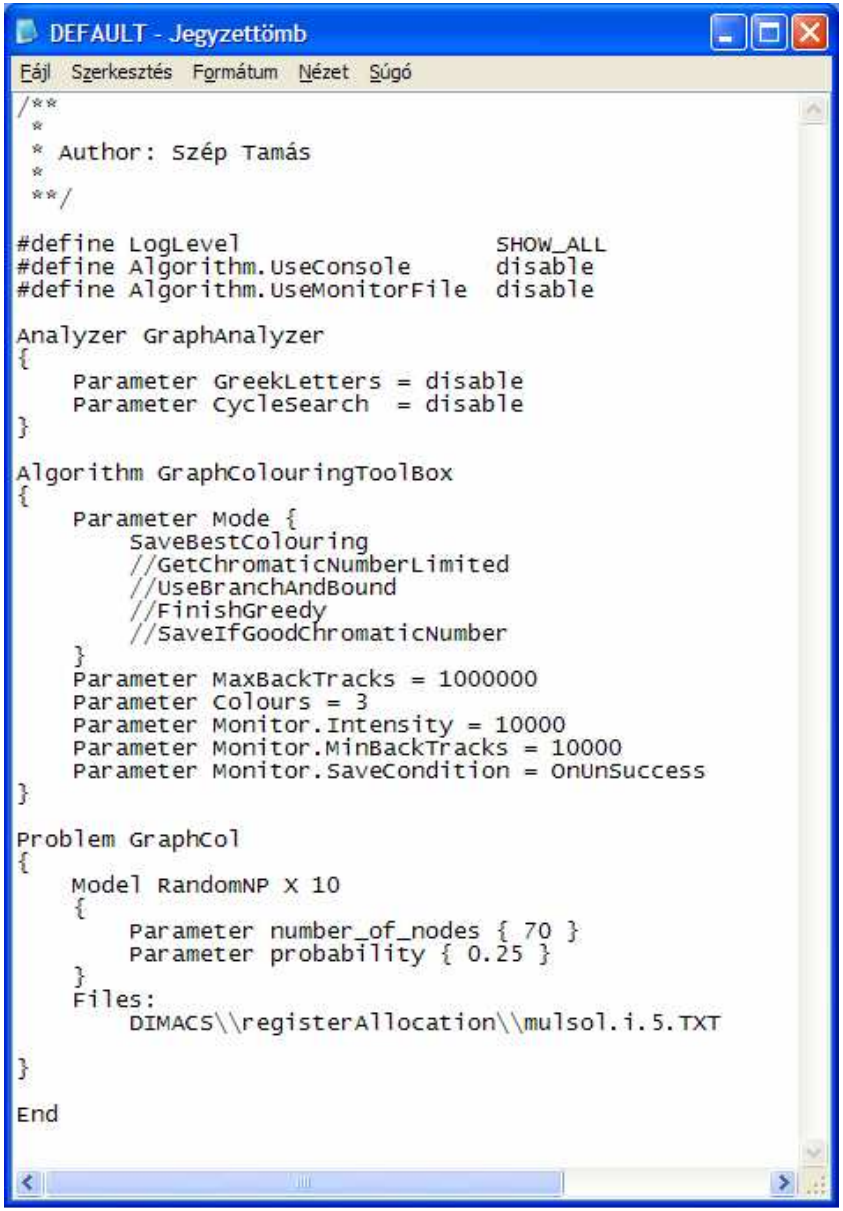
Bevezetés

A gráfszínezést igen sok helyen alkalmazzák, ismertebb felhasználási területek:

- Regiszter allokáció
- Feladatütemezés (Task scheduling)
- Mobil távközlési hálózatok csatorna kiosztása
- NYÁK-tesztelés
- Biológiai, régészeti adatok analízise
- Sudoku

