sciendo

Műszaki Tudományos Közlemények vol. 13. (2020) 137–141.
DOI: Magyar: https://doi.org/10.33895/mtk-2020.13.25
Angol: https://doi.org/10.33894/mtk-2020.13.25

EME MTSZ MTK

KITERJESZTETT PROJEKTÜTEMEZÉSI FELADATOK MEGOLDÁSA

SOLVING EXTENDED PROJECT SCHEDULING PROBLEMS

Mihály Krisztián,[1] Kulcsár Gyula[2]

Miskolci Egyetem Alkalmazott Informatikai Intézeti Tanszék, Miskolc, Magyarország

[1] *altmihaly@uni-miskolc.hu*

[2] *iitkgy@uni-miskolc.hu*

Abstract

Project based planning and execution is used in various phases of a product lifecycle, starting from the conceptual idea and design through to the manufacturing and maintenance. It is common, that an enterprise or an organization executes more than one project in parallel. These projects may share the same resources and may have differing goals. Based on experience the order of task execution is a key factor, having a major impact on the key performance indicators. Our paper presents an extended model and a scheduling method for resource constrained project scheduling problems.

Keywords*: project scheduling, generation scheme, multi-objective, multi-project, extended RCPSP.*

Összefoglalás

A projektalapú tervezés és végrehajtás egy termék életciklusának különböző fázisaiban jelenthet meg, kezdve a koncepcionális elképzeléstől és tervezéstől egészen a gyártásig és karbantartásig. Általánosságban elmondható, hogy egy vállalat vagy szervezeti egység egynél több projektet hajt végre párhuzamosan. Ezek a projektek közös erőforrásokon osztozhatnak, és különböző célokat szeretnének megvalósítani. Gyakorlati tapasztalatok alapján az elvégzendő feladatok végrehajtási sorrendjének kulcsszerepe van,és alapvető hatást gyakorolnak a teljesítménymutatókra. Cikkünkben az erőforráskorlátosprojekt-ütemezési probléma egy kiterjesztett modelljét és ütemezési módszerét mutatjuk be.

Kulcsszavak*: projektütemezés, generálási séma, többcélúság, párhuzamos projekt, kiterjesztett RCPSP.*

1. Projektütemezés

A projektalapú feladat-végrehajtás napjainkban nagyon sok üzleti és ipari területen megtalálható. Projektelven lehet megszervezni például egy egyedi termékfejlesztést, egy sportesemény megszervezését vagy egy projektalapú gyártást. A projektütemezés témaköre nagy múltra tekint vissza, és a felhalmozott ismeretanyag ellenére egy továbbra is aktívan kutatott terület, mivel– a speciális eseteket most nem tekintve – az NP nehéz feladatok körébe soroljuk **[1]**.

Kutatásunkban egy ismert erőforráskorlátosprojekt-ütemezési feladattípust új szempontokkal egészítettünk ki, melyre keresünk hatékony leíró modellt és projektütemező eljárásokat. Kidolgoztunk egy új megoldási koncepciót, melyben reaktív szabályalapú módszereket, keresési technikát és szimulációs módszereket kombináltan használunk

2. Erőforráskorlátosprojekt-ütemezési probléma

Az erőforráskorlátosprojekt-ütemezési (RCPSP) feladatot az alábbiak szerint adhatjuk meg **[2]**:

Adott az elvégzendő feladatok egy előre ismert halmaza T = {1,2,3,…,n}.

Adottak előre ismert erőforrástípusok K = {1,2,3,…,m}. Az erőforrástípusok kapacitása korlátozott, azaz nem végtelen mennyiségben állnak rendelkezésre az erőforrástípusokhoz tartozó erőforrások. Emiatt előfordulhat, hogy adott időpillanatban a feladatok párhuzamos végrehajtásához nem áll rendelkezésre megfelelő mennyiségű erőforrás. Az erőforrások kapacitása előre ismert, melyet a kezdeti ismert állapotból származtatva bármely ütemtervtől függő időpillanatban determinisztikusan meg tudunk határozni. Ezt a kapacitást R_k, k ∈ K-módon jelöljük. Az erőforrástípusokat megújulónak tekintjük, azaz bármely feladat végrehajtása után a feladat végrehajtásához szükséges kapacitás újra rendelkezésre áll (pl. személyek, gépek, eszközök stb.).

