

**BUDAPESTI GAZDASÁGI FŐISKOLA
GAZDÁLKODÁSI KAR ZALAEGERSZEG**

Vállalatirányítási rendszerbe integrált termelésütemező rendszer specifikációja az Anton Kft. számgépjártási tevékenységére

Belső konzulens: Dr. Gubán Miklós PhD

Külső Konzulens: Dr. Németh András PhD

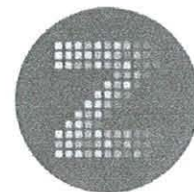
Sztárcsevics Gábor

Nappali tagozat

Gazdaságinformatika alapszak

Logisztikai informatikus szakirány

2015



NYILATKOZAT

a szakdolgozat digitális formátumának benyújtásáról

A hallgató neve: Sztárcsevics Gábor

Szak/szakirány: Gazdaságinformatika szak/Logisztikai informatikus szakirány.

Neptun kód: KJJ3ZD.

A szakdolgozat megvédésének dátuma (év): 2015.

A szakdolgozat pontos címe:

Vállalatirányítási rendszerbe integrált termelésütemező rendszer specifikációja az Anton Kft. szerszámgyártási tevékenységére.

Belső konzulens neve: Dr. Gubán Miklós PhD.

Külső konzulens neve: Dr. Németh András PhD.

Legalább 5 kulcsszó a dolgozat tartalmára vonatkozóan:

Vállalatirányítási rendszer, termelésütemezés, egyedi gyártás, matematika, Anton Kft.

Benyújtott szakdolgozatom **nem titkosított / titkosított.**

(Kérjük a megfelelőt aláhúzni! Titkosított dolgozat esetén kérjük a titkosítási kérelem egy eredeti példányát leadni: a kérelem digitális másolatának a szakdolgozat digitális formátumában szerepelnie kell.)

Hozzájárulok / nem járulok hozzá, hogy nem titkosított szakdolgozatomat a főiskola könyvtára az interneten a nyilvánosság számára közzétegye. *(Kérjük a megfelelőt aláhúzni!)* Hozzájárulásom - szerzői jogaim maradéktalan tiszteletben tartása mellett – egy nem kizárólagos, időtartamra nem korlátozott felhasználási engedély.

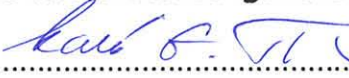
Felelősségem tudatában kijelentem, hogy szakdolgozatom digitális adatállománya mindenben eleget tesz a vonatkozó és hatályos intézményi előírásoknak, tartalma megegyezik a nyomtatott formában benyújtott szakdolgozatommal.

Dátum: 2015.01.05.


.....
hallgató aláírása

A digitális szakdolgozat könyvtári benyújtását és átvételét igazolom.

Dátum: 2015.01.05.


.....
könyvtári munkatárs

P.H.

Tartalomjegyzék

Köszönetnyilvánítás	5
Bevezetés.....	6
1. Jelölésjegyzék.....	7
2. A termelésütemezés elméleti alapjai	10
2.1.1. Az ütemezés vállalati funkciója	13
2.1.2. Vállalatirányítási rendszer modell.....	13
2.2. Determinisztikus modellek	15
2.2.1. Egygépes ütemezés	15
2.2.2. Párhuzamos gépes ütemezés	16
2.2.3. Ütemezés független gépeken.....	18
2.2.4. Ütemezés uniform gépeken.....	18
2.2.5. Az open shop feladat	19
2.2.6. A flow shop feladat	20
2.2.7. A job shop feladat	21
2.3. Sztochasztikus modellek.....	22
2.3.1. Eloszlások és algoritmusok	22
3. Az Anton Kft. szerszámgyártási folyamata.....	25
3.1. A vállalat bemutatása.....	25
3.2. Termelésütemezés az Anton Kft. üzemeiben	26
3.3. Szerszámgyártási tevékenység	26
3.4. Egyedi gyártási folyamat sajátosságai	28
3.4.1. Egyediség	28
3.4.2. Projekt prioritások.....	29
3.4.3. Darabjegyzék, művelettervek.....	29
3.4.4. Műveletterv alternatívák	29
3.4.5. Alvállalkozói műveletek	30

3.4.6.	Szakaszos anyagellátás	30
3.4.7.	Kötött műszakrend	30
3.4.8.	Zavaró tényezők, javítások	31
4.	Az ütemező rendszer specifikációja	32
4.1.	Gépek modellezése	32
4.1.1.	Kalendáriumok	33
4.2.	Munkák modellezése	33
4.2.1.	Gyártási rendelések	34
4.2.2.	Műveleti tervek	34
4.2.3.	Precedenciák.....	35
4.3.	Anyagok modellezése.....	36
4.4.	Szerszámok modellezése	36
4.5.	Emberi erőforrás modellezése	37
4.6.	Korlátozások.....	37
4.6.1.	Csoportos végrehajtás	37
4.6.2.	Alapanyagok.....	37
4.7.	Célfüggvények.....	37
4.8.	Globális paraméterek	38
4.8.1.	Alapértelmezett késleltetés.....	38
4.8.2.	Időhorizontok	38
4.9.	Ütemterv	38
4.9.1.	Végrehajtási idők számítása	39
4.9.2.	Kezdési és befejezési idők számítása	39
4.9.3.	Félkész termékek raktárkészleteinek nyilvántartása	40
5.	Termelésütemező szoftverek.....	41
5.1.	A termelésütemező szoftverek összehasonlítása	41
6.	Bevezetési terv	46

6.1. A bevezetés folyamata.....	46
7. Összefoglalás.....	48
8. Irodalomjegyzék.....	51

Köszönetnyilvánítás

Ezúton szeretném megköszönni Dr. Németh András vállalati konzulensemnek, hogy szak-tudásával és ötleteivel segített a szakdolgozatom megírásában, illetve tapasztalatai átadásával megismerhettem egy vállalat működését.

