

Heurisztikus módszerek alkalmazása az S-gráf keretrendszerben

1. BEVEZETÉS

Az S-gráf keretrendszer^[2] egy ütemezési feladatokra kifejlesztett, gráf alapú, egzakt megoldási módszertan. Az operációkutatáson belül az ütemezés a kezdetektől fogva fontos és ma is aktív kutatási terület. Számos különféle ütemezési feladat merül fel sok különböző alkalmazási területen, például ipari gyártófolyamatok tervezésénél, logisztikában, projektmenedzsmentben, tömegközlekedésben és humán erőforrás menedzsmentben.

A szakirodalomban sokféle megoldási módszer jelent meg ütemezési feladatokra. Köztük vannak egzakt és heurisztikus módszerek is. Egzakt optimalizálási módszer alatt azt értjük, hogy az összes lehetséges megoldás közül bizonyítottan képes megtalálni a célfüggvény szerinti legjobbat, azaz az optimumot. Ez azonban önmagában nem elegendő, hiszen ezt elérhetjük úgy is, hogy az összes lehetséges megoldást megvizsgáljuk, ami viszont a nagy számítási igény miatt a gyakorlatban már kis példákön sem kivitelezhető elfogadható időn belül. Ezért olyan technikákat kell alkalmazni, melyek csökkentik a keresési teret, azaz a megvizsgálandó esetek számát. Míg az egzakt módszerek olyan technikákat alkalmaznak erre, melyek bizonyítottan megtartják az optimális megoldást – ha több is van, akkor legalább egyet –, addig a heurisztikus módszerek nem rendelkeznek ezzel a tulajdonsággal. Emiatt nem garantált, hogy megtalálják az optimális megoldást, de nagyobb feladatokra is belátható idő alatt optimálishoz közeli megoldást tudnak adni.

Mindkét megközelítésnek megvan az előnye, ezért nagyon ígéretesek a hibrid módszerek, melyek kombinálják a két technikát. Erre sokféle lehetőség van, közülük az egyik, hogy az egzakt keresési algoritmust heurisztikus szabályok szerint irányítjuk a keresési térben. Jelen tanulmányban ennek az alkalmazását mutatom be az S-gráf keretrendszerben.

2. AZ S-GRÁF MÓDSZERTAN

Az S-gráf alapú módszer már több különféle ütemezési feladatra ki lett terjesztve

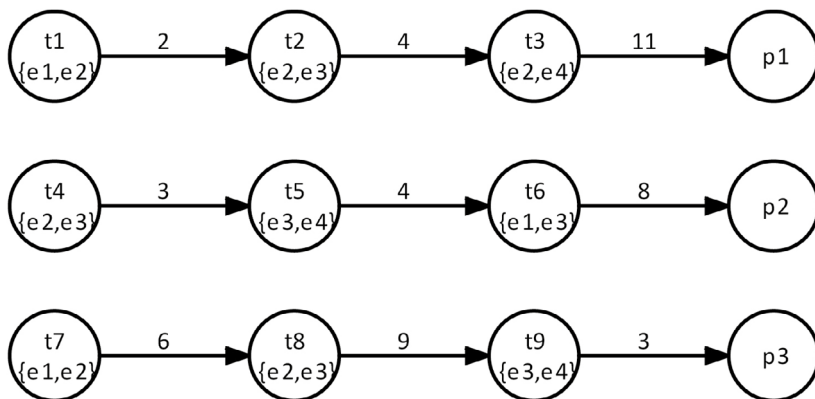
[1] SZE-GIVK Informatika Tanszék, egyetemi tanársegéd. Témavezető: Dr. Hegyháti Máté, SZE-GIVK Informatika Tanszék, tudományos főmunkatárs.

[2] SANMARTÍ, EDUARD – PUIGJANER, LUIS – HOLCZINGER TIBOR – FRIEDLER FERENC: Combinatorial framework for effective scheduling of multipurpose batch plants, in *AIChE Journal*, 2002/11., vol. 48., 2557–2570.

az eredeti algoritmus megjelenése óta. Ezeknek sajátos bemeneti paramétereik és korlátozásaik lehetnek,^[3] de a következő alapvető elemekben nagy részük megegyezik: adottak gyártási munkafolyamatok (taszkok), ezek végrehajtására képes erőforrások (berendezések), és el kell dönteni, hogy mikor melyik berendezés melyik taszkot hajtsa végre. Gyakori megkötés, hogy egyes gyártási lépéseket megadott sorrendben kell végrehajtani. A cél pedig többféle lehet, de a leggyakoribb, hogy a befejezési időt kell minimalizálni.