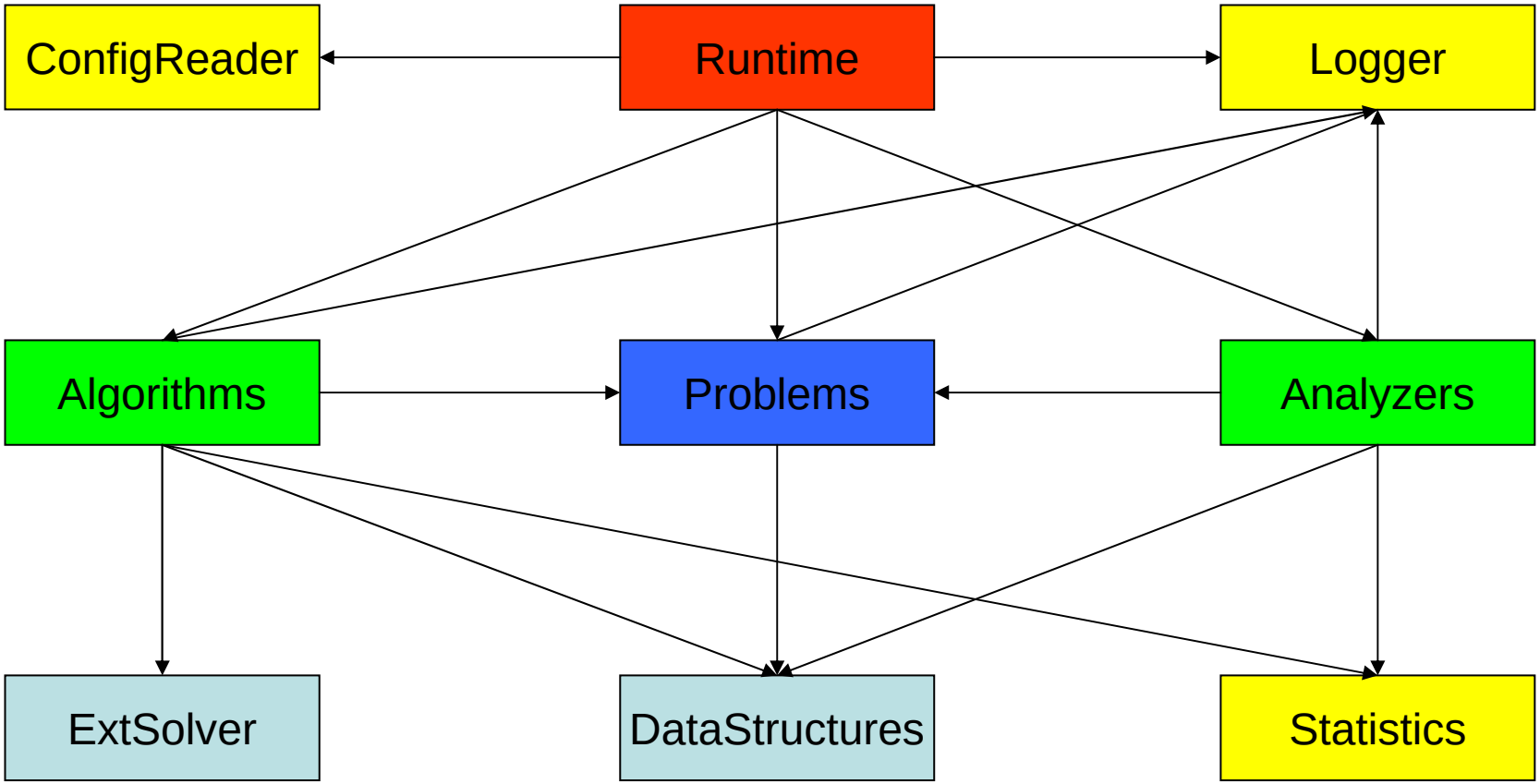
A gráfszínezés NP-teljes, nagy a variabilitás a futásidőben. A klasszikus bonyolultságelmélet nem ad arra választ, hogy egy konkrét problémapéldány megoldása egy adott algoritmussal mennyi ideig tart. Ezért empirikus úton vizsgáltuk a kérdést. A kapott eredmények alapján tételeket bizonyítottam illetve kvalitatív magyarázatokat adtam.

A BCAT-ről



- Budapest Complexity Analysis ToolKit
- Optimalizálási és eldöntési problémák implementálhatóak benne
- Elkülönülő matematikai absztrakciókat megvalósító osztályok
- Loggolás, kivételbiztonság, egységes output
- A rendszer egy saját nyelvet használ, ebben konfigurálható
- Nem probléma- vagy algoritmusspecifikus – absztrakt algoritmusok
- Az algoritmusok működése megfigyelhető Monitor-osztályokkal

A BCAT felépítése



A használt gráfszínező Branch and Bound

- Egyszerű Branch and Bound a szokásos gráfszínezési és CSP-heurisztikákkal és egyéb kiegészítésekkel
- Meghiúsulás előre és fokszám heurisztika változókiválasztáskor
- Minimális konfliktusra törekvés értékkiválasztáskor
- Élkonzisztenciaterjesztés
- Preprocesszálás: egy klikk megszínezése
- Egyéb trükkök a felesleges keresési ágak levágására