Továbbá szeretném hálám kifejezni Dr. Gubán Miklós főiskolai konzulensemnek, hogy időt és energiát nem sajnálva segített, hogy a dolgozatom elkészülhessen. Meglátásai, szakmai és matematikai ismeretei által a munkám értékesebbé válhatott.

Végül, de nem utolsó sorban köszönetemet fejezem ki Zsupanekné Dr. Palányi Ildikó, és Dr. Szekeres Diána Tanárnőknek, akik folyamatos segítségükkel támogattak a tanulmányaim során.

Bevezetés

Diplomamunkám keretében az Anton Kft. szerszámgyártási tevékenységére szabott termelésütemező rendszer specifikációját készítem el. Ennek célja a későbbiekben bevezetendő rendszer kiválasztásának és bevezetésének előkészítése. Ennek segítségével a vállalat vezetősége döntést tud hozni a megfelelő ütemező rendszer kiválasztásában.

Mivel a dolgozat nagyban támaszkodik matematikai összefüggésekre és jelölésekre, ezért a 2. fejezetben bemutatom az ütemezéselmélet matematika alapjait, különböző csoportosításokban.

A dolgozat 3. fejezetében bemutatásra kerül Anton Kft. szerszámgyártási tevékenysége, minden fontos jellemzőjével együtt, mely teljes képet fog adni a feladat átgondolásához, és az ütemező rendszer alapkövetelményeinek meghatározásához.

A 4. fejezet a specifikáció meghatározását tartalmazza, definiálva a szükséges paramétereket a rendszerhez. Ezen felül tartalmazza a termelési modellt, a különböző erőforrások, korlátok ismertetését és a különböző célfüggvény alternatívákat is bemutatja.

A piacon számos termelésstervezési szoftver található, ezekből néhány potenciálisan használható szoftvert mutat be az 5. fejezet. Ezen programok a vállalat által meghatározott szempontok szerint táblázatos formában vannak összehasonlítva egymással. Ezek alapján javaslatot teszek a megfelelőnek tartott rendszer kiválasztására.

A 6. fejezet egy bevezetési terv, mely általánosságban bemutatja, hogy egy ilyen rendszer beüzemelése esetén milyen tevékenységeket kell elvégezni, és ezek hozzávetőlegesen mennyi időt vesznek igénybe egy vállalatnál.

A 7. fejezetben összefoglalom a modellezés tapasztalatait, a legfontosabb következtetéseket, valamint a javasolt rendszerek kiválasztásának indoklását. Bemutatom továbbá a bevezetéshez kapcsolódó feladatokat, a szükséges erőforrásokat és az implementáció időtervét.

1. Jelölésjegyzék

Jelölés	Megnevezés	Mértékegység
p_j	A j -edik munka megmunkálási ideje	időegység
p_{ij}	A j -edik munka i -edik gépen töltött megmunkálási ideje	időegység
(a_j, b_j)	A j -edik munka első és második gépen töltött megmunkálási ideje	időegység
r_j	Rendelkezésre állási idő	időegység
d_j	A j -edik munka határideje	időegység
w_j	A j -edik munka súlya	
C_j^S	Az az időpillanat az S ütemezésben, amikor a j -edik munka elkészül	időegység
L_j^S	A j -edik munka késése az S ütemezésben	
U_j^S	Az S ütemezésben j -edik munkához rendelt 0 v. 1 érték, attól függően, hogy a késés negatív, vagy pozitív	
s_i	Az i -edik gép sebessége	darab/időegység
$prec$	Megelőzési feltételek definiáltak	
$pmtn$	Megszakíthatósági feltétel teljesül	
P, Q, R, O, J, F	A gép- és feladattípusok jelölése (Párhuzamos, Uniform, Független, Open shop, Job shop, Flow shop)	
C_{max}^S	Teljes átfutási idő	időegység
$\frac{\sum_{j=1}^n C_j^S}{n}$	Átlagos átfutási idő	időegység
$\sum_{j=1}^n w_j C_j^S$	Súlyozott átlagos átfutási idő	időegység
L_{max}	Maximális késés	időegység
$\sum_j w_j U_j^S$	Késéssel elvégzett munkák számának súlyozott összege	
C_{max}^*	Optimális teljes átfutás idő	időegység
J_{max}	A leghosszabb munka megmunkálási ideje	időegység
M_{max}	A legtöbbet dolgozó gép futási ideje	időegység
X_{ij}	Sztochasztikus modelleknél az i -edik gépen a j -edik munka megmunkálási ideje	időegység

