

Válasz Dr. Pataricza András Professzor Úr bírálatára

Az értekezés címe: **Planning of fault-tolerant solutions of operations research problems and multi-level heuristic scheduling**

Mindenekelőtt nagyon köszönöm, hogy Professzor Úr elvállalta disszertációm bírálatát, továbbá azt, hogy gondos, alapos és segítő szándékú észrevételeivel támogatta dolgozatom minőségének javítását.

Először Professzor Úr tézispontjaim vonatkozásában tett javaslataira reflektálok. Egyetértek azzal, hogy szerencsésebb lett volna az általánosabb, túlrészletezés nélküli megfogalmazás, ugyanakkor a benyújtott dolgozatban megfogalmazott tézispontok szövegén már nem megengedett, hogy változtassak (és azokat a nyilvános vitán is a benyújtott formában kell, hogy bemutassam). Mivel mind a bírálatokat, mind az azokra adott választ közzé kell tenni az egyetemi könyvtár weboldalán keresztül, bízom benne, hogy elfogadható megoldás az, ha jelen válaszomban kifejezem, hogy szerencsésebb lett volna az első tézispontom első altézisének az 1. ábrán látható, eredeti megfogalmazását a 2. ábrán láthatóak szerint, túlrészletezés nélkül közreadni. Ugyanígy, a 2. tézispontom 3. ábrán látható, eredeti megfogalmazásának a 4. ábrán látható, részletes parametrizálás nélküli változata is szerencsésebb.

1.1 I have compared the results of the heuristic algorithm with the result of a MILP solver on 1000 random inputs. In about half of the cases, the heuristic solver found the optimum. The relative error was at most 2% in 80% of the cases and at most 5% in 95% of the cases. The computation time of the heuristic solver was 2 minutes *altogether for the* 1000 inputs. That means 0.12 seconds for one instance on average. The exact solver's calculation usually took more time: there were cases when 10000 seconds were not enough for getting the optimal schedule for *one* instance. Based on these results, it can be stated that the heuristic algorithm is efficient.

1. ábra: Az első tézispontom első altézisének eredeti megfogalmazása

1.1 I have compared the results of the heuristic algorithm with the result of a MILP solver. Based on the results, the heuristic algorithm proved to be efficient.

2. ábra: Az első tézispontom első altézisének túlrészletezés nélküli, tömör változata

Thesis 2. I have developed and improved a genetic algorithm (GA) that can solve an extended FJSP scheduling problem with transportation devices (the considered problem is much more general than the problem of Thesis 1). I have investigated the algorithm's parameter settings and determined its efficient values for the analyzed problem (population size: 100, crossover rate: 0.3, mutation rate: 0.38). The improvement covers the creation of the initial population, the design of the genetic operators and the selection mechanism, and the development of the fitness function.

3. ábra: A második tézispontom eredeti megfogalmazása

Thesis 2. I have developed and improved a genetic algorithm (GA) that can solve an extended FJSP scheduling problem with transportation devices (the considered problem is much more general than the problem of Thesis 1). I have investigated the algorithm's parameter settings and determined its efficient values for the analyzed problem. The improvement covers the creation of the initial population, the design of the genetic operators and the selection mechanism, and the development of the fitness function.

4. ábra: A második tézispontom részletes parametrizálás nélküli változata

A dolgozat szaggatottságának (főszöveg-függelék szerkezet) oka a főszöveg 100 oldalas felső oldalkorlátja. Igyekeztem mind tárgyyszerű lenni, mind az érthetőséget növelni magyarázó példák és illusztrációk dolgozatomban történő elhelyezésével, de nem volt egyszerű az egyensúlyt megtalálni a megalapozott, érthető érvelés és a feszes szerkesztés között. Egyetértek Professzor Úr azon észrevételével, hogy még feszebb szerkesztéssel a szaggatottság csökkenthető és az érthetőség javítható lett volna.

Professzor Úr bírálatában feltett kérdésekre az alábbiakban adom meg a válaszokat:

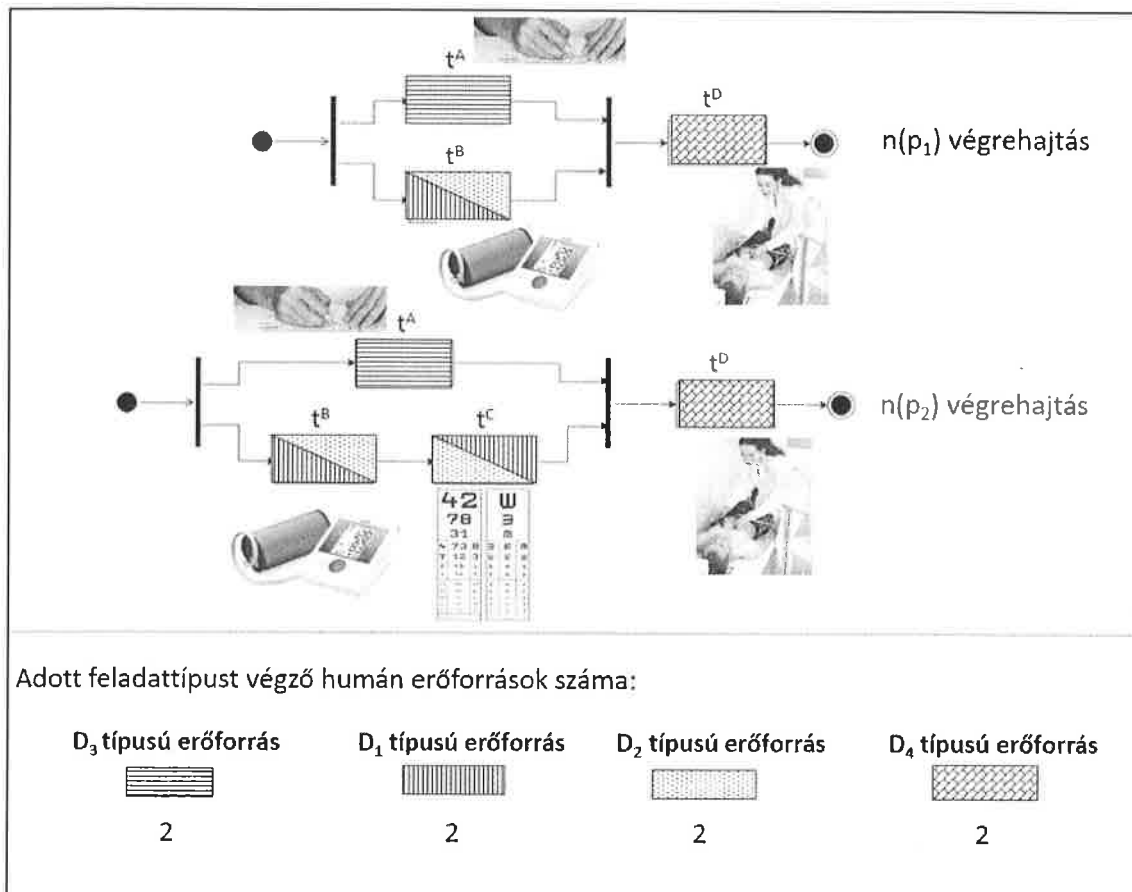
1. kérdés: A célkitűzés rögtön a legelején a termelési és a kapcsolódó kiszolgáló folyamatokra korlátozza az értekezés témáját. Ugyanakkor már a motivációról szóló 1.2 alfejezetben is megjelennek infokommunikációs példák. Értékelése szerint az eredmények mennyire specifikusak a termelési környezetre illetve az azok mögött rejlő matematikai apparátus általánosabb alkalmazhatósága milyen további lehetőséget kínál az alkalmazási terület esetleges bővítésére?

Válasz: A 2. és a 3. fejezet eredményei tetszőleges olyan ütemezési feladatokra alkalmazhatóak, amelyek a fejezetek végén összefoglalóan bemutatott (a 2.5.1. és a 3.7.1. alfejezetekben leírt) problémaosztályba tartoznak. Ez nem csupán termelési környezet esetén, de akár üzleti folyamatok, vagy infokommunikációs rendszerek ütemezése esetén is fennállhat.

Egy példa lehet erre egy tömeges üzemorvosi vizsgálat feladata (5. ábra), amely akár folyamatszerkezet szempontjából is analóg lehet a 2. fejezet példafeladatával. Megjegyzendő, hogy a példaként szerepeltetett feladat-struktúra helyett jóval szélesebb szerkezetű feladatokon alkalmazható mind a 2., mind a 3. fejezetben ismertetett megoldási módszer (ahogy ezt a 2.5.1. és a 3.7.1. alfejezetek is leírják). Fontos megemlíteni azt is, hogy a humán erőforrást alkalmazó feladatok esetén az erőforrások műveletvégzési idejének szórása jellemzően jóval nagyobb egy gépi erőforrásénál.

Az infokommunikációs hálózatok esetén is gyakran merülnek fel ütemezési feladatok (pl. feladatok párhuzamos processzorokra történő ütemezése, ahol a cél gyakran a legrövidebb teljes átfutási időt eredményező ütemezés megtalálása; szenzorhálózatok esetén az egyes szenzorok adatainak begyűjtése, ahol a cél az adatgyűjtés minél gyorsabb sebességének elérése mellett jellemzően az energiahatékonyság; kommunikációs hálózatok elemeinek – például antennák, műholdak, routerek – ütemezése az adatok minél kisebb teljes átfutási idejű továbbítása érdekében). Ha a disszertációban példaként bemutatott termelési folyamattal vizsgáljuk az analógiát, ezekben a feladatokban jellemzően egy-egy hálózati elem (node, router, antenna) jelenti az erőforrást, míg a tevékenység az adat átvitele és/vagy valamilyen transzformációja. Az infokommunikációs hálózatokra jellemző ütemezési feladatok folyamataiban – ellentétben a termelési folyamatokkal – a műveletvégzési idő jellemzően nem