A probléma definiálásakor két korlátozó típusú feltételt adhatunk meg. Minden feladat esetén ismert az elkezdéshez szükséges feladatok halmaza, melyet P_i-vel jelölünk. Egy i ∈ T feladat akkor végezhető el, ha minden j ∈ P_i feladat végrehajtásra került. Egy feladat végrehajtásához szükség lehet egy vagy több erőforrástípusból egyedileg előre meghatározott kapacitásra. A feladat akkor hajtható végre, ha a végrehajtáshoz szükséges kapacitásszükséglet minden erőforrástípusból egyszerre kielégíthető.

Az ütemezési feladat megoldása egy olyan feladat-végrehajtási-sorrendmegtervezését jelenti, melyet a végrehajtás során követve, a kapacitáskorlátokat nem sértjük meg, és az előzési relációk teljesülnek. Ezt végrehajtható projektütemezésnek (röviden ütemezésnek) nevezzük. Amennyiben több végrehajtható ütemezés is létezik, akkor az ütemezési feladat célfüggvény megadásával specializálható. Az ütemezési feladat megoldása során arra törekszünk, hogy a célfüggvény figyelembevételével a lehető legjobb végrehajtható ütemezést állítsuk elő.

3. Kiterjesztett modell

A második fejezetben bemutatott feladatot az alábbi szempontok figyelembevételével terjesztettük ki.

3.1. Több célfüggvény együttes alkalmazása

Két végrehajtható ütemezés közül azt tekintjük egy minimalizálandó célfüggvény szempontjából jobbnak, amely megoldás esetében a kapott függvény-érték kisebb. Maximalizálandó célfüggvény esetében a nagyobb célfüggvényérték jelenti a jobb megoldást. Ismert és gyakran használt minimalizálandó célfüggvény például a legkésőbbi feladat befejezési ideje (C_{max}), a késéssel befejezett munkák száma és a legnagyobb késés (L_{max}). Maximalizálandó célfüggvényre példa az erőforrástípusok átlagos kihasználtságának mértéke. Valós példák alapján egy ütemezés beváltságának kifejezése nehezen fogalmazható meg egyetlen célfüggvénnyel.

Ekkor egy lehetséges módszer az, hogy egy új, több szempontot figyelembe vevő célfüggvényt alakítunk ki. Ez történhet egy új teljesítménymutató bevezetésével, melyre célfüggvényt lehet definiálni, vagy – praktikusan – egy új összetett célfüggvény kialakításával, melyet az ismert célfüggvények súlyozásával definiálunk.

A kidolgozott megoldási stratégiánkban a célfüggvényértékek relatív változásainak súlyozási módszerét használjuk. Ennek lényege, hogy mindig két lehetséges ütemezést hasonlítunk össze. Több célfüggvény szerint kiszámítjuk a két megoldás értékeit. Adott célfüggvény esetében az értékek különbségét elosztjuk a nagyobb értékkel, így megkapjuk a relatív változás előjeles értékét. Ezt elvégezve minden célfüggvény esetében, majd az elemi relatív változásokat összegezve megkapjuk a változások eredőjét. Ha a célfüggvények nem egyformán fontosak, akkor súlyfaktorok hozzárendelésével fejezzük ki azok fontosságát. Végül az elemi relatív változások súlyozott összege mutatja meg a két vizsgált megoldás egymáshoz viszonyított hatékonyságát. Ez a módszer tetszőleges számú és típusú célfüggvényre alkalmazható.

3.2. Párhuzamosan futó projektek

Az RCPSP-feladat egy projektet és az ahhoz tartozó feladatokat és megkötéseket írja le. Ennek kiterjesztéseként modellezzük azt az esetet, amikor az erőforrásokat nem egyetlen projekt feladataihoz kell hozzárendelnünk, hanem több, párhuzamosan futó projekt feladataihoz. Az egyes projektek egymástól nemcsak a végrehajtandó feladatokban és korlátozó feltételekben térnek el, hanem eltérő célfüggvényekkel is rendelkezhetnek.