$1/\lambda_{ij}$	X_{ij} várható értéke	időegység
$parent_l$	szülő hierarchieaelem az l hierarchiacsoportban	
$Children_l$	gyerek hierarchieaelemek halmaza az l hierarchiacsoportban	
stp_l	átállási idő az l hierarchiacsoportban	időegység
η_l	géphatékonyság az l hierarchiacsoportban	
ξ_l	átállási idő minimalizálandó-e az l hierarchiacsoportban	logikai változó
Cal_l	az l hierarchiacsoport kalendáriuma	
$start_c$	érvényesség kezdete a c kalendáriumban	Dátum, perc felbontással
end_c	érvényesség vége a c kalendáriumban	Dátum, perc felbontással
$avail$	az érvényességi időben a bejegyzés munkát vagy szünetet határoz meg	Logikai változó
$product_j$	tételkód	
d_j	határidő	dátum
$type_j$	projekt típusa, valós v. tervezett rendelés	
$OpPlan_{j,k}$	műveleti terv a $PO_{j,k}$ gyártásrendelésben	
$qty_{j,k}$	A $PO_{j,k}$ gyártási rendelésben szereplő alkatrészek darabszáma	
$r_{j,k}$	A $PO_{j,k}$ gyártási rendelésben szereplő legkorábbi kezdési idő	időegység
$Precs_p$	A műveletek közötti precedenciák a p műveletterv esetén	
Ops_p	A műveletek halmaza a p műveletterv esetén	
$id_{j,k,o}$	műveletazonosító a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	
$class_{j,k,o}$	műveletosztály a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	
$L_{j,k,o}$	géphierarchia-elem a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan; az ebbe tartozó gépek közül bármelyien elvégezhető a	

	művelet	
$BOM_{j,k,o}$	darabjegyzék a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	
$ToolTypes_{j,k,o}$	a művelet elvégzéséhez szükséges szerszámtípusok a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	
$minBatchSize_{j,k,o}$	minimális sorozatnagyság a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	
$t_{j,k,o}^{queue}$	sorbanállási idő a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	időegység
$t_{j,k,o}^{stp}$	beállítási idő a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	időegység
$t_{j,k,o}^{proc}$	műveleti idő a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	
$t_{j,k,o}^{extern}$	alvállalkozói idő a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	időegység (általában egy fix intervallum, amíg visszajön a szállítmány az alvállalkozótól)
$t_{j,k,o}^{delay}$	anyagmozgatási idő a $PO_{j,k}$ gyártási rendelésben az o művelethez kapcsolódóan	időegység
$r_{j,k,o}$	az alapanyag rendelkezésre állásából számított legkorábbi kezdési idő	időegység
$\tilde{r}_{j,k,o}$	kézzel megadott legkorábbi kezdési idő	időegység
$S_{j,k,o}$	tervezett kezdési idő	időegység
$C_{j,k,o}$	tervezett befejezési idő	időegység
$\hat{C}_{j,k,o}$	lejelentett befejezési idő	időegység
$\tilde{C}_{j,k,o}$	befejezési idő ígéret	időegység
$qtyfnsh_{j,k,o}$	már elkészült darabszám	
P_p	Anyag (p-edik)	
$level_p$	Anyag hierarchia szintje	
Cal_p	Termék rendelkezésre állása az idő függvényében	
T_r	Segédeszköz (r-edik)	
num_r	Rendelkezésre álló segédeszköz	darab

2. A termelésütemezés elméleti alapjai

Termelési feladatainkban a cél az, hogy gyártásra alkalmas gépek segítségével legyártssuk a vevők által igényelt termékeket, valamilyen célfüggvény, vagy célfüggvények figyelembe vétele mellett. Könnyen elképzelhetünk egy rendelést, melyben a vevő minőségi teljesítést vár el, megadja a rendelni kívánt mennyiséget, de fontos szempontként tünteti fel, hogy az áruja minél gyorsabban elkészüljön. Ilyenkor a vevő tulajdonképpen megfogalmazza a célfüggvényt, mely ebben az esetben azt jelenti, hogy úgy kell ütemeznünk a gépek gyártási munkáját, hogy az átfutási idő minimális legyen. Ahhoz, hogy teljes képet kapjunk az ütemezési problémákról, meg kell nézni, hogy pontosan milyen adatokra van szükség, és tisztázni kell a matematikai fogalmakat.

„Ütemezési problémákban *munkák*, elvégzését ütemezzük *gépeken*, bizonyos *feltételek* mellett, egy adott *célfüggvényt* optimalizálva.”[1] Ezen n számú munkákhoz definiálhatunk egy halmazt, méghozzá a $J=\{j_1, j_2, \dots, j_n\}$ ($n \in \mathbb{Z}_+$) halmazt, melyben minden elem egy munkát jelöl az n darabból, az indexétől függően. Ugyanígy definiálhatunk egy halmazt a gépeinkre is, az $\mathcal{M}=\{m_1, m_2, \dots, m_m\}$, mely m darab gépet tartalmaz. A gépek száma és típusa nagyban befolyásolja a feladatok megoldhatóságát, valamint az optimális megoldástól való esetleges eltérésben is nagy részt tudhat magáénak. A gépek típusairól az alfejezeteinkben lesz szó.

Minden munkához tartozik egy p_j megmunkálási idő, mely azt az időt jelenti, amennyit a munka az adott gépen tölt. A több gépes feladatok egy részében ezt az értéket már p_{ij} -vel fogjuk jelölni, ami a j -edik munka az i -edik gépen töltött idejét fogja jelenteni. Speciális esetben ezt a jelölést tovább módosíthatjuk, például mondhatjuk azt, hogy legyen az (a_j, b_j) páros a munka első, és második gépen töltött megmunkálási ideje. Egyes feladatokban a munkák megszakíthatóak, ami azt jelenti, hogy a munka félbehagyható, és egy későbbi időpontban, több gép esetén esetleg egy másik gépen befejezhető. Több gép használata esetén meg kell határozni, hogy a munkát melyik gép fejezze be.