A színszám és a bonyolultság összefüggései

- Branch and Bound algoritmusnál, az általam használt heurisztikák esetén bizonyítottuk:

$$k_1 < k_2 < \chi(G) \Rightarrow BT(k_1) \leq BT(k_2)$$

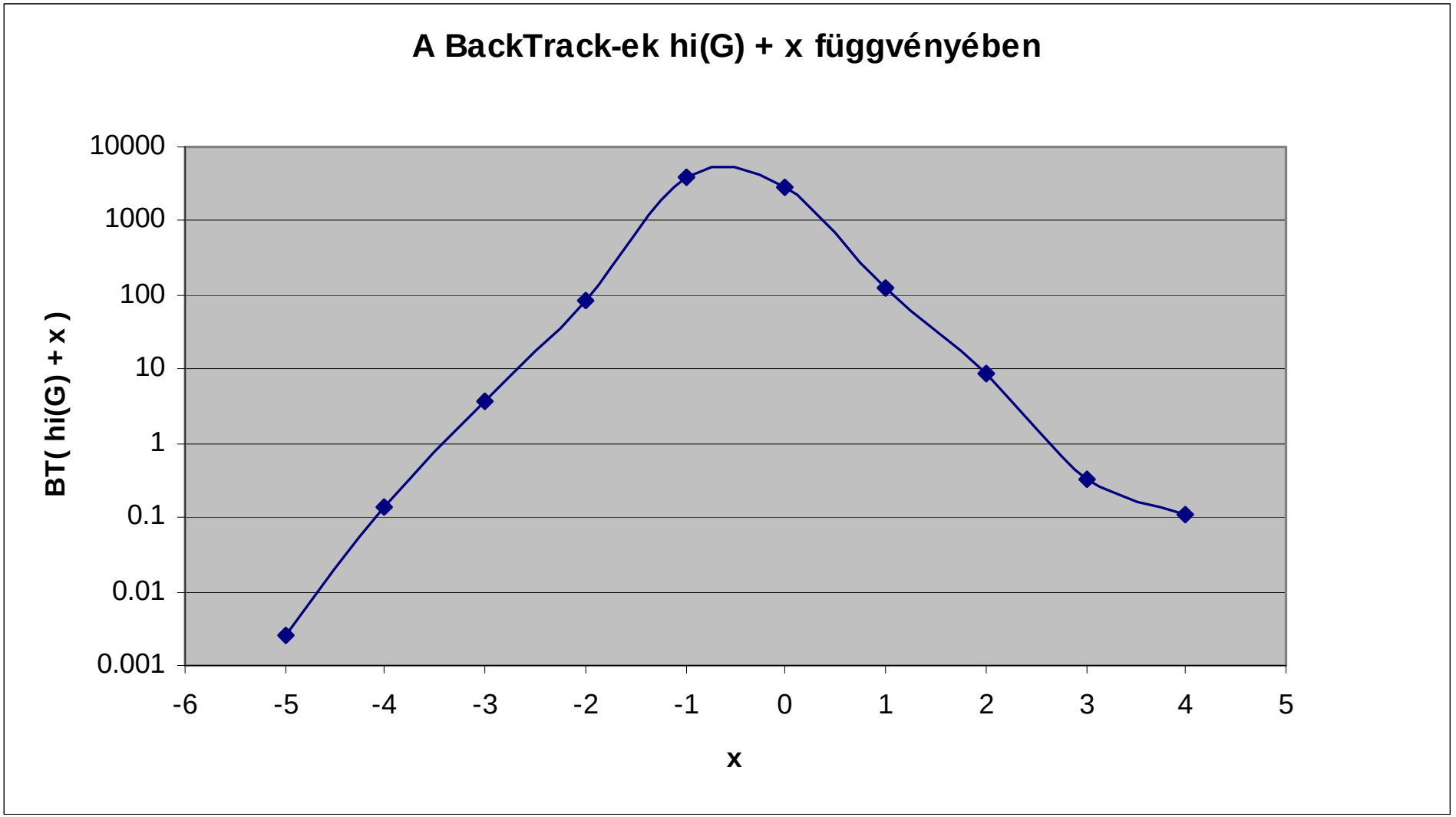
- A gyakorlatban ekkor azonban az is igaz, hogy:

$$BT(k_1) \square \; BT(k_2)$$

- Létezik ellenpélda, de általában az is igaz lesz, hogy:

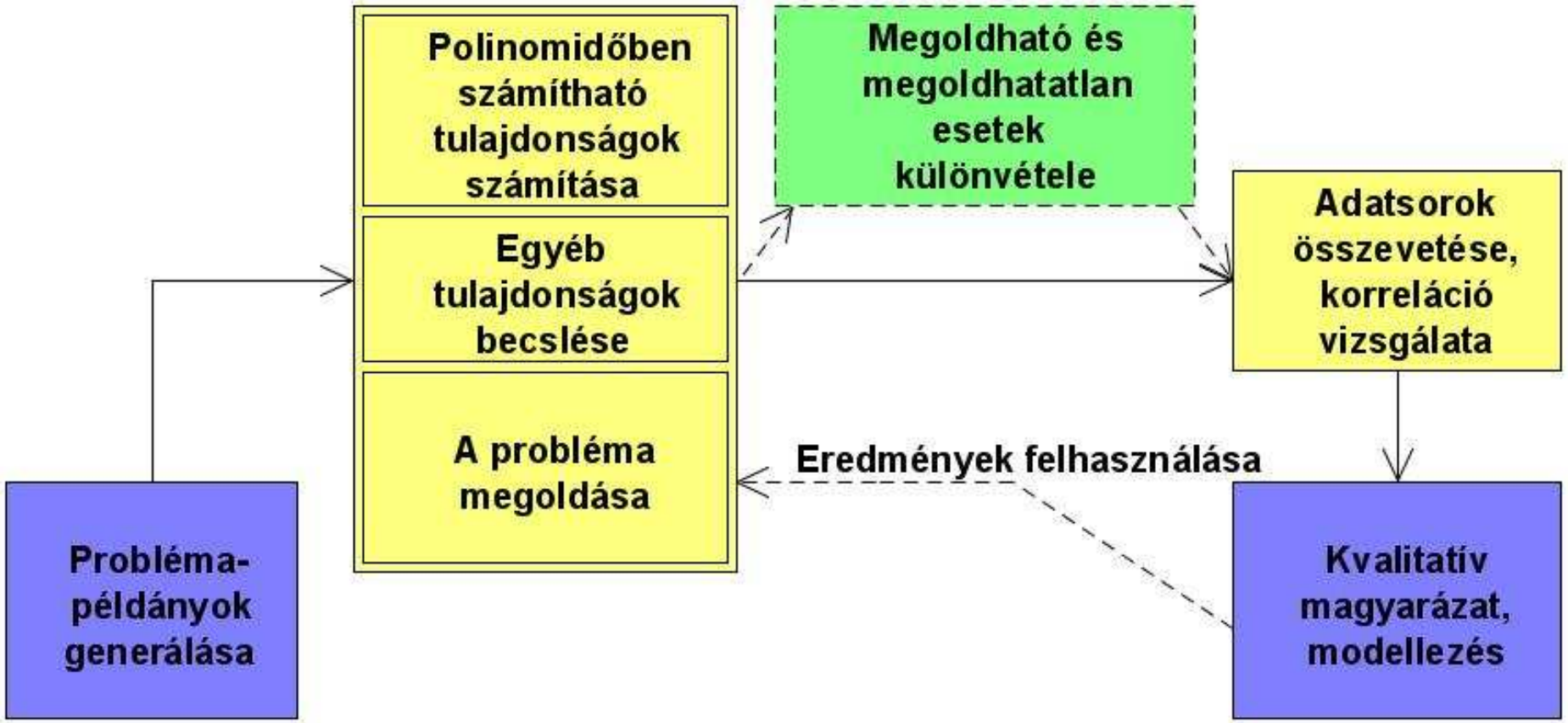
$$\chi(G) \leq k_1 < k_2 \Rightarrow BT(k_1) \square \; BT(k_2)$$

Itt BT a Branch and bound algoritmus futása során történő visszalépéseket jelenti, k pediga probléma inputjának a színek számát megadó részét képezi.



- Gyors felső- és alsókorlátbecslő algoritmusok
- Egzakt kromatikus szám meghatározó algoritmus
- Hibrid Branch and bound – mohó algoritmus

Módszer



- A vizsgálatokat itt random(n, p) gráfokon végzem, ezek tartalmaznak véletlenszerű elemet és könnyen megkonstruálhatók

A gráf néhány jellemző mérőszámát becslő algoritmus

- Klikk-kereső algoritmus: fix számú iterációval a következőt végzi. Kiindul egy csúcsból, és minden lépésben az előző csúcshalmazhoz hozzáveszi azt a legnagyobb fokú csúcsot, ami a csúcshalmaz minden csúcsával össze van kötve.
- Független csúcshalmaz-kereső algoritmus: Veszi a gráf komplementerét, ebben klikket keres.
- Körkereső algoritmus: utakat keres úgy, hogy egy újraindítási blokkban, fix maximális visszalépéssel egy Branch and bound-ot végez, ahol nincs változókiválasztás, csak értékkiválasztás. Az így kapott utak mentén próbál meg köröket keresni. Amennyiben maximális kört keres, akkor csak „ritkán” kell megvizsgálnia, hogy létezik-e az út végéről kiindulva - az út egy részét felhasználva - kör, és Hamilton-kör megtalálása esetén azonnal kiléphet.