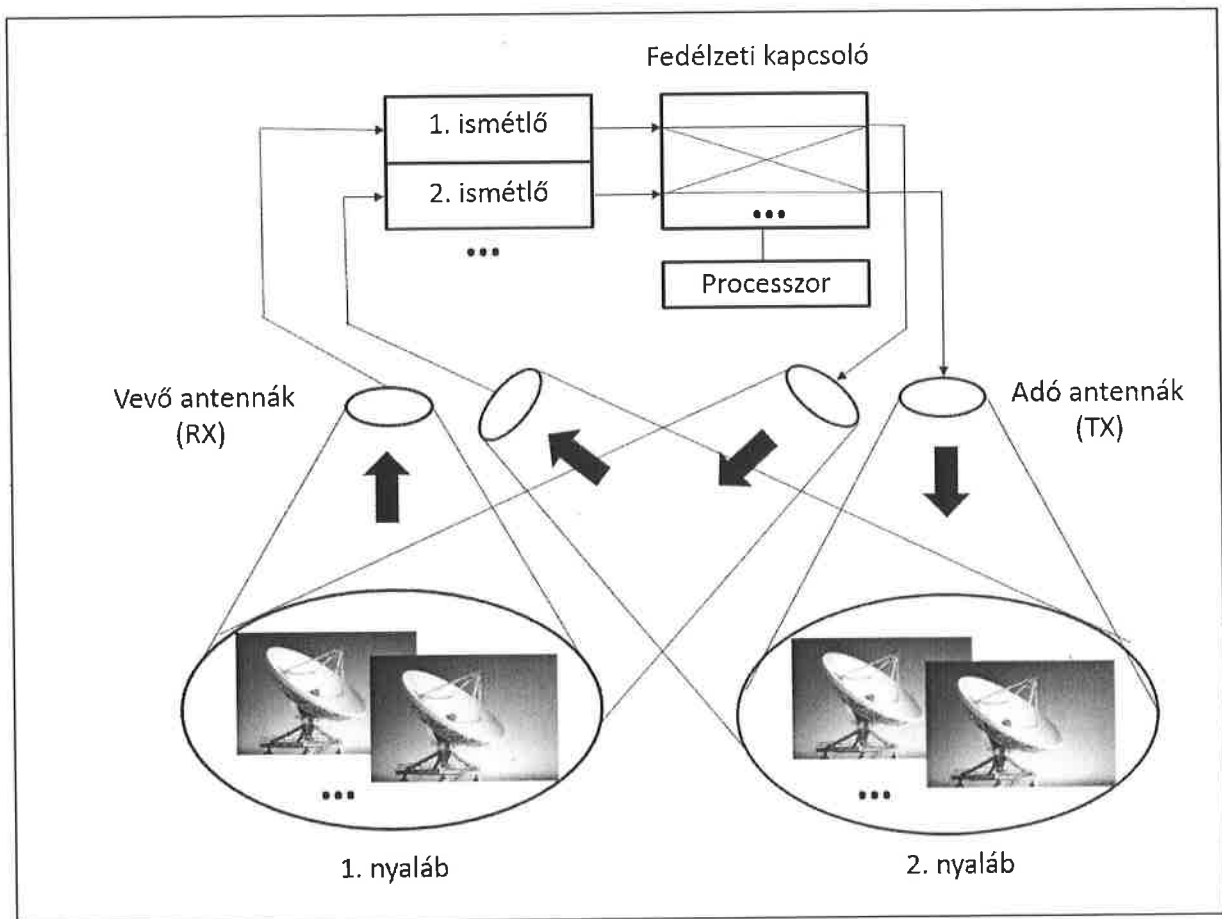
csak az erőforrástól és a feladat típusától függ, hanem annak tárgyától, azaz az adattól is (például az adat mennyiségétől, ami akár átvitelnként is más-más lehet, alkalmazkodva a felhasználói igényekhez). Viszont vannak olyan technológiai megoldások, ahol ez nem így van: például TDMA rendszerek esetén előre rögzített hosszúságú időszeletek jutnak minden kommunikáló fél számára, avagy bizonyos csomagkapcsolt rendszerek esetében a csomagok mérete előre rögzített (pl. ATM cellák). Ezen esetekben egy-egy időrésnyi adat, avagy egy-egy adatcsomag esetén – a disszertációmban bemutatott modellhez hasonlóan – a műveletvégzési idő az erőforrástól és a tevékenység típusától függ, a tevékenység tárgyától nem.



5. ábra: Egy példa a tömeges üzemorvosi vizsgálat feladatára

További fontos különbség az infokommunikációs rendszerek jellemző ütemezési feladatai és az általam vizsgált ütemezési feladatok között az, hogy az infokommunikációs rendszerek gyakran dinamikus környezetben, valós időben jelentkező, változó igények mellett működnek (szemben egy tömegtermelési feladat előre rögzített, nagy számosságú, azonos ütemezendő folyamataival). Emiatt az infokommunikációs rendszerek ütemezési igényeire leginkább az online ütemezési módszerek adnak megoldást. Mégis, előfordulhatnak olyan feladatcsoportok, amikor a disszertációmban tárgyalt offline megoldási módszerek jól alkalmazhatóak infokommunikációs rendszerek ütemezési feladataira is (többek között, mert az adatátviteli igények előre ismertek, nem kell őket valós időben megoldani – pl. bufferelés alkalmazása miatt –, továbbá minden feladat előre rögzített és azonos mennyiségű adaton kell, hogy történjen). Ilyen feladat lehet például földi (vagy más, űrbéli) állomások között átvinni kívánt bufferelt adatcsomagok műholdak transzpondereinek kommunikációs egységeire (vevőire/ismétlőire/adóira) történő ütemezése egy fedélzeti antennanyalábkapcsolást alkalmazó időosztásos többszörös hozzáférésű (Satellite-Switched

Time Division Multiple Access - SS/TDMA) rendszerben^{1,2} (6. ábra), ahol minden átviteli folyamat egy TDMA időrésnyi adat átvitelét jelenti egy adó és egy vevő állomás között.



6. ábra: Fedélzeti antennanyalábkapcsolást alkalmazó műholdas rendszer

Ennek az alkalmazási esetnek is számos ponton növelhető a komplexitása. Például itt is számolhatunk setup time-mal (pl. a fedélzeti kapcsoló általi új kapcsolás kialakításának ideje). Ezenkívül előfordulhat akár uplink nyalábok fedélzeti nyalábolása (pl. SkypeX) egy nagyobb sebességű downlink multiplex-re, amikor is az ütemezendő folyamat szerkezete már nem egyszerű szekvenciális struktúra (hanem az adási tevékenységet több közvetlenül megelőző más tevékenység is szerepel az ütemezendő folyamatban, akár csak a doktori dolgozatomban ismertetett ütemezési feladatmodell összeszerelési műveletének esetében).

2. kérdés: A gyorsító heurisztikák alkalmazása esetében a leírtak szerint bizonyos egyszerűsítések nem mindig alkalmazhatóak. Van egzakt kritérium arra, hogy ezek az esetek milyen feladatoknál következnek be?