Definiálhatunk úgynevezett rendelkezésre állási időket, melyet r_j -vel jelölünk és minden munkához hozzárendeljük ezt az értéket. Ez azt a legkorábbi időpillanatot jelenti, akkor a j -edik munkán elkezdhetünk dolgozni. Néha szükséges súlyozni a munkákat, amit a w_j jelölés segítségével tudunk megoldani, és szintén minden munkához hozzárendelhető. A súlyozások értelem szerűen a munkák fontosságát mutatják meg. Fontos fogalom a határidő, melyet a d_j

érték jelöl. Nevéből adódóan azt az időpillanatot jelöli, amelyre a j munkának el kell készülnie.

Az általunk megadott úgynevezett S ütemezésben szerepelnek C_j^S befejezési idők, mely az az időpillanat, amikor a j munka elkészül. Ha határidős munkákkal dolgozunk, akkor definiálhatunk $L_j^S = C_j^S - d_j$ késéseket is, melyek a befejezési időkhöz viszonyítva adják meg a késés mértékét. Itt jön a képbe egy új paraméter, az U_j^S , melynek értéke 1, ha a munka késése pozitív, egyébként 0. Ez azoknál a feladatoknál fontos, amikor a késő munkák számát kell minimalizálni, tehát azt szeretnénk, hogy minél kevesebb $U_j^S = 1$ érték legyen. További megszorítás lehet, ha megelőzési feltételeket adunk meg. Ez a fogalom azt jelenti, hogy egy adott j munkát csak akkor lehet elkezdeni, ha a munkák valamely részét előtte befejeztük.

Többféle típusú gépeink lehetnek, ugyanis számuk és egymással való kapcsolatuk eltérő lehet. Tudjuk, hogy m számú gép áll rendelkezésre. Ha $m = 1$, akkor értelemszerűen egygépes ütemezési feladatról beszélünk, ha viszont egynél több, akkor több változat is ismert. Ha $m \geq 2$, és a gépek teljesen egyformák, akkor gépek párhuzamosak. Az uniform gépek tulajdonsága, hogy sebességeik eltérőek, ennek értékét minden gépnél s_i -vel jelöljük. Vagyis egy uniform gépen a j munka p_j/s_i ideig tart. Független gépek esetén definiálni kell a fentebb már említett p_{ij} értéket, mivel itt a megmunkálási idő minden gép-munka párra más lehet. „A shop modellben minden munka elvégzése *műveletekre* bomlik, melyek elvégzése egy-egy meghatározott gépen történik.”[2] A shop feladatoknak három alapvető fajtája van:

- Open shop: a műveletek elvégzésének sorrendje minden munka esetén tetszőleges.
- Job shop: minden munkához adott egy sorrend, mely meghatározza, hogy milyen sorrendben kell a műveletek végrehajtani.
- Flow shop: a job shop modelltől annyiban különbözik, hogy itt a műveleti sorrend minden munkára ugyan az.

Mindig a célfüggvény határozza meg, hogy az adott feladatnak mi a célja, milyen érték minimalizálásra, vagy maximalizálására kell törekedni. Mindig valamilyen paraméter szerinti optimális megoldás elérése a cél, de ha ez valami miatt nem lehetséges, akkor úgynevezett α -approximációs megoldást kell keresni. Az approximatív szó jelentése: „hozzávetőleges, tetszőleges.” [3] Az alfa jelentése, hogy a megoldás legfeljebb α -szor rosszabb az optimumnál. A feladatok célja sokféle lehet, például:

- Teljes átfutási idő (C_{max}^S érték) minimalizálása, hogy az ütemezés a lehető leggyorsabban a végéhez érjen.
- Átlagos átfutási idő, vagyis a $\frac{\sum_{j=1}^n C_j^S}{n}$ minimalizálása.
- Ritkábban előforduló eset, de lehetőség van a maximális késés, vagyis az $L_{max} = \max_j L_j^S$ érték minimalizálására.
- Ha a munkák súlyozottak, akkor a súlyozott átlagos átfutási idő $\sum_{j=1}^n w_j C_j^S$ értéke is minimalizálható.
- Utolsó példának pedig az elvégzett munkák súlyozott összegének esete, mint lehetséges célfüggvény. $\sum_j w_j U_j^S$ érték jelöli e feltételt, melyben ha egy adott munka nem késik, vagyis rá $U_j^S=0$, akkor nem emeli az összeg értékét. Nyilván minél kisebb a végső összeg, az annál kedvezőbb.

Minden egyes ütemezési problémát egy $\alpha | \beta | \gamma$ rendezett hármas jelöl. Az α jelöli a gépek számára és típusára vonatkozó megkötéseket. Egy gép esetén az első mezőben 1 áll, több gépes feladatoknál a következő jelölések fordulnak elő:

- P, vagyis párhuzamos gépek
- Q, vagyis uniform gépek
- R, vagyis független gépek
- O, vagyis Open shop feladat
- J, vagyis Job shop feladat
- F, mint Flow shop feladat

Az első mező tartalmazhat még pozitív egész számot, mely a gépek számát adja meg. A β mezője tartalmazza az egyéb feltételeket, ezek: $p_j=1$ (vagyis minden munka egységnyi időt vesz igénybe), $r_j, pmtn, prec$. Ezen kívül a γ rész pedig a célfüggvényt tartalmazza.

Például: $1 | r_j, pmtn | C_{max}$. Azt jelenti, hogy egy gép elérhető, megadott rendelkezésre állási idők vannak, a munkák megszakíthatóak, és a teljes átfutási idő minimalizálása a cél. Az egyes célfüggvények, változók jelölésére és magyarázatára az azt megkövetelő helyeken lesz szó.