Fontosabb korrelációk

Tulajdonság/Eset	Nem megoldható	Megoldható
Karakterisztikus úthossz	Közepesen erős **+**	Közepesen erős **-**
Klaszterezettség	Közepesen erős **-**	Gyenge **+**
Fokszám tulajdonságai	Nem látható korreláció	Nem látható korreláció
Maximális és átlagos klikkméret	Erősebb **-**	Nem látható korreláció

A random gráfok köreiről

- A fent ismertetett körkereső algoritmus igen hatékony random(n, p) gráfok esetén. Ez azért van így, mert még viszonylag ritka random(n, p) gráfokban is nagy köröket találunk.

- Ebből kiindulva látható, hogy legnagyobb körének mérete nem lesz jó jelzője a gráfszínezés bonyolultságának.

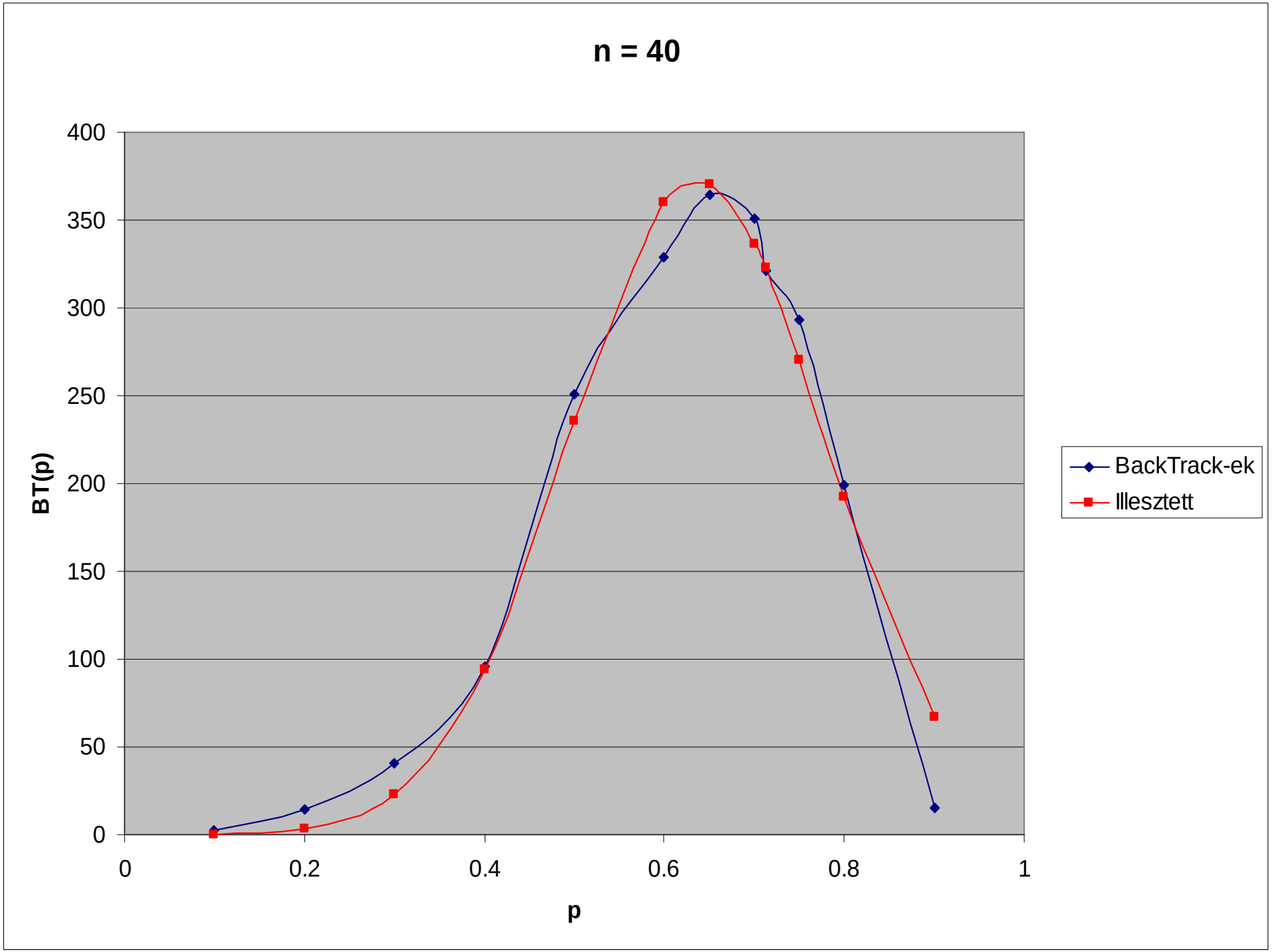
- Itt bizonyítottuk a következő állítást:

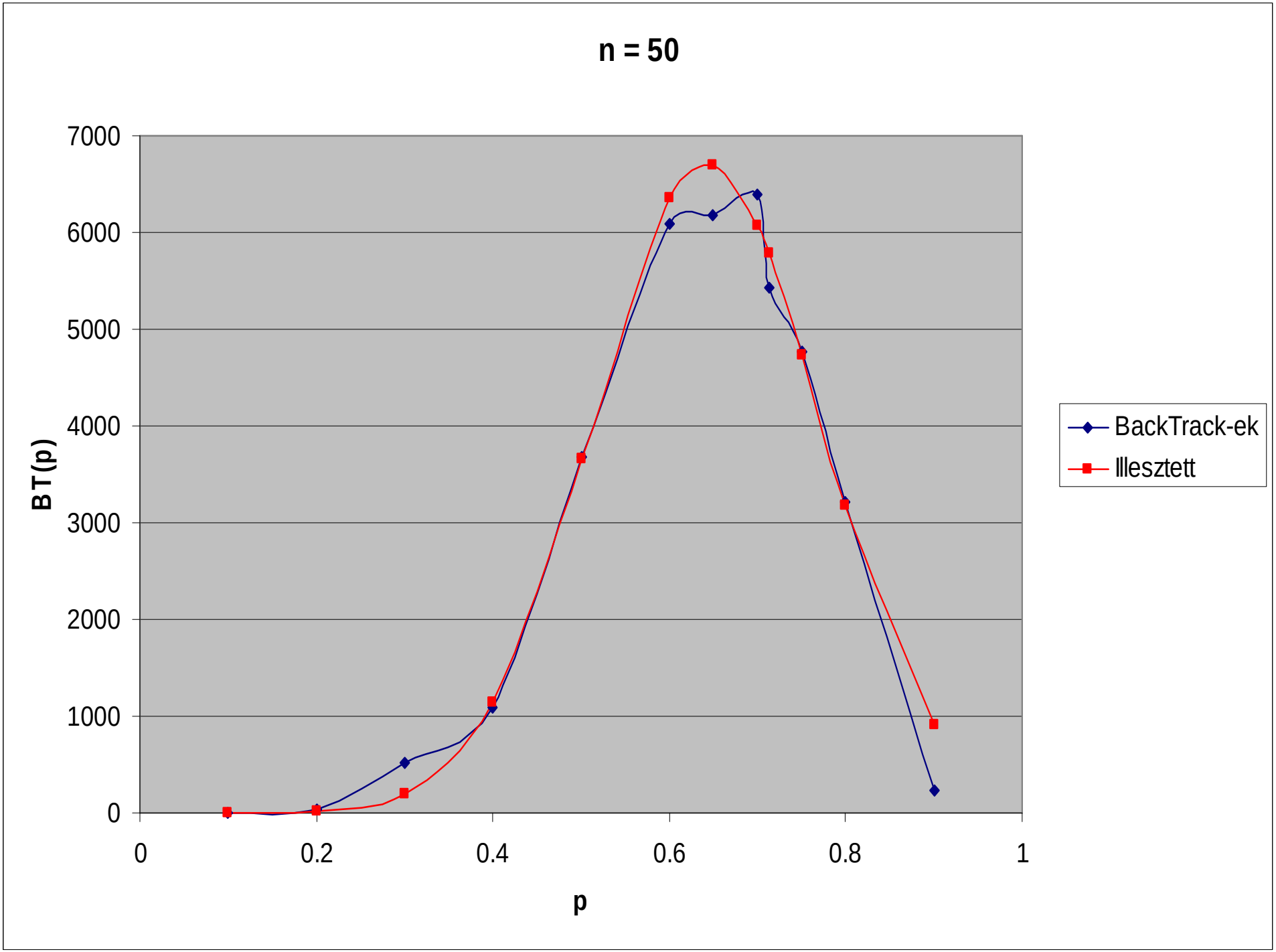
  Ha G egy $random(n, p)$ gráf, akkor:

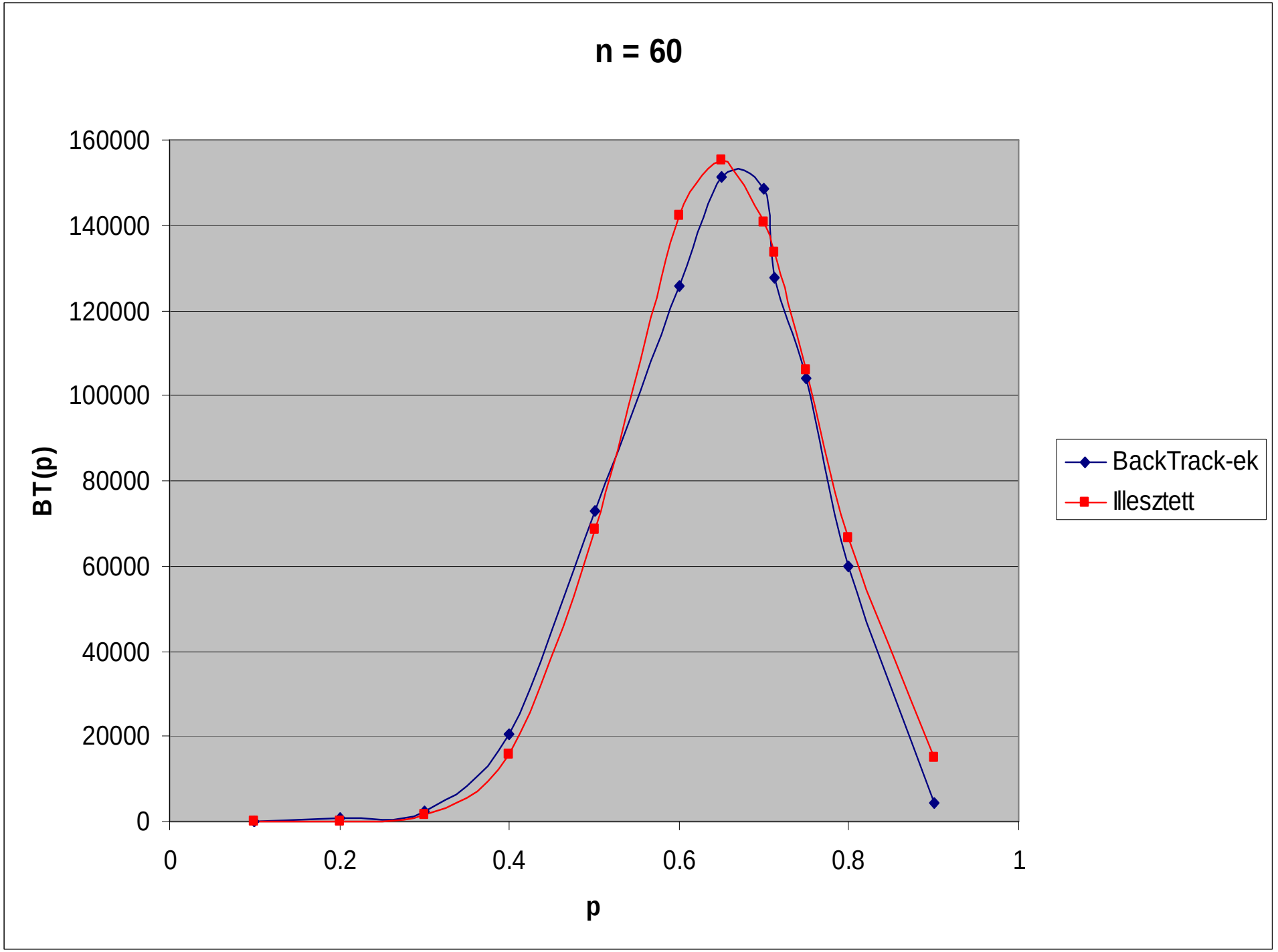
  $p \neq 0 \Rightarrow \exists\, G$-ben Hamilton-kör **whp**.