Az S-gráf módszerrel egy súlyozott irányított gráf modellt használ az időzí-
tési relációk reprezentálására. A csúcsok az időzítendő eseményeket (a taszkok
elkezdését és a termékek elkészülését) jelölik, a köztük lévő élek pedig az időkül-
önbségeket. A bemeneti paraméterekből és a rögzített sorrendű taszkpárokból
felépíthető a receptgráf, amelyre az 1. ábra mutat egy példát. A csúcsokon a taszk
neve alatt a végrehajtásra alkalmas berendezések halmaza található.

1. ábra: Receptgráf



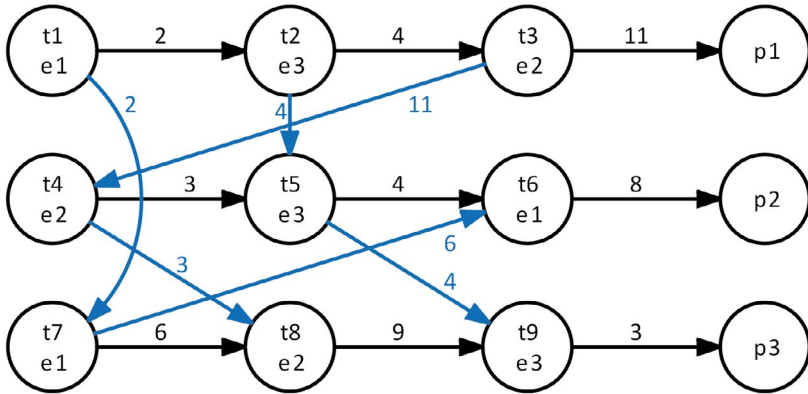
A megoldó algoritmus a branch-and-bound (szétválasztás és korlátozás) eljárás
alapul. Ennek az a lényege, hogy a megoldásteret egyre kisebb részhalma-
zokra bontjuk fel (szétválasztás), és a halmazokra kiszámoljuk a benne található
megoldások célfüggvény értékeinek elméleti korlátját (korlátozás). Ha ez a korlát
nem jobb, mint egy már megtalált megoldás értéke, akkor a teljes részhalmazt
kizárhatjuk a keresésből.

A receptgráf nem tartalmaz ütemezési döntést, innen indul az algoritmus.
A szétválasztási lépés során pedig egy új döntést hozunk meg, melynek lehetsé-
ges kimenetelei alapján újabb részhalmazokat kapunk. Például eldöntjük, hogy
a t1 taszkot az e1 vagy e2 berendezés fogja végrehajtani. Az azonos berendezések

[3] HARJUNKOSKI, IRO (ed.): Scope for industrial applications of production scheduling models and
solution methods, in *Computers and Chemical Engineering*, 2014, vol. 62., 161–193.

által végrehajtott taszkoknak pedig a sorrendjét is el kell dönteni, amit további élek hozzáadásával jelzünk a részalmazokat reprezentáló gráfokban. Ha minden ilyen döntést meghoztunk, akkor egy megoldást leíró ütemezési gráfot kapunk, mint ami a 2. ábrán látható. A kék élek az ütemezési döntéseknek megfelelő sorrendet állítják be, és kizárják, hogy két azonos berendezéshez rendelt taszk végrehajtása között időbeli átfedés legyen.

2. ábra: Ütemezési gráf



A köztes S-gráfokban az irányított utak súlyai alsó korlátot adnak a befejezési időre, hiszen újabb élek hozzáadásával nem rövidülhetnek az utak és az általuk reprezentált időkülönbségek. Ha kör keletkezne, az végtelen súlyú útnak felel meg, és azt jelentené, hogy egy taszknak önmaga után kellene elkezdődnie. Ebben az esetben kizárható a teljes részalmaz, mert nem tartalmaz érvényes megoldást.

A fentebb felvázolt megoldó algoritmusban két helyen hozhatunk döntést a keresésre vonatkozóan:

1. Melyik részalmaz kerüljön szétválasztásra?
2. Milyen ütemezési döntés szerint bontsuk további részalmazokra?

Jelen kutatásban a 2. pontra koncentráltam, de az 1. pont is egy ígéretes jövőbeli kutatási téma, melyet már vizsgáltak^[4] más feladatoknál alkalmazott branch-and-bound algoritmusokon, és sikeresen növelték a teljesítményüket heurisztikus módszerekkel.