2.1.1. Az ütemezés vállalati funkciója

Napjainkban a vállalkozásoknak nehéz dolga van, gondolok itt arra, hogy mennyire nehéz talpon maradni az egyre erősödő piaci versenyben, és főként mennyire nehéz virágzó és fejlődő céget igazgatni. Fontossá válik egy termelő vállalat életében az, hogy mennyire megbízható, mennyire tud reagálni az új, és hirtelen jövő piaci igények változásainak.

Ezekben a feladatokban van a cégek nagy segítségére egy vállalatirányítási rendszer. Egy jó rendszer végre tudja hajtani saját specifikus feladatait, valamint képes összegyűjteni az információt, és ezeket a megfelelő alrendszerek számára elérhetővé tenni, hogy az információáramlás zavartalanul bonyolódhasson le. Az integrált rendszer segítségével az adatfeldolgozási eszközök sokkal hatékonyabban és teljesebb mértékben használhatók ki.

Egy vállalatnak különböző problémák kombinációjával kell szembenéznük, ezek például:

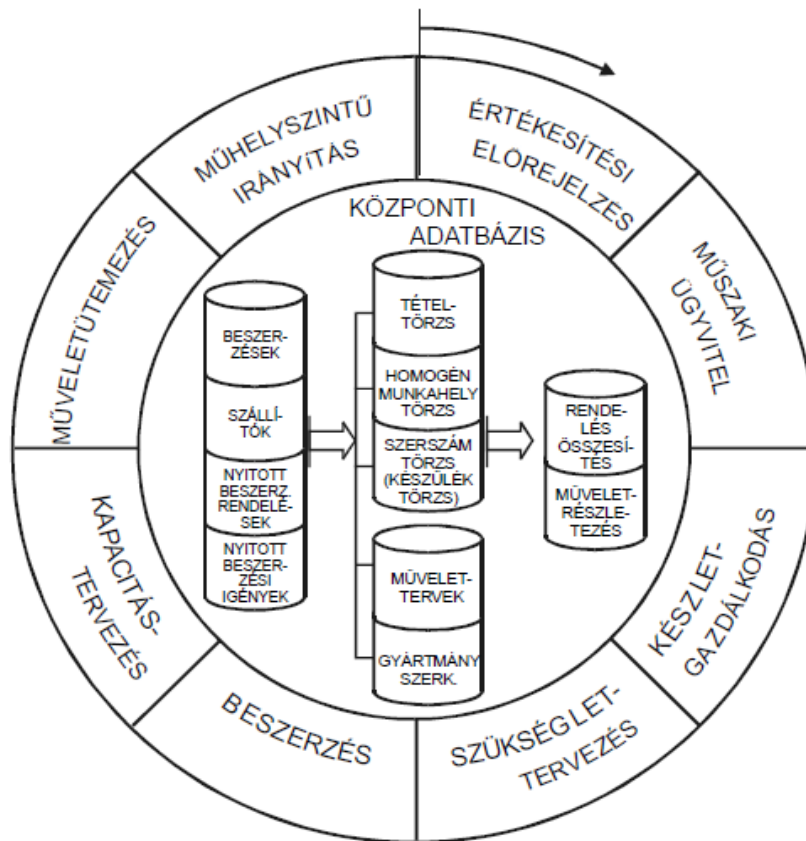
- Költségek mértéke (nyersanyag, berendezések költségei, stb.), fizetendő munkabérek;
- A jobb működés érdekében az erőforrásokkal (akár emberekkel) történő jobb gazdálkodás megtervezése, és fenntartása;
- A hirtelen változó piaccal való lépéstartás.

A termelés zavartalan, és hatékonyan működő folyamatához szükségünk van előzetes tervezésre, majd a megfelelő szintű kivitelezésre. A tervezés szakaszában fontos, hogy az elfogadott tervvariáció megfelelő mértékű nyereséget hozzon, tehát semmiképpen se legyen veszteséges, erőforrásokat tekintve reális fedezete van, valamint lehetőség szerint tükrözze a vállalat hosszú távú stratégiáját, fejlesztési céljait. Továbbá fontos, hogy figyelembe vegye a felmerülő kapacitás-korlátokat. A terv kivitelezésében pedig a művelettervezés és a gyártás ütemezésére fog nagy szerep hárulni.

2.1.2. Vállalatirányítási rendszer modell

Most pedig nézzük meg, hogy hogyan néz ki egy vállalatirányítási rendszer modellje. A következő ábra egy jól szervezetten felépített modellt mutat:

Amint látható, egy központi adatbázis van a rendszer magjában, valamint az egyes tevékenységek nyolc különböző alrendszerbe vannak integrálva. Mi a műveletütemezés szemszögéből fogjuk megvizsgálni az összes többit, mivel szervesen kapcsolatban állnak egymással.