Válasz: Az alkalmazott egyszerűsítések (5 darab) vonatkozó válaszmát a dolgozatban szereplő sorszámuk alapján, sorban taglalom:

(S1) Gépenként legfeljebb egyszer fordul elő átállítási idő (setup time)

¹ Christian Prins: An overview of Scheduling Problems Arising in Satellite Communications (1994), *The Journal of the Operational Research Society*, Vol. 45., No. 6., pp. 611-623.

² S.D. Ilčev: Implementation of Multiple Access Techniques Applicable for Maritime Satellite Communications (2013), *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, Vol. 7., No. 4., pp. 529-540.

Ez az egyszerűsítés technikailag bármely olyan feladat esetén alkalmazható, ahol egy gép legfeljebb kétféle feladattípus elvégzésére képes, hiszen a gép által elvégezhető kétféle tevékenységtípusba (t^B és t^C típus) tartozó tevékenységekből álló tevékenység-sorozatnak minden esetben létezik olyan permutációja, amelyben egy t^B típusú tevékenységet sem előz meg t^C típusú tevékenység, illetve olyan is, amelyben egyetlen t^C típusú tevékenységet sem előz meg t^B típusú tevékenység. Fontos megemlíteni, hogy a dolgozatomban bemutatott heurisztikus algoritmus alkalmazhatósági feladatosztályára kikötöttem, hogy egy gép legfeljebb kétféle feladattípus elvégzésére képes (2.5.1. fejezet) – emiatt a gépenként legfeljebb egyszer alkalmazott átállítási idő minden ilyen feladat esetén alkalmazható. Ugyanakkor azt érdemes megjegyezni, hogy ezzel az alkalmazhatósággal nem mindig érdemes élni. Bizonyos esetekben ez az egyszerűsítés nem ront, vagy nem sokat ront a megoldáson (tesztjeim során 1000-ból kb. 500 esetben megtaláltam az optimumot és a relatív hiba az esetek 80%-ában legfeljebb 2% volt), de az optimumhoz képesti elérhető távolság függ a termékek darabszámától, a műveletvégzési és az átállítási időktől. Az előforduló rosszabb eredmények oka az, hogy gépenként legfeljebb egy átállítást alkalmazva az egyik alkatrész felszaporodik, miközben az összeszereléshez szükséges más tevékenység még nem kerül elvégzésre. Azt, hogy pontosan hogyan függ a megoldás minősége a feladat paramétereitől (termékek darabszáma, műveletvégzési és átállítási idők), nem tudtuk egzakt módon meghatározni a probléma összetettsége miatt.

(S2) Minden gépen annak bármely t^B -típusú művelete megelőzi a gép bármely t^C -típusú műveletét.

Az (S1) pontban adott válasz alapján ez az egyszerűsítés is bármely feladatra alkalmazható.

(S3) Vagy a D_1 -típusú gépek nem végeznek t^C -típusú műveletet, vagy a D_2 -típusú gép nem végez t^B -típusú műveletet. (A feltevés második része általánosabban – ha a szobában mindkét géptípusból több van – a következő: „vagy a D_2 -típusú gépek nem végeznek t^B -típusú műveletet”. Ugyanakkor a tárgyalt modellben csak egyetlen D_2 -típusú gép van (r_4).)

Mivel mindkét típusú gép képes mindkét művelettípus elvégzésére, így ez a feltevés is mindig alkalmazható. A feltevés (illetve annak két opciója közül az egyik) alkalmazása ugyanakkor hatással lehet az optimum maximális megközelíthetőségének mértékére.

(S4) A (vagy általánosan: bármely) D_2 -típusú gép a feladatait megállás nélkül végezheti (azzal a kitételrel, hogy a (vagy általánosan: bármely) D_2 -típusú gép $\text{dur}(t^B, D_1)$ időpontban kezdhet el dolgozni, ahol $\text{dur}(t^B, D_1)$ egy t^B -típusú tevékenységnek egy D_1 -típusú gépen való elvégzésének műveleti ideje). Továbbá bármely olyan D_1 -típusú gép, amelyik t^C -típusú műveletet is végez, folyamatosan végezheti a t^C -típusú műveleteit az utolsó t^B -típusú művelete utáni átállítási idő letelte után.

Ez az egyszerűsítő feltevés azt jelenti, hogy a t^C -típusú műveleteket folyamatosan tudják végezni a gépek. Ez a feltevés csak abban az esetben érvényes, ha minden t^C -típusú művelet megkezdése esetén már elvégzésre került egy neki megfeleltethető t^B -típusú művelet. Ennek megvalósíthatósága viszont abból adódik, hogy a heurisztika a lehetséges esetekből azt az alesetet választja, amelyikben ez teljesül (azaz e feltétel mentén határozza meg azt, hogy mely gépek legyenek azok, amelyek kétféle műveletet is végezhetnek).

(S5) A dolgozat (2.1)-(2.3) egyenletrendszerének megoldását csak nemnegatív számok alkotják, azaz $x, y, v, w \geq 0$.