Az elsőként publikált S-gráf megoldó algoritmus a berendezés alapú szétválasztási módszert alkalmazza:

[4] GUPTA, K. OMPRAKASH – RAVINDRAN, A.: Branch and Bound Experiments in Convex Nonlinear Integer Programming, in *Management Science*, 1985/12., vol. 31., 1533–1546.

1. Válasszunk egy berendezést, amihez még van beütemezetlen taszk.
2. Minden lehetséges taszkhoz tartozzon egy új részhalmoz, melyben ezt a taszkot a választott berendezés hajtja végre következésként a már hozzárendelt taszkok után.
3. Ha minden lehetséges taszk végrehajtható más berendezéssel is, legyen egy olyan részhalmoz is, ahol a választott berendezés nem hajt végre több taszkot.

Ez a módszer további döntésre ad lehetőséget az 1. lépésében, mert nem határozza meg, hogy mi alapján válasszuk ki a berendezést. Nemes Zsolt Ádám azt vizsgálta szakdolgozatában,^[5] hogy mekkora hatása van a megoldási folyamat hosszára a berendezés kiválasztási módszere. A vizsgálatok alapján nagyságrendi különbségek lehetnek a megoldási időben különböző kiválasztási stratégiák esetén. Dolgozatában 12 heurisztikus kiválasztási stratégiát javasolt, melyeket a következő fejezetben mutatok be a továbbfejlesztéseimmel együtt.

3. HEURISZTIKUS BERENDEZÉS KIVÁLASZTÁS

A korábbi módszerek a megoldási folyamat kezdetén határoztak meg egy sorrendet, amely alapján kiválasztásra kerülnek a berendezések az algoritmusban. A rendezés alapjául heurisztikus stratégiák szolgálnak, melyek a bemeneti adatok alapján olyan sorrendet próbálnak meghatározni, amely segíti a megoldástér hatékony bejárását. Ehhez a berendezések terhelése kerül kiszámításra, különböző módokon.

A 12 stratégia az alábbi 3 paraméter kombinációiból jön létre:

1. Összehasonlítandó érték
 - a. Taszkok száma (T)
 - b. Taszkok végrehajtási időinek összege (P)
2. A számítás során figyelembe vett taszkok
 - a. A csak az adott berendezéssel végrehajtható taszkok (F)
 - b. Az adott berendezés által végrehajtható összes taszk (A)
 - c. Az adott berendezés által végrehajtható taszkok, az alkalmas berendezések számának reciprokával súlyozva (P)
3. Rendezés iránya
 - a. Növekvő (A)
 - b. Csökkenő (D)

Ezekből alakul ki a következő 12 stratégia: TFA, TFD, TAA, TAD, TPA, TPD, PFA, PFD, PAA, PAD, PPA, PPD.

A heurisztikus elveknek köszönhetően ezek a stratégiák általában jobbak, mint egy véletlenszerű sorrend követése. Azonban a 12 sorrend közül nincs

[5] NEMES ZSOLT ÁDÁM: *S-gráf megoldó heurisztikus gyorsításai*, 2014, Mérnök informatikus BSc Szakdolgozat, Pannon Egyetem Műszaki Informatikai Kar Rendszer- és Számítástudományi Tanszék.

egyértelmű legjobb. Feladattól függően más-más stratégiával érhető el leggyorsabban az optimális megoldás, és eddig nem sikerült egyértelmű összefüggést találni a feladat paraméterek és az ideális sorrend között. Egy jövőbeli kutatási irány ezen kapcsolatok feltárása a gépi tanulás eszközeivel, és a stratégia kiválasztása neurális hálózat segítségével.

Mivel a berendezések terhelése egy jó mutatónak bizonyult a kiválasztási stratégiában, kutatásom egyik része annak megvizsgálása volt, hogy javíthatók-e a megoldási idők, ha már meghozott ütemezési döntések információival kiegészülnek a fenti stratégiák. A megoldás során egyre fogynak a még nem ütemezett taszkok, így ezekre korlátozva a terhelési mutatók kiszámítását több információval rendelkezünk a berendezéseket érintő hátralévő döntésekről. Ugyanakkor minden szétválasztási lépésnél újra el kell végezni a rendezést, ami nagyobb számítási igényű, mint a korábbi megoldás, ahol ezt csak egyszer kellett megtenni, induláskor.