1. ábra

Egyszerűsített számítógépes TIR modell

Forrás: Dr. Tóth Tibor: Számítógépes vállalatirányítás (saját fejlesztésű tananyag), Miskolci Egyetem Gépészmérnöki Kar, Miskolc 2002, 28. oldal

Egy termelő vállalatnál termelésütemezés egyik legfontosabb feladata, hogy egy finomprogramozást végezzen, amely megadja a gyártó berendezéseknek, hogy milyen munkákon kell dolgozniuk, milyen sorrendben, milyen megszakíthatóságokkal, milyen időintervallumokkal, valamint figyel a különböző erőforrások elérhetőségére. Nagyon fontos számára az öt megelőző kapacitástervezés és készletgazdálkodás, mivel a gépek, a munkaerő és az erőforrások nem végtelenek, és a gyártókapacitás is korlátozott. Az ütemezési folyamatot ezen kívül megelőzi a szükséglettervezés, mely az előrejelzések és a konkrét megrendelések alapján igyekszik a beszerzési alrendszeren keresztül biztosítani a konkrét gyártáshoz szükséges mennyiségeket. A műszaki ügyvitel feladata a vállalat műszaki-technológia alapadatainak számítógépes nyilvántartása és kezelése, pl. darabjegyzékek, művelettervek tárolása. A darabjegyzék tartalmazza ugyanis a végtermék egy darabjának elkészítéséhez szükséges összes részegység, alkatrész, alapanyag, stb. típusát, és az egyes típusból szükséges mennyiséget. A művelettervek pedig nevükből adódóan tartalmazznak minden olyan folyamatot, műveletet,

időpontot, határidőt, stb. mely alapján a gyártás konkrét ütemezése megindulhat, és zavartalanul folyhat.

2.2. Determinisztikus modellek

„Egy modellt determinisztikusnak nevezünk, ha a benne szereplő minden konstans és változó előre meghatározott, nem függ a véletlentől.”[4] A determinisztikus szó jelentése: előre meghatározott. Ennél modellenél a feladatok beérkezési ideje és megmunkálási ideje ismert. Ebben a részben az egygépes, párhuzamos gépes, uniform gépes, független gépes, valamint az Open-, Job-, és Flow shop feladatokról lesz szó. Minden egyes típusnál bemutatásra kerülnek a fontosabb algoritmusok, valamint a használatuk helye.

2.2.1. Egygépes ütemezés

Mint a nevében is benne van, egyetlen egy gép áll rendelkezésre, amin különböző célfüggvény figyelembe vételével kell megadni a lehető legjobb ütemezést, esetlegesen a különböző feltételek figyelembe vételével. Mivel rengeteg feladat létezik még elméleti szinten is, amelyek egy gépre vonatkoznak, így célszerű inkább a különböző algoritmusokat megvizsgálni, melyekre később hivatkozni lehet. A különböző eljárások végszávaül szerepelni fog egy-két feladat, mint példa, mely azokkal megoldható:

SPT sorrend, mely a shortest processing time rövidítése. Mivel egygépes feladatokról van szó, ezért itt még csak p_j érték van definiálva a megmunkálási időkre, ami értelem szerűen a j -edik munka gépen töltött idejét jelenti. Az algoritmus lényege, hogy a munkákat megmunkálási idő szerint nem-csökkenő sorba kell a gépéhez hozzárendelni ($p_1 \leq p_2 \leq \dots \leq p_n$). Ez a módszer optimális megoldást ad például az $1 \mid \mid \sum C_j$ feladatra, amikor az átlagos átfutási idő minimalizálása a cél.

EDD sorrend: a munkákat a d_j határidők szerint nem csökkenő sorban végzi el a gép ($d_1 \leq d_2 \leq \dots \leq d_n$). Optimális megoldást ad, ha a feladat a maximális késés minimalizálás, és akkor, ha a feladat ugyan ez, csak teljesül a megszakíthatósági feltétel, és vannak rendelkezésre állási idők is megadva (vagyis az $1 \mid \mid \sum L_{max}$ és $1 \mid r_j, pmtn \mid \sum L_{max}$ problémákra).

SPRT sorrend: mindig az a rendelkezésre álló munka következik, melynek a megmaradó megmunkálási ideje a legkisebb. Optimális megoldást ad arra az esetre, amikor egy gép dolgozik, adottak rendelkezésre állási idők, megszakíthatóak a munkák, és a befejezési idők minimalizálása a cél.

CPS algoritmus: az első approximációs algoritmus, mely 2-approximációs a $1 \mid r_j \mid \sum C_j$ feladatra, vagyis az optimális megoldásnál maximum kétszer rosszabb eredmény ad. Lépései:

1. „Az $1 \mid r_j \mid \sum C_j$ feladat inputját használva az $1 \mid r_j, pmtn \mid \sum C_j$ –re adunk egy S ütemezést.
2. Rakjuk sorba a munkákat az S -beli befejezési idők szerint növekvő sorrendbe, indexelés: $C_1^S < C_2^S < \dots < C_n^S$.
3. Ütemezzük a munkákat j_1, j_2, \dots, j_n sorrendbe megszakítások és szükségtelen állásidők nélkül, figyelembe véve a rendelkezésre állási időket”[5]

Hodgson algoritmus:

1. Először rendezzük a munkákat a d_j határidők szerint nem-csökkenő sorrendbe.
2. Vegyünk egy \mathcal{A} halmazt, melyben az időre elkészülő munkák lesznek, és vegyünk egy $p = 0$ értéket, melyhez folyamatosan hozzáadjuk az időre elkészülő munkák megmunkálási idejét.
3. A soron következő munkát rakjuk bele az \mathcal{A} halmazba, megmunkálási idejét pedig adjuk hozzá a p értékéhez. Ha az így kapott p érték kisebb vagy egyenlő, mint a munka határideje, a következő munkára térünk. Ha a határidő a nagyobb, akkor az \mathcal{A} halmazból kivesszük azt a munkát, melynek a megmunkálási ideje maximális, a késők közé tesszük és az egyre növekvő p értékből is levonjuk ezen munka megmunkálási idejét.
4. Ha minden munkára kész, akkor az \mathcal{A} -beli munkákat ütemezzük EDD sorrendben, a többit tetszőlegesen.
5. Optimális megoldást ad az algoritmus az $1 \mid \mid \sum U_j$ feladatra (ahol a késő munkák darabszámát szeretnénk minimalizálni).