Ez az egyszerűsítés – kiindulva az alább látható, a dolgozatból idézett (2.1)-(2.3) egyenletekből – néhány jól értelmezhető egzakt kritériumot definiál a feladat vonatkozásában.

$$\alpha * \gamma + \beta * v = n(p_1) + n(p_2) \quad (2.1)$$

$$\beta * w + \gamma * x = n(p_2) \quad (2.2)$$

$$\begin{aligned} \gamma * dur(t^B, D_1) &= v * dur(t^B, D_1) + w * dur(t^C, D_1) + setup(D_1, t^B, t^C) = \\ &= x * dur(t^C, D_2) + dur(t^B, D_1) \end{aligned} \quad (2.3)$$

Röviden a származtatható kritériumok:

- Legalább egy olyan D_2 -típusú gép kell, hogy legyen a feladatban, amelyik csak t^C -típusú tevékenységek elvégzésére képes.
- Egy t^B -típusú művelet hossza egy D_1 -típusú gépen legalább akkora kell, hogy legyen, mint ugyanezen típusú gép átállítási ideje t^B és t^C -típusú tevékenységek között.
- Az ütemezendő t^B -típusú tevékenységek számának legalább annyinak kell lennie, mint a t^C -típusú tevékenységeket is elvégezni képes D_1 -típusú gépeké.

Azt, hogy ezen egzakt kritériumokat hogyan származtattam, az alábbiakban részletezem.

A (2.3) egyenlet bal és jobb oldalát tekintve és alkalmazva azt, hogy $x \geq 0$, valamint, hogy műveletvégzési idő nem lehet negatív szám, azt kapjuk, hogy

$$\gamma * dur(t^B, D_1) \geq dur(t^B, D_1).$$

Ebből az következik, hogy $\gamma \geq 1$, azaz legalább egy olyan D_2 -típusú gép kell, hogy legyen a feladatban, amelyik csak t^C -típusú tevékenységek elvégzésére képes.

Ha a (2.3) egyenlet bal és középső részét vesszük figyelembe, valamint az egyszerűsítő feltétel vonatkozásában azt, hogy $v \geq 0$ és $w \geq 0$, tudván, hogy műveletvégzési idő nem lehet negatív szám, az adódik, hogy

$$\gamma * dur(t^B, D_1) \geq setup(D_1, t^B, t^C).$$

Ez azt jelenti, hogy a csak t^C -típusú tevékenységek elvégzésére képes D_2 -típusú gépek számának (γ), és a t^B -típusú művelet D_1 -típusú gép általi elvégzésének idejének szorzata nem kisebb, mint a D_1 -típusú gépek átállítási ideje t^B -ről t^C -típusú tevékenység elvégzésére. Mivel az előbb láttuk, hogy az egyszerűsítő feltétel alapján $\gamma \geq 1$ kell, hogy legyen, ezért az is elmondható, hogy **egy t^B -típusú művelet hossza egy D_1 -típusú gépen legalább akkora kell, hogy legyen, mint ugyanezen típusú gép átállítási ideje t^B és t^C -típusú tevékenységek között.**

Ha a (2.2) egyenletből kifejezzük w -t, és behelyettesítjük a (2.3) egyenletbe, majd tekintjük az egyenlet középső és jobb oldali részét, azt kapjuk, hogy

$$\begin{aligned} v * dur(t^B, D_1) + \frac{n(p_2) - \gamma * x}{\beta} * dur(t^C, D_1) + setup(D_1, t^B, t^C) &= \\ &= x * dur(t^C, D_2) + dur(t^B, D_1). \end{aligned}$$

Az egyenletet átrendezve a következő eredményre jutunk:

$$(v-1) * dur(t^B, D_1) + setup(D_1, t^B, t^C) + \frac{n(p_2) * dur(t^C, D_1)}{\beta} \\ = x * \left(dur(t^C, D_2) + \frac{\gamma * dur(t^C, D_1)}{\beta} \right).$$

Mivel sem műveletvégzési idő, sem átállítási idő nem lehet negatív, továbbá a feltevés alapján $x \geq 0$, valamint semmilyen géptípus halmaza nem lehet negatív számosságú (azaz $\beta \geq 0$ és $\gamma \geq 0$) / β azon D_1 -típusú gépek száma, amelyek t^C -típusú művelet elvégzésére is képesek, γ pedig a csak t^C -típusú műveletek elvégzésére képes D_2 -típusú gépek száma/, ezért

$$v - 1 \geq 0.$$

Mivel a dolgozatban v az olyan D_1 -típusú gépeken, amelyek t^C -típusú tevékenységeket is képesek elvégezni, a t^B -típusú tevékenységek számát jelentette, így kijelenthető, hogy minden ilyen gépen legalább 1 t^B -típusú tevékenységnek lennie kell.

Bár ez az eredmény a feladat megoldására nézve megszorítás, mégis, magára a feladatra nézve is korlátot jelent, mégpedig azt, hogy **az ütemezendő t^B -típusú tevékenységek számának legalább annyinak kell lennie, mint a t^C -típusú tevékenységeket is elvégezni képes D_1 -típusú gépeké** (ez utóbbit β jelöli). Mivel a folyamatok mindegyike (mind az első, mind a második típusba tartozóak) pontosan egy t^B -típusú tevékenységet tartalmaznak, ezért a feladatra vonatkozó egzakt kritériumot így fogalmazhatjuk meg:

$$n(p_1) + n(p_2) \geq \beta.$$

Ez a kritérium adja magát a (2.1) egyenletből is rögtön, alkalmazva, hogy $\gamma \geq 0$ és az előbb megkapván, hogy $v \geq 1$.

3. kérdés: A hatékonyság értékelésénél mindhárom problémánál véletlen feladathalmazokat alkalmaz. Mi biztosítja azt, hogy ezek az adott feladatkörre általános reprezentatívak?

Válasz: Az ütemezési feladatok esetén nem törekedtem teljes mértékű általánosságra. Ezzel szemben feladatosztályokat definiáltam, és azokon belül vizsgáltam az algoritmusok hatékonyságát. Professzor Úr kérdését az egyes fejezetek vonatkozásában külön-külön válaszolom meg a következőkben.