Egy lehetséges módszer, hogy a feladatok halmazához hozzáadunk két új feladatot az alábbiak szerint:

- egy virtuális, globális GS-projekt-kezdőfeladatot, GS ∈ T. A GS-feladat minden eredeti projekt, minden előfeltétel nélküli feladatának megelőző feladataként vezetjük be;
- egy virtuális, globális GT-projektzárófeladatot, GT ∈ T. A GT feladatnak előfeltétele lesz minden olyan feladat, amely az eredeti definíció szerint nem előfeltétele más feladatnak.

Ezzel a modellezéssel a párhuzamosan futó feladatok problémája egy egyprojektes ütemezési feladattá redukálható, azonban a projektek közötti eltérő célfüggvények miatt egy új, egyesített célfüggvény létrehozása szükséges. Ez a gyakorlatban sokszor nehézségbe ütközik.

4. Alkalmazott modell

A kiterjesztett feladat modellezése során az RCPSP-probléma alapelemeiből indultunk ki. Így értelmezzük a feladatokat, a projekteket, az erőforrástípusokat. Egy projekt megadásakor a projekt azonosító adatain és a projekthez rendelt feladatokon kívül a projekthez tartozó célfüggvények súlyát lehet megadni. A lehetséges célfüggvények listája a jelenlegi implementációban előre megadott, de az alkalmazott szoftverarchitektúra lehetővé teszi később saját célfüggvények implementálását.

A rendelkezésre álló erőforrástípusokat projektektől függetlenül lehet megadni. Egy erőforrástípust az azonosítóján kívül a kapacitását leíró függvénnyel lehet megadni. Jelenlegi megvalósításban egy erőforrástípus kapacitáskorlátja konstans függvénnyel írható le, de a kialakított architektúra lehetővé teszi időben változó kapacitáskorlát modellezését is.

A végrehajtandó feladatokat, a feladatok erőforrástípusonként szükséges kapacitásigényét, valamint a feladatok közötti megelőzési relációt feladatonként lehet megadni.

5. Alkalmazott ütemező

Az alkalmazott ütemezőalgoritmus két fő komponensből áll: egy generálási sémán alapuló ütemterv készítőt, illetve egy heurisztikus keresőt integrál.

A generálási séma elvén alapuló projektütemező működését tekintve egy üres ütemtervből indul ki, és minden iterációban a még nem ütemezett feladatok listájából egy feladatot ad a projektütemtervhez. Az iterációban a következő ütemezendő feladat választásakor a megadott korlátokat mindig figyelembe veszi, azaz csak olyan feladatot tekint végrehajthatónak, melynek minden előfeltétele teljesült, és a szükséges erőforrásigénye egyszerre kielégíthető. A végrehajtható feladatok meghatározását tekintve megkülönböztetünk soros és párhuzamos generálási sémát. A soros ütemezési séma elsődlegesen az előfeltételek teljesülését veszi figyelembe. A döntési halmazba azok a feladatok kerülnek, amelyeknek minden előfeltétele teljesült. Ezek közül az aktuális pillanatnak megfelelően a legnagyobb prioritású (összességében a legfontosabbnak tűnő) feladat kerül kiválasztásra. Ezt a feladatot ütemezi be az algoritmus úgy, hogy az erőforrástípusokból igényelt kapacitások együttállása szempontjából a lehető legkorábbi kezdési időponttal rögzíti az ütemezésben.

A párhuzamos generálási séma elsődlegesen a feladatok legkorábbi indíthatóságára helyezi a hangsúlyt. Egy közbenső állapotban a döntési halmazt úgy állítja elő, hogy meghatározza a legkorábban indítható feladatok listáját. A soros generálási sémától eltérő módon, itt nem az összes indítható munkát veszi alapul a döntési halmaz meghatározásához. A döntési halmazból viszont ugyanazzal a prioritásalapú módszerrel választ jelöltet, amelyiket a soros séma is használja.

Amennyiben több feladat is megfelel a kritériumoknak, (azonos prioritásúak) úgy a generálási séma az ilyen feladatok közül véletlenszerűen választ.

Megközelítésünkben a véletlenszerű választás helyett egy determinisztikus kiegészítést alkalmaztunk, hasonlóan más publikált kiegészítésekhez. Egyetlen heurisztikus módszer helyett azonban több feladatkiválasztási heurisztikát együttesen alkalmazunk, melyek súlyát a felhasználó által megadott paraméterek alapján vesszük figyelembe. A generálási séma működése így közvetlenül a felhasználó által megadott paraméterek szerint befolyásolható.