2.2.2. Párhuzamos gépes ütemezés

A gépek párhuzamosak, ha legalább két gép van, melyek teljesen ugyan olyanok. Egy munka egy időben legfeljebb egy gépen futhat, és egy gép legfeljebb egy munkán dolgozhat.

$AP \mid C_{max}$ feladat

A feladat a teljes átfutási idő minimalizálása. Ehhez felírható két korlátozás a C_{max}^* értékre, mely az optimális átfutási időt jelöli. Ezek a következők:

$$C_{max}^* \geq \frac{(\sum_1^n p_j)}{m} \text{ és } C_{max}^* \geq \max\{p_j\}$$

Vagyis az optimális teljes átfutási idő semmiképpen nem lehet kisebb, mint az az időtartam, amit a gépek átlagosan dolgoznak, és mint a leghosszabb megmunkálási idő. A lehetséges megoldásokról később lesz szó.

Graham listás ütemezése:

1. Rögzítsük a munkák sorrendjét tetszőlegesen.
2. Ütemezzük am munkákat, hogy amint egy gép elérhetővé válik, kezdje meg az általa megadott sorrendben lévő következő munkát, amelyik még egyik géphez sincs hozzárendelve. Ha nincs ilyen, akkor feltehetőleg nincs több munkája az adott gépnek.

Ebből kifolyólag nem lesz fölösleges állásidő, hiszen egy gép csak akkor áll, ha már nincs több munka, mely számára elérhető. A listás ütemezéssel többfajta feladatot is meg lehet oldani, noha csak approximációs a megoldás. Íme a feladatok, approximációs faktoriall együtt:

- $2 - (\frac{1}{m})$ -approximációs algoritmus a $P \mid \mid C_{max}$ feladatra.
- 2-approximációs algoritmus $P \mid r_j \mid C_{max}$ feladatra.
- 2-approximációs a $P \mid prec \mid C_{max}$ feladatra.

$Prec$ = megelőzési feltételek adottak. Emiatt fontos tudni, hogy mikor nevezünk egy munkát elérhetőnek: akkor nevezzük a j munkát elérhetőnek a t időpillanatban, ha minden olyan k munka kész, mely a j -nek megelőzési feltétele.

LPT sorrend: a munkákat nem-növekvő sorba kell rakni a p_j értékek szerint, tehát a hosszúakkal kezdődik a megmunkálás. Ezzel egy úgynevezett π sorrend határozható meg, és a listás ütemezés eszerint a sorrend szerint fogja a munkákat ütemezni. Ez a módszer $\frac{4}{3}$ -approximációs algoritmus a $P \mid \mid C_{max}$ feladatra.

McNaughton algoritmus:

A módszerrel a $P \mid pmtn \mid C_{max}$ feladat oldható meg. A $pmtn$ megszakíthatóságot jelent. „Párhuzamos gépeknél a **megszakíthatóság azt jelenti**, hogy megszakíthatjuk, majd egy későbbi időpontban, esetleg másik gépen folytathatjuk.”[6] Bevezethető egy úgynevezett D értéket, mely alsó korlát a feladat optimumára. Értéke:

$$D = \max\left\{\frac{\sum_1^n p_j}{m}; \max_j(p_j)\right\}$$

Vagyis a két mennyiség közül a nagyobbat veszi fel értékének. Az algoritmus ez alatt a D idő alatt fogja elvégezni az összes munkát, mivel itt gépenként kell haladni az ütemezéssel.

Lépések:

1. A munkákat tetszőleges sorrendbe rakjuk.
2. A gépeket feltöltjük a D értékig munkákkal úgy, hogy a soron következő munka még el ne végzett részét adjuk hozzá az adott géphez, amíg annak futási ideje el nem éri a D értéket. Ha elérte, a következő géppel tesszük meg ugyanezt.

2.2.3. Ütemezés független gépeken

Ebben a fejezetben egy gráfelméleti megoldás következik, mivel bonyolult munkák esetén nagyon sokat segíthet egy ügyesen felrajzolt, és különböző értékekkel áttekinthetően ellátott gráf.

Először is, a független gépek lényege, hogy a p_{ij} érték bevezetésre kerül, ami az i -edik gépen a j -edik munka megmunkálási idejét jelenti. Ez azért lényeges, mert a gépek már nem teljesen ugyan olyanok.

Inputok: n darab munka, m darab gép, p_{ij} megmunkálási idő. Az $R \mid \sum C_j$ feladat szolgál példaként.

Legyen egy $a_{i,k}$ érték, ami azt jelenti, hogy az adott munka az i -edik gépen hátulról a k -adik. Ha a j munka hátulról a k -adik az i -edik gépen, akkor a $\sum C_j$ összeget $k \cdot p_{ik}$ -vel fogja növelni. Elkészíthető a $G = (A, B, E)$ gráf, legyenek c kapacitások az élekhez hozzárendelve. Az A osztály tartalmazzon minden munkának megfelelően egy pontot, a B osztály pedig minden $a_{i,k}$ értéknek megfelelően egy pontot ($v_j \in A$ és $w_{ik} \in B$; ezekkel a jelölésekkel). Az E élhalmazban pedig legyen benne az összes (v_j, w_{ik}) pont, a hozzá tartozó $c(v_j, w_{ik}) = k \cdot p_{ij}$ élkapacitással. Így az optimális megoldás megfelel az A osztályt fedő minimális súlyú párosítások halmaza.