A 2. fejezetben bemutatott modell új, emiatt a munka során kitalált feladatosztállyal dolgoztam, korábbi inputok ilyen jellegű probléma vonatkozásában nincsenek jelen a szakirodalomban. Ezen a feladatosztályon – ahogy az a fejezetben is olvasható – sok feladatot megoldottam, és a kapott eredményeket sokféle szempont alapján hasonítottam össze. Ugyanakkor várhatóan lényegesen más eredményt kapnánk, ha például a feladat gépcsoportjainak aránya más lenne (pl. 4 darab D_1 -típusú, 5 darab D_2 -típusú, 3 darab D_3 -típusú és 10 darab D_4 -típusú). A kutatás célja nem egy általánosan érvényes algoritmus létrehozása volt, hanem egy példa mutatása arra, amikor a bemutatott „trükkök” hatékonyan működnek (dekompozíció, mohó algoritmusok alkalmazása a részproblémák egy részére, másik részére többfázisú heurisztika készítése). A vizsgált input osztályon belül viszont nagyszámú (1000) és változatos feladatot vizsgáltam meg. Erre a feladatosztályra nézve általános következtetések levonása érdekében a feladatok egyes paramétereinek értékét egymástól függetlenül, adott intervallumból történő, egyenletes eloszlást követő véletlenszerű választással határoztam meg.

A 3. fejezetben tárgyalt modell a 2. fejezet modelljének általánosítása. Mivel már a 2. fejezet modellje is új volt, így ez is új abban az értelemben, hogy a szóban forgó feladatcsoportot nem vizsgálták, a szakirodalomban nem volt ilyen jellegű modell tárgyalva eddig. E fejezet feladatosztályát is alaposan megvizsgáltam, emellett a fejezetben bemutatott algoritmust a szakirodalom jelentős számú és különböző jellegű benchmark feladatán is teszteltem. A vizsgált benchmark feladatok nem csak a műveletvégzési időkben, az ütemezendő folyamatok számában és a folyamatonkénti tevékenységszámban tértek el, de az alkalmazott gépek számában, és abban is, hogy egy tevékenységet átlagosan hány gép képes elvégezni.

A 4. fejezetben a kidolgozott algoritmus jellege biztosítja azt, hogy az abban reprezentált alapelv általánosan alkalmazható tetszőleges kapacitásos, egy depóval rendelkező, időablak nélküli kiszállítási probléma megoldására. A bemutatott algoritmus meghatározza az átlagosan legjobb „útirány kompozíciót”, ezt az összes lehetőség kipróbálásával – a nyers erő módszerével – biztosítja. Emiatt az elmondható, hogy a legjobb „útirány kompozíció” kiválasztása általánosan reprezentatív kapacitásos, egy depóval rendelkező, időablak nélküli kiszállítási problémák megoldására, ugyanakkor az elért nyereség mértéke nem általánosítható. Itt szintén a célom egy újszerű megközelítés megadása volt, amit számos benchmark feladaton vizsgáltam (véletlenszerűen generált saját feladaton, illetve a szakirodalomban széleskörűen alkalmazott Solomon teszteken), ami megfelelő arra, hogy megadhassam azt, hogy adott szintű eredmények elérésére alkalmas a módszerem, de annak pontos határainak megadhatóságát nem biztosítja.

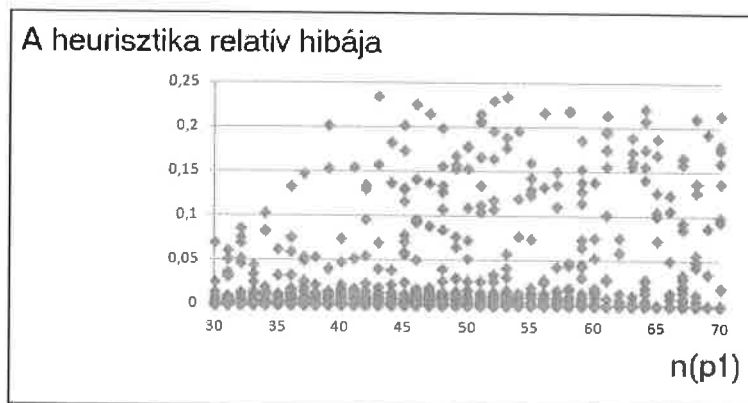
4. kérdés: Van-e lehetőség arra, hogy egy új feladat felmerülése esetén algoritmikusan döntsünk legalább jó valószínűség erejéig a szóban lévő alternatívák közötti választásról!

Válasz: Annak eldöntéséhez, hogy egy új feladat esetén a heurisztikát vagy az egzakt megoldót érdemes-e használni, azt vizsgáltam meg, hogy függ-e a disszertációban vizsgált két folyamat típus végrehajtási számosságának arányától ($n(p_1)/n(p_2)$) az, hogy a heurisztika mennyire optimum-közelit eredményt tud adni. Ha meghatározhatóak lennének olyan arány értékek, amelyekre szignifikánsan jobb (az optimumhoz közelebbi) eredményt kapnánk a heurisztikával, mint a többi esetben, akkor kijelenthető lenne, hogy egy új feladat esetén algoritmikusan eldönthető, hogy várhatóan jó eredményt szolgáltat rá a gyorsan végrehajtható heurisztikus algoritmus (így érdemesebb a heurisztikával megoldani, mint az egzakt megoldóval, hiszen nagy valószínűséggel nem csak gyors, de jó eredményt is kapunk).

