

Egylépéses gyártási feladatok költségoptimális  
ütemezése időzített automatával

Vida Judit  
Témavezető: Dr. Hegyháti Máté  
Széchenyi István Egyetem

2018. február 19.

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>2</b>
<b>2. Irodalmi áttekintés</b>	<b>3</b>
2.0.1. Az egylépéses ütemezés ábrázolása . . . . .	4
2.0.2. MILP modellek . . . . .	4
2.0.3. S-gráf . . . . .	5
2.0.4. Petri háló . . . . .	5
<b>3. Problémadefiníció</b>	<b>6</b>
<b>4. Időzített automaták</b>	<b>7</b>
4.1. Időzített automaták . . . . .	7
4.2. UPPAAL Cora . . . . .	7
<b>A. Jelölések</b>	<b>9</b>

# 1. fejezet

## Bevezetés

Ütemezési problémákkal az élet számos területén találkozhatunk, hiszen hétköznapi feladatainkat is be kell osztanunk. Vannak azonban speciális problémák, amelyre már léteznek kidolgozott megoldások, és adottak egyéb megoldási lehetőségek is. Dolgozatomban egy gyártási probléma ütemezését választottam, amelyet költségfüggvénnyel kiterjesztett időzített automata használatával optimalizálok, és elemzem az eredményeket.

Az irodalomban több lehetséges megoldó módszerről esik szó, egy lépéses gyártási probléma ütemezését elvégezték már többféle technika segítségével. A leggyakoribbak a MILP megoldó módszerek, melyek vegyes-egész lineáris programozáson alapulnak, de az S-gráf alapú megoldások is jellemzőek, valamint Petri-hálóval is oldottak már meg hasonló ütemezési feladatokat. Ezekről az eredményekről szó esik a kapcsolódó irodalomban is, amelyeket fel tudunk használni arra, hogy az automatával elvégzett optimalizálás eredményeivel összehasonlítsuk őket.

A feladat célja, hogy minél több feladatot elvégezzenek a gépek, és minél kevesebb legyen a kész termékek tárolási költsége.

Időzített automatával még nem járták körül bővebben a problémát, de érdemes vele foglalkozni. A feladat egy irodalmi példa, amelyben a meghatározott számú feladatot a gépek egy lépésben végzik el. Dolgozatom második fejezetében az irodalmi háttérrel foglalkozom, a harmadik fejezetben részletezem a problémát, majd az időzített automatákról, a használt szoftverről és az LPTA alapú ütemezésről teszek említést. A negyedik fejezetben a teszteredményekről lesz szó, amelyeket korábbi megoldásokkal hasonlítok össze. A végén található az összefoglalás az ütemezés eredménye alapján, majd a dolgozat végére kerülnek a hivatkozások és a függelék.

## 2. fejezet

# Irodalmi áttekintés

Bármely ütemezési feladat alapját ugyanazok a megadott paraméterek adják, bármilyen módszert használunk a megoldáshoz. Közös bennük, hogy adottak a feladatok, a feladatokat elvégezni képes berendezések vagy eszközök, egy adott időintervallum minden feladathoz, a cél pedig, hogy a kijelölt feltételek mellett kielégítsük a meghatározott végcél. Az egyes feladatok intervalluma az az időtartam, ami a munka elvégzéséhez szükséges, ez általában minden esetben egyedien meghatározott, tehát minden különböző feladatnak eltérő mennyiségű időre van szüksége. A feladatoknak lehet elvégzési határideje is, amit szintén figyelembe kell venni az ütemezés folyamán. Az irodalomban a feladatokat task megnevezéssel is használják, a munkát elvégző gépeket unit-ként, a termékeket pedig product-ként.

A problémákat különböző szempontok alapján lehet kategorizálni, az egyik csoportosítás alapján megkülönböztetünk sztochaikus és determinisztikus ütemezési feladatokat, ahol a sztochaikus típus azokat a problémákat jelöli, ahol a paraméterek futás közben kapnak értéket. A determinisztikus esetében az értékek előre be vannak állítva. A két csoportot viszont nem lehet élesen elkülöníteni egymástól, mivel lehetnek olyan esetek is, ahol egyes paramétereknek muszáj még futtatás előtt értéket adni, míg a többi tényező futás közben veszi fel őket. Szintén két csoportba oszthatóak a problémák a következő szempontból, ha az ütemezés nem elégít ki legalább egy korlátozást, akkor az nem megoldható (infeasible), minden egyéb esetben megoldható, tehát feasible.