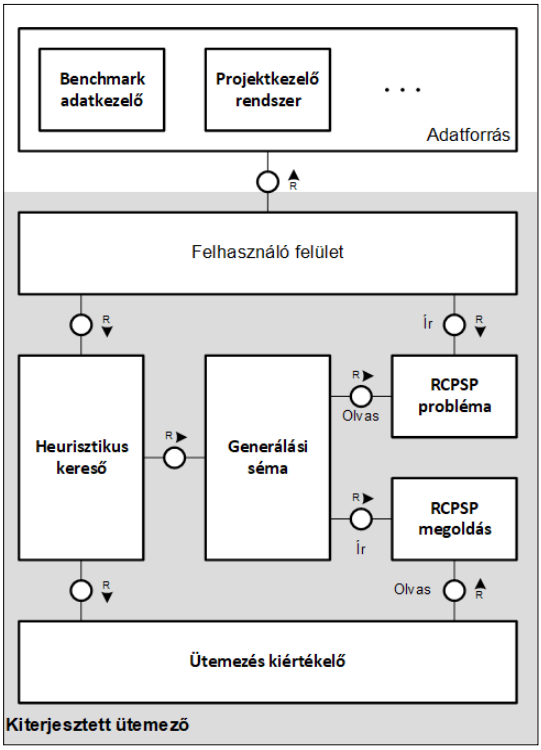
A generálási sémát szimulációs modulként tekintve, egy heurisztikus keresőt is implementáltunk, mely során a heurisztikus kereső feladata a feladatkiválasztási heurisztikák súlyának iteratív módosítása.

6. Megvalósítás

A kiterjesztett probléma megoldására egy új ütemező szoftvert fejlesztettünk ki, melynek fő moduljait szemlélteti az **1. ábra**.

Minden modul objektumorientált ABAP felhasználásával került kialakításra. Az egyes modulok egymástól ABAP-interfészek használatával elkülönítettek. Az alkalmazott interfészek célja a modulok szeparált funkcionális tesztautomatizálásának lehetősége és az implementációs alternatívák rugalmasságának elősegítése.

Az egyes modulok által megvalósított funkciók és a modulok felelősségi körei:



1. ábra. *Kiterjesztett ütemező moduláris felépítése*

– RCPSP-probléma
Feladata a kiterjesztett modell adatainak futásidejű tárolása. Modellezett írási és olvasási interfésszel rendelkezik. A modellezett interfészek célja, hogy a használati viszony más modulokkal jól szeparált legyen;
– RCPSP-megoldás
Feladata a kiterjesztett modell egy megvalósítható ütemezésének reprezentálása. Tárolja az egyes feladatok hozzárendelésének sorrendjét és ütemezési idejét a különböző erőforrástípusokhoz. Modellezett írási és olvasási interfésszel rendelkezik;
– Generálási séma
Feladata a kiterjesztett RCPSP-probléma soros generálásiséma-alapú megoldása, melyhez bemenetként az RCPSP problémaolvasási interfészét, valamint a választási lépést befolyásoló szelekciós heurisztikai paramétereket és súlyokat leíró adatát használja. A generálási séma a determinisztikusan kialakított megoldást az RCPSP-megoldás-író interfészén keresztül tárolja el;
– Heurisztikus kereső
A heurisztikus kereső a felhasználó által megadott célfüggvények, a korábban generált RCPSP megoldás(ok) alapján módosít a heurisztikus vezérlőparamétereken, majd újabb RCPSP megoldás(oka)t generál;
– Ütemezés kiértékelő
Az ütemezési kiértékelő a végrehajtható ütemezést értékeli ki a felhasználó által megadott projek célfüggvények alapján;
– Adatforrás
Több adatforrás-illesztő áll rendelkezésre, melyek különböző céllal rendelkeznek. A benchmark adatkezelő feladata, hogy ismert és más módon leírt ütemezési feladatokat a kiterjesztett modell szerinti leírásra készítsen. A projektkezelő rendszerillesztő feladata, hogy SAP PM-modul által leírt projektek adatait a kiterjesztett ütemező által elvárt formátumra képezze. További lehetséges adatforrás a példafeladatok adatbázisa, mely szerepe az implementáció helyességének ellenőrzése integrációs teszteken keresztül.