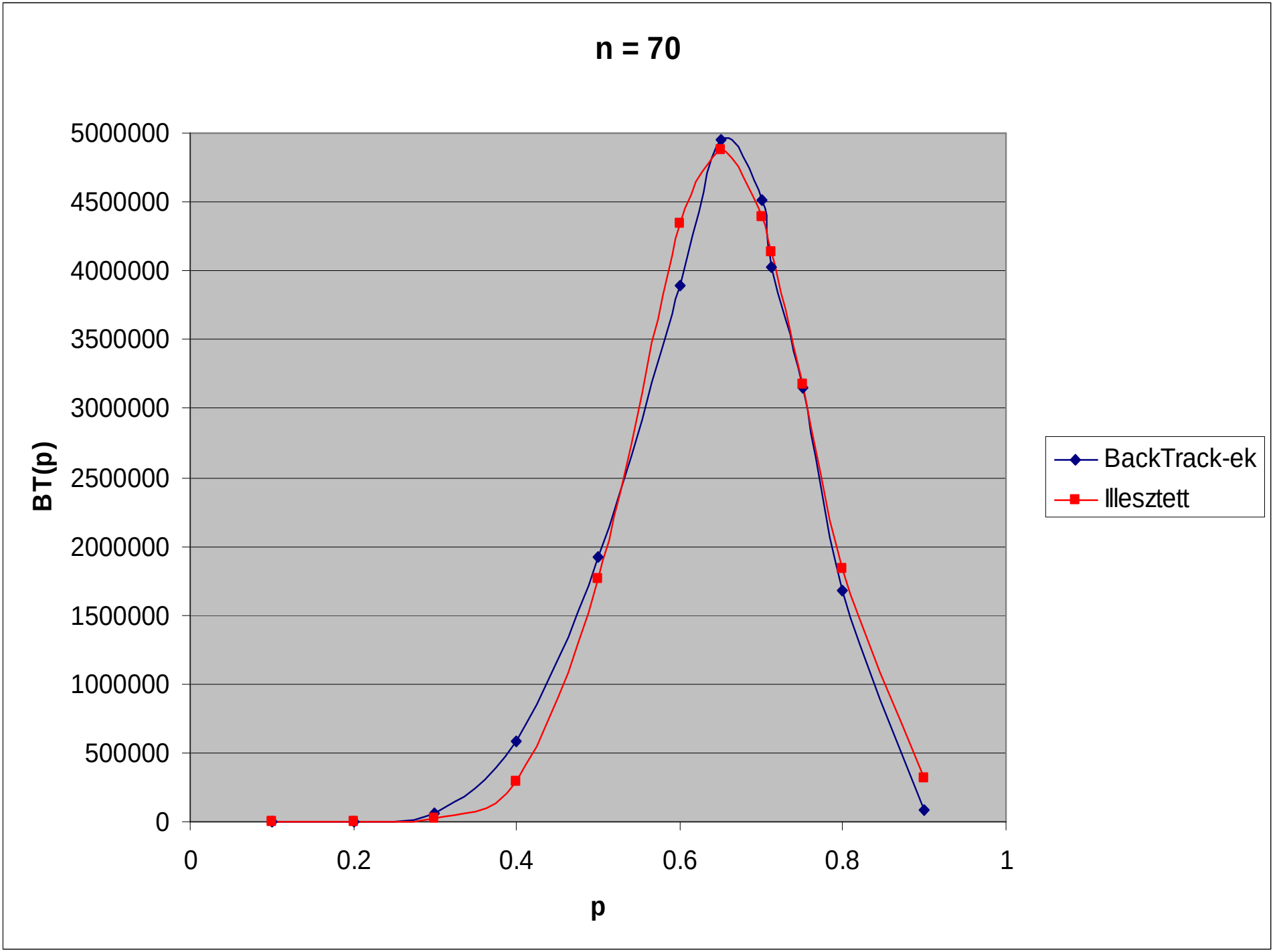
A grafikonokról

- A következő grafikonok elkészítéséhez minden eredménynél 1000db random(n, p) gráf mért értékeinek az átlagát vettem alapul
- Az illesztett görbe egy haranggörbe
- Nagyon erős korreláció a két görbe között: 99-99,5%
- A random(n, p) gráfoknál egy adott (n, p) párra nem tapsztalunk extrém méretű szórást a gráfszínezés bonyolultságában
- Ha egy gráf a tulajdonságai alapján jól modellezhető random(n, p) gráfként, akkor színezésének bonyolultságára nagyságrendben egész jó becslést adhatunk


n = 40
400
350
300
250
200
150
100
50
0
BT(p)
0
0.2
0.4
0.6
0.8
1
p
BackTrack-ek
Illesztett


n = 50
7000
6000
5000
4000
3000
2000
1000
0
BT(p)
0
0.2
0.4
0.6
0.8
1
p
BackTrack-ek
Illesztett


n = 60
160000
140000
120000
100000
80000
60000
40000
20000
0
BT(p)
0
0.2
0.4
0.6
0.8
1
p
BackTrack-ek
Illesztett


n = 70
BT(p)
5000000
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
0
0.2
0.4
0.6
0.8
1
p
BackTrack-ek
Illesztett