Kutatásom másik fejlesztési kísérletének célja a legjobbnak bizonyult kiválasztási stratégiák kombinálása. Ahelyett, hogy az algoritmus végig minden berendezés kiválasztásnál ugyanazt a heurisztikát alkalmazná, a továbbfejlesztett módszer váltakozva alkalmazza a legjobb stratégiákat. Így várhatóan jobb lesz a teljesítmény, mintha olyan stratégiát választottunk volna, amely az adott példára éppen nem megfelelő, és közel lesz ahhoz az esethez, mintha az ideális stratégiát választottuk volna.

4. TELJESÍTMÉNYTESZT

A továbbfejlesztéseket implementáltam a megoldó szoftverbe és teljesítményüket szakirodalmi példákon ellenőriztem. A tesztek egy Intel i7-8750H processzorral felszerelt laptopon futottak.

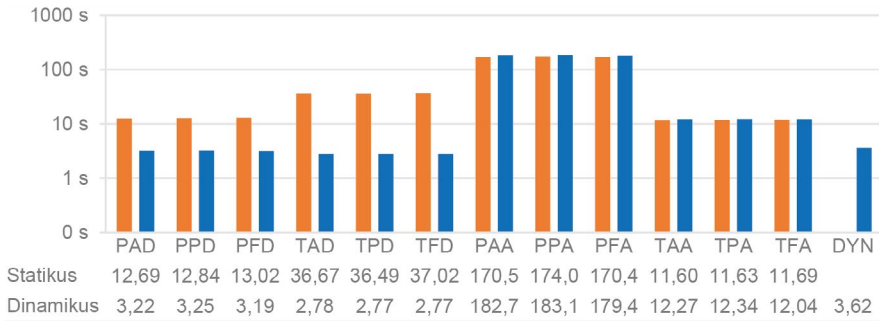
A váltakozó stratégiát alkalmazó heurisztika (DYN) a következő sorrendben haladva alkalmazta a stratégiákat a keresés során: PAD, PPD, PFD, TFD, TPD, TAD, TFA, TPA, TAA. A keresési fában a szülő csúcsban használt stratégia után következőt alkalmazta. A PAA, PPA és PFA stratégiák teljesítettek a leggyengébben a korábbi teszteken, így ezeket nem alkalmazta a módszer.

Az 1. példa Ferrer-Nadal et al. 1. esettanulmányán^[6] alapul. 4-féle terméket kell legyártani, 4 berendezésen. Minden termék 3 meghatározott sorrendű lépésből áll, és minden lépést csak 1 berendezés tud végrehajtani. Utóbbi tulajdonság miatt a stratégiáknál nem számít, hogy a más berendezések által is végrehajtható taszkokat hogyan vesszük figyelembe (2. betű). Az eredeti mennyiségek helyett minden termékből 3 adag szerepelt a tesztekben, hogy nagyobb, könnyebben mérhető legyen a számítási igény.

[6] FERRER-NADAL, SERGIO (szerk.): Material transfer operations in batch scheduling. A critical modeling issue, in *Industrial & Engineering Chemistry Research*, 2008/20., vol. 47., 7721–7732.

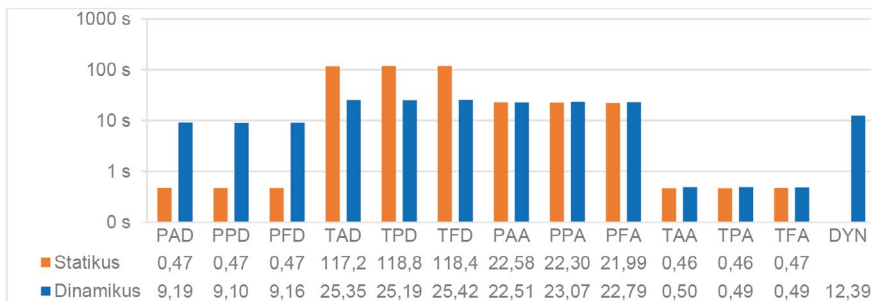
A 3. ábrán láthatók a teszteredmények. A futási idők a nagy eltérések miatt logaritmikus skálán vannak ábrázolva. A statikus sorrend volt a korábbi módszer, a dinamikus pedig a továbbfejlesztett változat. Megfigyelhető, hogy növekvő sorrend (**A) esetén nincs befolyása a dinamikus kiválasztásnak, hiszen ha növekvő terhelés szerint választunk berendezést, akkor a kiválasztott berendezésnek tovább csökken a fennmaradó terhelése, így nem változik a sorrend. A csökkenő sorrendet használó stratégiák esetén viszont javított a dinamikus módszer. A váltakozó stratégia hasonlóan jó teljesítményt ért el.