Az ütemezési feladatok egyik altípusa az egylépéses ütemezési probléma, ahol a feladatokon egy tevékenységet kell végrehajtani, hogy azt befejezetté lehessen nyilvánítani. Az egylépéses problémák mellett vannak más gyakori típusok is, például a simple multiproduct, ahol a feladatot lineárisan több lépésben kell elvégezni, a general multiproduct, ami a simple multiproduct-hoz hasonló, de ki lehet hagyni lépéseket. A multipurpose (többcélú) problémátípusban a lépéseknek nincs meghatározott sorrendje, tetszőlegesen hajthatók végre. Megkülönböztethetjük a precedens típust, amely hasonló a többcélúhoz, de nem feltétlenül lineárisan hajtódnak végre a feladatok, és a general network típust, ahol a feladatokat az inputjaik és outputjaik alapján különböztetnek meg.

/\*Az ütemezési feladatokat típusuktól függetlenül egyéb paramétereik alapján is csoportosítják, a kiválasztott problémánkat UIS (Unlimited Intermediate Storage - )\*/

A single stage problémáknak a megoldásához néhány paraméternek adottnak kell lennie, például a gépek számának, valamint annak, hogy ezek a gépek azonosak-e. Két gép akkor tekinthető azonosnak, ha ugyanazokat a munkákat képesek elvégezni ugyanannyi idő alatt. Szükség van továbbá a feladatok számára és típusára. A gépek az alábbiakban felsorolt típusúak lehetnek.

- **1 - Single Machine:** Egy gép elérhető, amelyen minden feladatot végre lehet hajtani. A termékeknek (feladatoknak) különböző feldolgozási ideje van.
- **Pm - Identical paralell machines**(Azonos párhuzamos gépek): m számú azonos gép áll rendelkezésre, amelyeken bármelyik egy lépéses feladat végrehajtható.
- **Qm - Paralell machines with different speed** (Párhuzamos gépek különböző sebességgel): Hasonló az előző pontban említett típushoz, de minden gépnek meghatározott sebessége van.
- **Rm - Unrelated machines in paralell**(Párhuzamos független gépek): Hasonló a Single stage típushoz, de a feladatok elvégzési ideje egy-egy gépen inputként meghatározott.

### 2.0.1. Az egy lépéses ütemezés ábrázolása

Az egy lépéses ütemezési problémákat általában táblázat segítségével adják meg, ahol a sorokban tüntetik fel a munkákat, az oszlopokban pedig a rendelkezésre álló berendezéseket, a táblázatbeli metszéspontjaik ábrázolják az egyes munkák megfelelő berendezéseken való elvégzésének munkaidejét. Kopanos mutatott be erről egy esettanulmányt.

Az egy lépéses ütemezési feladatok témakörével még nem foglalkoztak részletesen az irodalomban, viszont vannak olyan módszerek, amelyekkel már a legtöbb problémaosztály szempontjából foglalkoztak. Az alábbi néhány eljárás a legnépszerűbb megoldók közé tartozik.

### 2.0.2. MILP modellek

A MILP modellek, tehát a vegyes-egész lineáris programozási modellek a legelterjedtebb megoldó módszerek közé tartoznak, és több altípusuk létezik.

#### Time discretization based - Időfelosztásos módszerek

Az időfelosztásos modellek előnye, hogy széles skálán mozog a megoldható problémák típusa. A módszer alapján időpontokat és időréseket különböztetünk meg. A időrés az az intervallum, amely egyik időponttól a következőig tart, az

időpont pedig ennek a fordítottja, tehát az időrés kezdete az egyik időpont, az időrés vége pedig egy másik.

A feladatokhoz bináris változókat rendelnek aszerint, hogy a feladat az adott időpontban elvégzésre kerül, vagy nem. Amikor a feladat abban az időpontban megvalósul, akkor 1 lesz a bináris változó értéke, ha nem, akkor 0. Így annyi bináris változóra lesz szükség, ahány időpontot meghatároztunk. Lehetőleg minél kisebb számú időpont felvételével kell megtalálni az optimális megoldást a modell bonyolultságának csökkentése érdekében.

Két altípust különböztetnek meg az alapján, hogy az időpontokat az optimalizálás előtt meghatározzák, ezek a Fix időpontos időfelosztásos módszerek, vagy pedig csak a feladat közben kerül meghatározásra az időpontok száma. Utóbbiakat Variable time model-eknek hívják, és a lényegük, hogy minél kevesebb bináris változóra legyen szükség. A modellekben folyamatos változókat használnak, amik meghatározzák mindegyik időponthoz tartozó feladatokat.