7. Eredmények

A kiterjesztett modellt és a megtervezett ütemezőt implementáltuk a tanszéken telepített SAP Netweaver 7.50fejlesztői tesztkörnyezetben. A modell- és az algoritmusimplementáció nem tartalmaz SAP-specifikus elemeket, azok más, objektumorientált nyelvre leképezhetőek. Választásunk oka, hogy SAP PM-modullal történő integráció SAP-platformon lehessen megvalósítható a későbbiekben.

Az elkészült új implementációt összevetettük ismert alapproblémákra ismert megoldásokkal, mely során két módszert alkalmaztunk.

Az elsőben a kiterjesztett modellre képeztünk olyan speciális megkötésekkel rendelkező ütemezési feladatokat, amelyekre létezik ismert, optimumot szolgáltató ütemező algoritmus. Ismeretes, hogy a kétgépes, előzés nélküli Flow Shop-ütemezési feladatra a Johnson-algoritmus optimális megoldást ad C_{max} célfüggvény esetén [3]. A keresési eljáráson alapuló modellünk kis problémák esetén nagy valószínűséggel (>95%) megtalálta az optimumot.

Második esetben az alap-RCPSP-problémát képeztük le a kiterjesztett modellre, és C_{max} célfüggvény szerint publikált legjobb eredményekkel vetettük össze [4], [5]. A benchmark feladatokra futtatva az ütemezőt összehasonlítottuk az ismert legjobb eredményekkel. Azt tapasztaltuk, hogy kisméretű problémák esetén döntő többségben (>70%) az ismert legjobb megoldást megtalálta az ütemezőrendszerünk.

8. Következtetések

Cikkünkben összefoglaltuk a kiterjesztett, erőforrás-korlátos, többprojektes, többcélú ütemező modellünk legfontosabb jellemzőit.

A kiterjesztett modell alkalmas a már korábban ismert feladatok kezelésére. Figyelembe véve a rugalmasságból adódó eltérést, a specializált algoritmusok jobb teljesítményt mutatnak.

Az általunk javasolt megoldási koncepció lehetővé teszi nagyon sokféle ütemezési feladat megoldását. Ez az implementált szoftver segítségével tudtuk igazolni. A klasszikus ütemezési feladatok jelentős része megfogalmazható a kiterjesztett feladat egy speciális eseteként. Ezáltal a szoftver támogatja az ilyes feladatok megoldását. Ez a gyakorlat szempontjából nagyon fontos. A különböző cégek számára egy ilyen alkalmazás nem igényel egyedi fejlesztést, nem kell a rendszerbe forráskódszinten újabb elemeket hozzáadni.

Az elért eredmények alapján folytatjuk a megoldási algoritmusok és az implementáció fejlesztését.

A továbbfejlesztés egyik fő iránya az, hogy újabb generálási sémát dolgozunk ki az ütemezési stratégia finomítása érdekében. Egy másik tervezett irány az, hogy a megoldás előállításának vezérfonalát jelentő, heurisztikus algoritmust kiegészítjük új módosító operátorokkal. A fejlesztési irányok közé tartozik továbbá az is, hogy az ütemezendő projektelem kiválasztásakor új összehasonlító algoritmust fejlesztünk ki.

Szakirodalmi hivatkozások

[1] Garey M. R., Johnson D. S.: *Computers and Intractability; A Guide to the Theory of NP-Completeness.* 1st Edition, Series of Books in the Mathematical Sciences, W. H. Freeman; 1979.

[2] Kolisch R, Hartmann S.: *Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis*. In: Weglarz J. (ed) *Project scheduling. Recent Models, Algorithms and Applications*. International Series in Operations Research & Management Science 14., Springer, 1999, 147–78.

[3] Johnson, D. B.: *Efficient Algorithms for Shortest Paths in Sparse Networks*. Journal of the ACM, 24/1. (1977) 1–13. https://doi.org/10.1145/321992.321993

[4] Kolisch R., Sprecher A.: *PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program*. European Journal of Operational Research, 96/1. (1996). 205–216. https://doi.org/10.1016/S0377-2217(96)00170-1

[5] Project Scheduling Problem Library – PSPLIB: *Datasets*. . (letöltve: 2019. december 1.) http://www.om-db.wi.tum.de/psplib/data.html