3. ábra: Megoldási idők az 1. példára



A 2. példa Voudouris és Grossmann tanulmányából^[7] származik. 4-féle 3-lépes terméket kell gyártani 5 dedikált berendezéssel. Itt is minden termékből 3 adag legyártása volt megadva a tesztekben. Az eredményeket a 4. ábra mutatja be. Erre a példára a P*D stratégia már statikus sorrend esetén is nagyon gyors megoldást eredményezett, amin a dinamikus módszer nem tudott javítani, a T*D stratégiánál viszont jelentős javulást ért el.

4. ábra: Megoldási idők a 2. példára

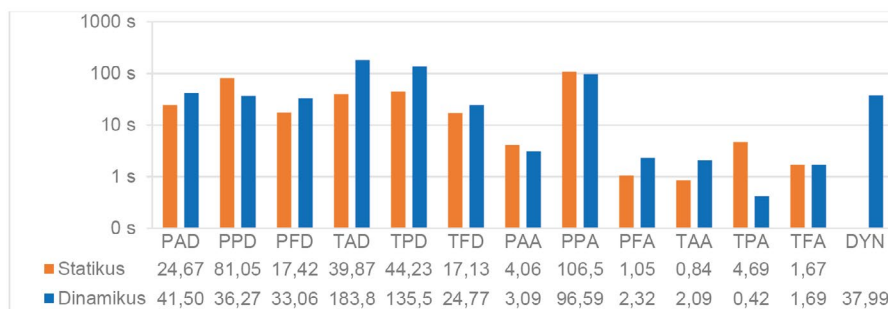


[7] VOUDOURIS, V. T. – GROSSMANN, I. E.: MILP model for scheduling and design of a special class of multipurpose batch plants, in *Computers and Chemical Engineering*, 1996/11., vol. 20., 1335–1360.

A 3. példa Papageorgaki és Reklaitis cikkéből^[8] származik, az ott található 3. példa. 5-féle termék található benne, különböző számú (2-5) lépéssel, melyeket 6 berendezés tud végrehajtani. Itt már vannak taszkok, melyekre 2 berendezés is képes. Minden termékből 2 adagot kellett gyártani. Fontos figyelembe venni, hogy a rendszerben vannak-e köztes anyag tárolók,^[9] amik lehetővé teszik, hogy két berendezés tartalma helyet cseréljen (cross-transfer). Ez a művelet folyadékoknál köztes tároló nélkül (NIS – No Intermediate Storage) nem kivitelezhető. Ahogy az első 2 példában, a 3. példában is NIS korlátozás szerepel. A 4. példa pedig a 3. példának az UIS változata (Unlimited Intermediate Storage).

A 3. példa teszteredményeit az 5. ábra mutatja be. A dinamikus módszer 4 esetben ért el javulást, de abszolút értékben csak a TPA stratégia esetén jelentős a gyorsulás mértéke. Több esetben viszont számottevően növelte a megoldási időt a statikus sorrendhez képest. Itt már növekvő sorrend esetén is lehet hatása a dinamikus stratégiáknak, mert ha az algoritmus úgy dönt, hogy egy berendezés már ne végezzen több taszkot, akkor a hátralévő taszkjai nagyobb súllyal jelennek meg más berendezéseknél. A DYN módszer nem közelítette meg a legjobb stratégiák teljesítményét, de a legrosszabbakhoz képest elfogadható megoldási időt ért el.

5. ábra: Megoldási idők a 3. példára

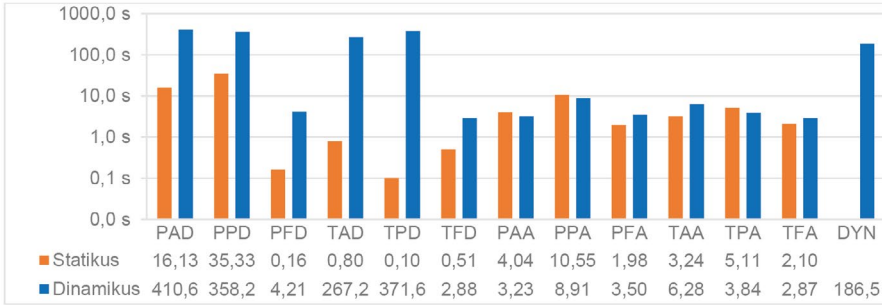


A 4. példára kapott eredmények a 6. ábrán láthatók. Az UIS példán jobban teljesítettek a statikus berendezés sorrendek. A dinamikusan újraszámított sorrendek több esetben jelentősen meghosszabbították a keresés idejét. A lassulás okának felderítése még további kutatásra szorul. A DYN stratégia is gyengén teljesített, hosszabb futási időt eredményezett, mint a legrosszabb statikus heurisztika.