### 2.0.3. S-gráf

Az első gráf alapú optimalizációra fejlesztett módszer, amely nemcsak vizuálisan szemlélteti a folyamatot, de egyben egy matematikai modell is. Irányított gráfokból áll, amelynek a csomópontjai feladatok és az azokon elvégzendő műveletek, amelyeket az élek kötnek össze őket. Ezen kívül tartalmaznak ütemezési íveket (nyilakat), amelyek a meghozott ütemezési döntéseket modellezzik.

Létezik ütemezési döntések nélküli S-gráf is, ezt recept gráfnak hívják.

A nyilak, amelyek a csomópontokat kötik össze, a függőségeket reprezentálják a következő esetekben:

### 2.0.4. Petri háló

A Petri hálót és az időzített automatát is gyakran alkalmazzák diszkrét esemény rendszerű ütemezési problémákhoz, és hogy kötegelt feladatok elvégzéséhez is alkalmas legyen, ki kellett egészíteni időzítéssel ezeket a módszereket. Hatékonyak, mivel jól szemléltetik a modellt, és a megfelelő felépítés mellett elkerülhetőek a hibák. Bár több előnyük is van a korábban említett népszerű megoldó módszerekhez képest, összességében hatékonyságuk még elmarad a MILP és S-gráf alapú megoldó módszerektől.

Az időzített Petri háló alapja, hogy az átviteli jel késleltetés (delay) alapján jön létre. Többen is foglalkoztak a témával, Ghaeli foglalkozott a kötegelt folyamatok ütemezésével ilyen módon, Soares pedig megpróbálta kiterjeszteni a modellt, és valós idejű ütemezést mutatott be kötegelt rendszerekre Petri háló segítségével.

### 3. fejezet

## Problémadefiníció

A probléma egy egylépéses szakaszos eljárás ütemezéséhez kapcsolódik, ahol minden termék egy termelési lépés alatt készül el, ezeket a hívjuk munkáknak. A munkákat bármelyik gép (unit) elvégezheti, de egy munka csak egy géphez rendelhető hozzá. Ugyanígy egy gép egyszerre csak egy feladaton dolgozhat, és ha már egy munkát elkezdett, azt egy másik nem előzheti meg.

Adott a munkák és a gépek száma, valamint a gépeknek van egy meghatározott üzembe állási ideje, ez mindegyik berendezés egyedi tulajdonsága. Két munka elvégzése között felszámolunk átállási időt, amíg a gép testre szabja saját beállításait a következő feladathoz. A gyakorlatban ez tisztítást, újra beállítást és egyéb karbantartást jelent. Az átállási időt kétféleképpen lehet megadni, az egyik típus a szekvenciafüggő, amikor a feladatok sorrendje határozza meg az értéket. Mennyiségét az szabja meg, hogy az előző és az utána következő munka között mennyi időre van szüksége a berendezésnek. A másik megadási mód a szekvenciafüggetlen típus, amikor csak a berendezéstől függ az átállási idő. Mindkét modellt be lehet állítani úgy, hogy minkét típus függjön a géptől és a feladattól is. Itt...

A munkákat minden gép különböző idő alatt tudja elvégezni, de olyan eset is lehet, amikor egy gép nem tudja elvégezni az adott munkát. Minden feladat rendelkezik határidővel, amit nem léphet át, miközben várakoznia kell, ha a határidő előtt elkészül. Az ütemezés célja, hogy minimalizáljuk a várakozás költségeit, emellett viszont előfordulhat, hogy a megoldás nem elégíti ki a korlátozásokat, így infeasible lesz. Ha van feasible megoldás, szeretnénk lehetőleg az összes munkát elvégezni határidőre, valamint a modellt kiegészíteni korlátozásokkal úgy, hogy minél kevesebb idő alatt elvégezze az ütemezést, és minél optimálisabb eredményt adjon.

A feladatban munkákat és gépeket különböztetünk meg, amelyeket később P-vel és U-val jelölünk, a product és unit szakirodalomban használt megnevezések után.

## 4. fejezet

# Időzített automaták

### 4.1. Időzített automaták

### 4.2. UPPAAL Cora



# Hivatkozások

bibtex

A. függelék

Jelölések