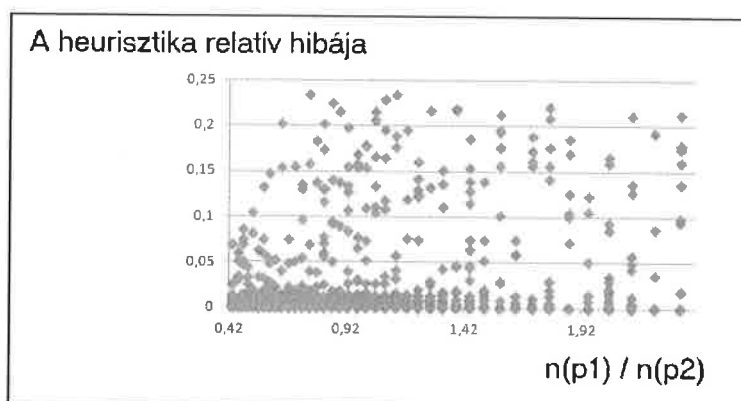
Ezt a vizsgálatot is a disszertációban bemutatott 1000 véletlenszerű inputon végeztem, ahol minden input esetén az összes ütemezendő folyamat száma ($n(p_1)+n(p_2)$) 100 volt úgy, hogy mindkét típusú folyamat végrehajtásának száma 30 és 70 közötti intervallumban helyezkedett el. Ennek megfelelően a 7. ábra a heurisztikus algoritmus relatív hibáját mutatja, miközben az első folyamat típus végrehajtási száma 30-ról 70-re nő, míg a 8. ábra függőleges tengelye szintén a heurisztika relatív hibáját ábrázolja, miközben a vízszintes tengelyen az első és a második folyamat típus végrehajtási számainak aránya változik 3/7 és 7/3 között.

Az ábrákról egyrészt látható, hogy a vízszintes tengely bármely tartományán vannak olyan inputok, amelyekre a heurisztika optimális megoldást ad, de ugyanígy vannak gyengébb megoldások is. Ugyanakkor bizonyos jellegzetességek megfigyelhetők, főleg, ha a 7. és 8. ábra adatait úgy szemléltetem, hogy a vízszintes tengely tartományát azonos méretű blokkokra bontom, és a blokkba illeszkedő inputokra kapott relatív hibáját a heurisztikának

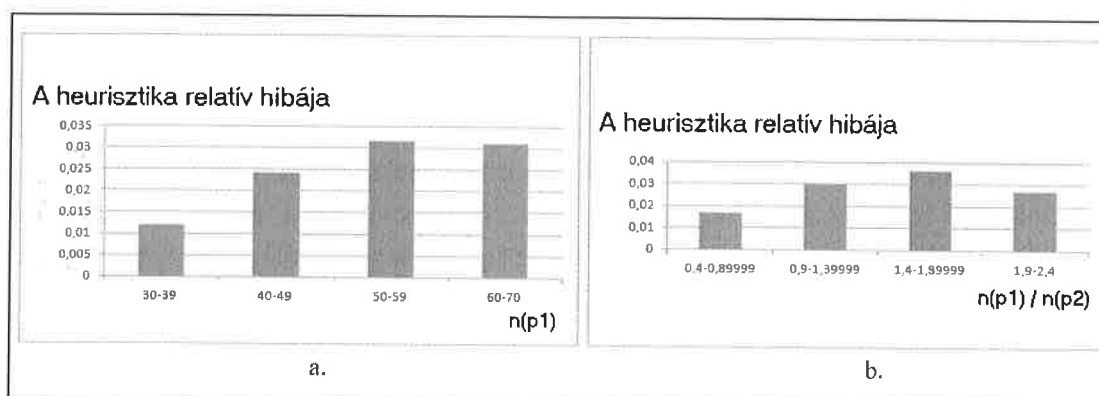
átlagolom. Ezt szemlélteti a 9. ábra. Megfigyelhető, hogy a heurisztika relatív hibája akkor a legkisebb, amikor az első típusú folyamat végrehajtási száma alacsony (legfeljebb 39; így a második típusú folyamat végrehajtási száma legalább 61). Ahogy a 9. ábra bal oldali oszlopa a legalacsonyabb, úgy a 7. és 8. ábra bal felső része is üres.



7. ábra: A heurisztikus algoritmus relatív hibája az első folyamattípus végrehajtási számának függvényében



8. ábra: A heurisztikus algoritmus relatív hibája a két folyamattípus végrehajtási számai arányának függvényében



9. ábra: A heurisztikus algoritmus relatív hibája az (a) első folyamattípus végrehajtási számának függvényében, blokkonként átlagolva (b) a két folyamattípus végrehajtási számai arányának függvényében, blokkonként átlagolva

Tehát, ha az inputban szereplő két folyamattípus közül az első kisebb számban kell végrehajtani, akkor mindenképpen érdemes a heurisztikus algoritmust használni.

Emellett, ahogy a dolgozatban olvasható, ha egzakt megoldót alkalmazunk, akkor azt érdemes a heurisztikus algoritmus által adott eredménnyel segíteni, leginkább a durva alsó korlát és a heurisztika által nyújtott értékek közti logaritmikus keresés módszerét (a dolgozatban Method5) alkalmazva.

5. kérdés: Az értekezés legvégén vázoltakon túl, mennyire általánosíthatóak az eredmények, ha például az egyes megszorító feltételeket elkezdjük relaxálni?

Válasz: Az alkalmazott megszorító feltételek relaxálásának a dolgozatban ismertetett eredményekre gyakorolt hatását az alábbiakban fejezetenként mutatom be.