[8] PAPAGEORGAKI, SAVOULA – REKLAITIS, GINTARAS V.: Optimal Design of Multipurpose Batch Plants. 2. A Decomposition Solution Strategy, in *Industrial & Engineering Chemistry Research*, 1990/10., vol. 29., 2062–2073.

[9] HEGYHÁTI MÁTÉ (ed.): Practical infeasibility of cross-transfer in batch plants with complex recipes: S-graph vs MILP methods, in *Chemical Engineering Science*, 2009/3., vol. 64., 605–610.

6. ábra: Megoldási idők a 4. példára



5. ÖSSZEFOGLALÁS

Kutatásom során az S-gráf megoldó algoritmusát futás közben kiszámított, heurisztikus döntési stratégiák alkalmazásával fejlesztettem tovább. A stratégiák hatékonyságát szakirodalmi példákon vizsgáltam.

A dinamikusan alkalmazott heurisztikák az esetek egy részében javítottak, más esetekben viszont rontottak a hatékonyságon. Ezért továbbra sincs egyértelműen legjobb módszer az ütemezendő berendezés kiválasztására. A kifejlesztett módszer viszont újabb lehetőségeket nyújt az algoritmus hatékonyságának növelésére. Segítségével a keresés különböző pontjain eltérő kiválasztási stratégiát lehet alkalmazni, amely figyelembe veszi a korábban meghozott ütemezési döntéseket.

A megfigyelések alapján a kiválasztási stratégiának nagy hatása van a teljesítményre, ezért érdemes további kutatást folytatni az ideális stratégia megválasztásának módszerét. Erre egy lehetőség a gépi tanulás használata, mellyel komplexebb összefüggések is felfedezhetők az ütemezés állapota és a stratégiák várható hatékonysága között. Ezen felül további kutatási lehetőség a részhalmoz kiválasztás lehetséges stratégiáinak vizsgálata, valamint a más szétválasztási eljárások esetén és feladatspecifikus paraméterek alapján történő kiválasztási stratégiák fejlesztése.

FELHASZNÁLT IRODALOM

- FERRER-NADAL, SERGIO (ed.): Material transfer operations in batch scheduling. A critical modeling issue, in *Industrial & Engineering Chemistry Research*, 2008/20., vol. 47., 7721–7732.
- GUPTA, K. OMPRAKASH – RAVINDRAN, A.: Branch and Bound Experiments in Convex Nonlinear Integer Programming, in *Management Science*, 1985/12., vol. 31., 1533–1546.
- HARJUNKOSKI, IIRO (ed.): Scope for industrial applications of production scheduling models and solution methods, in *Computers and Chemical Engineering*, 2014 (vol. 62), 161–193.

- HEGYHÁTI MÁTÉ (ed.): Practical infeasibility of cross-transfer in batch plants with complex recipes: S-graph vs MILP methods, in *Chemical Engineering Science*, 2009/3., vol. 64, 605–610.
- NEMES ZSOLT ÁDÁM: *S-gráf megoldó heurisztikus gyorsításai*, 2014, Mérnök informatikus BSc Szakdolgozat, Pannon Egyetem Műszaki Informatikai Kar Rendszer- és Számítástudományi Tanszék.
- PAPAGEORGAKI, SAVOULA – REKLAITIS, GINTARAS V.: Optimal Design of Multipurpose Batch Plants. 2. A Decomposition Solution Strategy, in *Industrial & Engineering Chemistry Research*, 1990/10., vol. 29., 2062–2073.
- SANMARTÍ, EDUARD – PUIGJANER, LUIS – HOLCZINGER TIBOR – FRIEDLER FERENC: Combinatorial framework for effective scheduling of multipurpose batch plants, in *AIChE Journal*, 2002/11., vol. 48., 2557–2570.
- VOUDOURIS, V. T. – GROSSMANN, I. E.: MILP model for scheduling and design of a special class of multipurpose batch plants, in *Computers and Chemical Engineering*, 1996/11., vol. 20., 1335–1360.