

# Kombinatorikus optimalizálási sejtések számítógépes ellenőrzése

Kelemen Ádám Olivér

2021. december

A projekt célja kombinatorikai sejtések ellenőrzése számítógépes módszerekkel, mely a C++ nyelvet és a LEMON library használatát jelentette, így a félévi munkának ezek megismerése tette ki a jelentős részét. A programozói tapasztalat előfeltétele volt a projektnek, amivel rendelkezem is, de ennek csak kisebb százalékát adta a C++ és már évek óta nem foglalkoztam vele, ami nehezítette a feladat elindítását. Újdonság volt számomra a CMake és az, hogy hogyan épül fel egy C++ projekt, ami nem csak néhány fájl tartalmaz. Ezzel párhuzamosan kezdtem el foglalkozni a LEMON-nal, amihez tanuló feladatnak a library-n belüli Dijkstra algoritmus futtatását kaptam.

A félévre kitűzött cél a Woodall-sejtés – pontosabban egy speciális esetének – a számítógépes ellenőrzése kis gráfokra, melyhez egy idén tavasszal megjelent cikk nyújtott segítséget. A sejtés azt mondja ki, hogy ha  $D$  irányított gráf, melyben a legkisebb irányított vágás mérete  $k$ , akkor  $D$ -ben van  $k$  db diszjunkt dijoin. Egy  $D = (V, A)$  digráf esetén, ha a csúcsainak van olyan  $X \subseteq V$  részhalmaza, melyből nem vezet ki él, akkor az  $X$ -be belépő élek halmazát *irányított vágásnak* nevezzük. Egy *dijoin* (vagy *kötés*) alatt az élek egy olyan részhalmazát értjük, mely  $D$  minden irányított vágását lefogja, azaz minden irányított vágásból tartalmaz legalább egy élt. A sejtést  $k = 2$ -re már igazolták, de nagyobb  $k$  értékekre eddig nem, sőt, már a sejtés következő gyengítése is nyitott. Nézzük a  $k = 3$  esetet, ekkor az eredeti sejtés szerint már annyi elég, hogy  $D$  legkisebb irányított vágása 3 méretű legyen ahhoz, hogy  $D$  tartalmazzon 3, páronként éldiszjunkt dijoin-t. Egyelőre viszont arra sincs válasz, hogy meg lehet-e adni egy olyan fix, egész  $k$  alsó korlátot, mely esetén bármely gráfra igaz, hogy ha minden irányított vágása legalább  $k$  méretű, akkor a gráf biztosan tartalmaz három, páronként éldiszjunkt dijoin-t.

A sejtés ezen változatának a síkgráfok egy speciális esetére való megszorítása még szintén nyitott. A síkgráfok duálisát felhasználva fogalmazható át a sejtés, ehhez bevezetjük a *feedback arc set* fogalmát, mely a digráf éleinek egy olyan részhalmaza, ami minden irányított körből legalább egy élt tartalmaz. Meggondolható, hogy ezáltal, ha a gráfból elhagyom a feedback arc set-be tartozó éleket, akkor a gráf körmentes lesz, azaz DAG. Tehát, ha  $D$  egy síkbarajzolható digráf, melynek az összes irányított köre legalább  $k$  hosszú, akkor  $D$  tartalmaz  $k$  db, páronként diszjunkt feedback arc set-et. Ez a sejtés bizonyítottan sérül, ha a síkbarajzolhatóság feltételét nem kötjük meg, de  $k = 3$  esetén sem bizonyított még.

Ennek a sejtésnek a  $k = 3$  esetét felhasználva kimondható még egy sejtés, melynek kis gráfokon való számítógépes ellenőrzése a félévem feladatát képezte. Ha  $D$  irányított, erősen összefüggő gráf, akkor tartalmaz három, páronként diszjunkt feedback arc set-et. Ehhez első lépésként egy feedback arc set megtalálását tűztük ki célul, melyhez a korábban említett cikket<sup>1</sup> kezdtem el feldolgozni. A cikkben több megoldás is található minimum feedback arc set keresésére és az alap ötlet az volt, hogy egy gráfalgoritmust valósítsak meg, de arra jutottunk, hogy ez több rutint igényel mind a C++, mind a LEMON esetén, így a cikk elején található, háromszög-egyenlőtlenséget felhasználó IP feladat programozásába kezdtem bele.

A feladat során alkalmam nyílt a LEMON LP részével is foglalkozni, amit össze kellett kapcsolni a korábbi Dijkstra algoritmus futtatásához használt gráf elemekkel, ezáltal is nagyobb képet kaptam a LEMON-ról. Az IP feladat megoldásához a Cplex-et használtam, aminek az integrálása egy hosszantartó kódelemzést hozott magával, mert Mac számítógépet használok és a LEMON erre nem volt felkészítve, így nem találta meg a Cplex-hez tartozó fájlokat. Ehhez végül a LEMON-hoz tartozó CMake-et kellett kicsit módosítani, de a megoldás keresésével töltött idő alatt értettem meg jobban a CMake projekt felépítését és működését, így ennek a keretében az én projektem is átépítésre került és érte el a jelenleg is fennálló formáját. Az IP feladat implementációja is eltér kissé a cikkben leírt modellhez képest, mert abban egy külön konstanst vezettek be a két csúcs közötti él létezésének és ebben az esetben az irányának a tárolására, de a LEMON-ban a csúcs és él iterátorok felhasználásával erre külön nem volt szükség.

A következő lépés ennek a modellnek a módosítása volt, hogy három diszjunkt feedback arc set keresését végezze el. Eddigre már biztosabban mozogtam a C++-ban is, így a modell felépítésével egy kód refaktorálást is elvégeztem, viszont a modell átalakításánál több ponton is Péter iránymutatására támaszkodtam. A modell végül elkészült és a kódban lévő hibákat is sikerült megszüntetni, így a program alkalmas lett arra, hogy sejtést ellenőrizzen, azaz a kis gráfokon való tesztelésre. A Nauty segítségével Péter legenerálta nekem 2-től 6 csúcsig az összes irányított síkgráfot, melyekre a tesztet elvégezhettem. A program több, mint 17 ezer gráfot ellenőrzött és ezeken igazolta is, hogy valóban tartalmaznak három, páronként diszjunkt feedback arc set-et, azaz a legfeljebb 6 csúcsú irányított síkgráfok között nincs ellenpélda.

A továbbiakban meg lehet próbálni ezt kiterjeszteni, a 7 és 8 csúcsú gráfok még elérhetőek lehetnek. A többszálú feldolgozás mentén lehet még magasabb csúcyszámig is el lehet jutni. Szeretnék implementálni egy gráfalgoritmust C++-ban, ami most még korai lett volna, de a későbbiekben tovább lehetne vinni a feedback arc set témakört egy minimum FAS keresővel, akár a LEMON keretein belül.

A projekt ezen szakaszában inkább a programozás szerepelt nagyobb hangsúllyal, de úgy érzem hasznos volt és a későbbi munka szempontjából szükséges is.

---

<sup>1</sup>An Exact Method for the Minimum Feedback Arc Set Problem