A 2. fejezet esetén alkalmazott megszorító feltételek a feladat részfeladatokra való szeparálhatóságát, továbbá az erőforrások korlátozását legfeljebb kétféle feladattípus elvégzésére foglalták magukba. A kiindulási feladat részfeladatokra bonthatósága nem jelent lényeges korlátot abban a tekintetben, hogy ha nem bontható fel a feladat részfeladatokra, akkor a megoldás során a teljes feladatot egyetlen „szobaként” tekintjük, és erre dolgozzuk ki a megoldáshoz szükséges módszert. A hatása ennek a scenáriónak annyi lesz, hogy a kezdeti feladat megoldásának komplexitását nem fogjuk tudni csökkenteni részfeladatokra bontással. Abban az esetben viszont, ha megengedünk egy erőforrás számára kettőnél többféle tevékenységtípus elvégzését, akkor a fejezetben ismertetett megoldó módszert jelentősen át kellene alakítani, hiszen számos alkalmazott egyszerűsítés azt a tényt használta ki, hogy egy erőforrás legfeljebb kétféle feladattípus elvégzésére képes.

A 3. fejezetben bemutatott genetikus algoritmus lényegesen általánosabb ütemezési feladatok megoldására képes, mint a 2. fejezetben ismertetett heurisztika. Az itt alkalmazott megszorító feltétel a szállítóeszközök 1 termékenyi kapacitása volt. Bár az algoritmus könnyedén módosítható oly módon, hogy nagyobb kapacitású szállítóeszközök kezelésére is képes legyen, ugyanakkor nincs beépítve olyan logika, amely a nagyobb kapacitások mentén való optimalizálást végezné, azaz várhatóan 1-nél nagyobb kapacitású szállítóeszközök megengedése esetén a kapott eredmény gyakran messzebb lenne az optimális megoldástól, mint a dolgozatban alkalmazott megszorító feltétellel rendelkező feladatok esetén.

A 4. fejezetben számos megszorító feltételt alkalmaztam. Ha szakítunk a járművek azonos és konstans sebességének feltételezésével, akkor annak hatására azok a pontok, amelyeken az egyes járművek tartózkodnak az egyikük meghibásodása esetén, hűbben fogják tükrözni a valós lokációkat. Ez hatással lehet a segítő jármű kiválasztására és annak útvonalának meghatározására. Ugyanilyen hatással bírna az, ha az ügyfelek kiszolgálási idejével, illetve a meghibásodott járműnél az átpakolással töltött idővel is számolnánk, továbbá ha megengednénk, hogy a járművek ne feltétlenül egyszerre induljanak a depóból.

Amennyiben azzal a feltevessel szakítunk, hogy minden jármű azonos kapacitású, a meghibásodott járműnek az algoritmusban leírt módon történő meglátogatásának többszöri megismétlését teheti szükségessé, ezzel kiterjesztve az algoritmussal megoldható feladatok körét, de mindez az eddigi módszerrel is megoldható feladatok eredményeire nincs hatással. Ha egynél több jármű segítségét is megengedjük, akkor – bár jóval komplexebb számítások válnak szükségessé, de – az eredmény várhatóan rövidebb teljes kiszállítást fog eredményezni azzal, hogy bár több jármű eredeti útvonalának hossza nő meg, de összesen nem akkora mértékben, mintha a teljes segítséget csak egyetlen jármű végezné.

Amennyiben a jármű-meghibásodás feltételezett lokációit nem csak az ügyfelek tartózkodási helyéhez kötném, hanem bárhol számolnék vele útjuk során, akkor – bár több számításra lenne szükség, de – egy-egy jármű meghibásodásának lehetőségeit figyelembe véve több pozíció útváltoztatási költségét (route change cost) átlagolnám, így pontosabban tudnám meghatározni a legjobb „útirány kompozíciót”.

A depók számára (1) vonatkozó megszorító feltétel relaxálásával az ügyfelek depókhoz rendelésével kibővül a feladat. Ez az alkalmazhatóságot bővíti, ugyanakkor az egy depóval rendelkező feladatok eredményére nincs hatással.

A bemutatott algoritmus tekintetében az időablak-nélküliség, továbbá a pontpárok közötti szimmetrikus távolságok megszorító feltétele nem relaxálható, hiszen az szembe menne az algoritmus alapfeltevésével, mely szerint a körutak bármely irányban bejárhatóak nélkül, hogy a bejárési irány megválasztása az adott körút bejárési költségére hatással lenne.

6. kérdés: Volt-e a kidolgozott algoritmusoknak gyakorlati alkalmazása, és ha igen az milyen eredményeket mutatott?

Válasz: A disszertációm írásának kezdete előtt kutatócsoportunkon belül foglalkoztam ütemezéssel két ipari partner gyártási- illetve tesztfolyamatai vonatkozásában. Ugyanakkor ezek a feladatok szekvenciális struktúrájú folyamatok ütemezését foglalták magába, és a feladatban megjelenő korlátozó feltételek vonatkozásában is gazdag irodalom állt rendelkezésre. A disszertációmban bemutatott komplexitású ütemezési feladatokkal valós környezetben még nem foglalkoztam, így a kidolgozott algoritmusoknak eddig nem volt gyakorlati alkalmazása. A kiszállítási feladatok vonatkozásában adott megoldásomat sem használtam eddig konkrét gyakorlati alkalmazásban. Megjegyzendő ugyanakkor, hogy a közelmúltban csatlakoztam egy, többek között a Műszaki Informatikai Karon végzett kutatási+fejlesztési projekt csapatához, akikkel egészségügyi feladatok ütemezésének megoldása a feladatunk. Remélem, hogy kutatómunkám eredményei e projekt keretei között eredményesen felhasználhatóak lesznek.

Végül ismételten köszönöm Professzor Úrnak a dolgozatom gondos, konstruktív és pozitív bírálatát, és kérem a válaszom elfogadását.

Veszprém, 2021. június 7.



Dulai Tibor

a PhD dolgozat szerzője

Informatikai Tudományok Doktori Iskola
Pannon Egyetem, Műszaki Informatikai Kar