Definíció: „Adott $G = (V, E)$ gráfban az $M \subseteq E$ élhalmazt egy párosításnak nevezünk, ha nem tartalmaz hurokért, és két különböző M -beli élnek nincsen közös csúcsa. Az M -beli éleket független éleknek, és az M -beli élek csúcsait M -fedett csúcsoknak nevezzük”[7]

2.2.4. Ütemezés uniform gépeken

A gépek jellegzetessége, hogy s_i -vel jelölt sebességük eltérő. Ebben a fejezetben két fontos algoritmus kerül bemutatásra.

Gonzalez algoritmus:

1. Tegyük a munkákat SPT sorrendbe (tehát $p_1 \leq p_2 \leq \dots \leq p_n$)
2. Az SPT sorrendben haladva ütemezzük a munkákat úgy, hogy a soron következő munka befejezési ideje a lehető legkisebb legyen. Tehát azt használjuk ki, hogy a leggyorsabb gépekre rakjuk először a legrövidebb munkákat, és amint egy gyorsabb gép elérhetővé válik, az aktuális legrövidebb munka átkerül arra.

Optimális megoldást kapunk vele a $Q | pmtn | C_{max}$ feladatra.

SCT algoritmus:

1. Rakjuk a munkákat LPT sorrendbe (tehát $p_1 \geq p_2 \geq \dots \geq p_n$)
2. LPT sorrendben haladva a soron következő munkát mindig úgy ütemezzük, hogy a befejezési idő minimális legyen (tehát minél előbb végezzen az adott munkával, valamint a korábbi részütemezést változatlanul hagyjuk). Ha ez több gépen is lehetséges, akkor a legkisebb indexűn hajtsuk végre.
3. Előnye: egyik gépen sem lesz állásidő
4. $(2 - \frac{1}{m})$ – approximációs algoritmus $Q | | C_{max}$ –ra.

2.2.5. Az open shop feladat

Az open shop modellben n darab munkát kell elvégezni el m darab gépen úgy, hogy a munkák a gépeket tetszőleges sorrendben járhatják végig, és minden munkát minden gépen csak egyszer kell megmunkálni. Egy munka most is egyszerre egy gépen futhat, azaz ugyanannak a munkának két részfeladata egyszerre nem futhat két gépen. A cél ezúttal a teljes átfutási idő minimalizálása lesz. Felírható két korlátot a C_{max}^* optimális értékre. Ezek:

$$J_{max} = \max_{1 \leq j \leq n} \sum_{i=1}^m p_{ij} \text{ valamint } M_{max} = \max_{1 \leq i \leq m} \sum_{j=1}^n p_{ij}$$

$$\text{Legyen: } C_{max}^* \geq J_{max}; M_{max}$$

Tehát a J_{max} érték annak a munkának az össz megmunkálási idejét jelenti, amelyik a legtöbb időt vesz igénybe. Ebből kifolyólag az M_{max} pedig annak a gépnek a „működési idejét”, ami a legtovább dolgozik. Ezek után kimondható az optimális teljes átfutási időre, hogy legalább akkora, mint ez a két érték.

O | $p_j \in \{0,1\}$ | C_{max}

A feladat tehát a teljes átfutási idő minimalizálása, úgy, hogy a megmunkálási idők az 1, vagy a 0 értéket veszik fel. Most is érdemes gráfelméleti módszert alkalmazni. Szükség van a $G = (M, J, P)$ gráfra, ahol M osztályban minden géphez tartozik egy pont, és a J osztályban minden munkához egy pont. Az i és j pontok között akkor megy él, ha $p_{ij} > 0$. Ennek az élhalmazát kell minimális színű éllel kiszínezni úgy, hogy azonos színű éleknek ne legyen közös pontja. Így az azonos színű élek halmazai a keresett párosításaink. A G páros gráf $\chi_c(G)$ elszínezési száma a ΔG maximális fokszámával.

Definíció: „Egy G gráfot páros gráfnak nevezünk, ha a G pontjainak $V(G)$ halmaza két részre, egy \mathcal{A} és \mathcal{B} halmazra osztható úgy, hogy G minden élének egyik végpontja \mathcal{A} -ban, másik végpontja \mathcal{B} -ben van.”[8]

O | | C_{max}

Ha nincsen megszabva, hogy 0 vagy 1 a megmunkálási idő, akkor a helyzet kicsit megváltozik. Ennek a megválaszolásához a sűrű ütemezés fogalmára van szükség. Egy ütemezés akkor sűrű, ha nincs olyan időpillanat, amikor egy gép áll, holott dolgozhatna egy számára elérhető részmunkán. Az elérhetőség azt jelenti, hogy a (rész)munka nem fut éppen másik gépen, és még a szóban forgó gépen sem futott. Minden algoritmus, ami sűrű ütemezést ad, 2-approximációs erre a feladatra.

2.2.6. A flow shop feladat

A flow shop feladatban minden munkát meg kell munkálni minden gépen, ugyanabban a sorrendben. Az alapgondolat a munkák sorrendjének hatékony megválasztása.

Permutáció típusú ütemezés: A munkák sorrendje minden gépen ugyan az.

F2 | | C_{max}

Ebben a példában két gép áll rendelkezésre, és a C_{max} értéket kell minimalizálni. Bejön egy módosított jelölés (a_j, b_j) ami a j munka első és második gépen töltött idejét jelenti. Jó gondolat lehet gyorsan megcsinálni azokat a munkákat, amik az első gépen kevés időt töltenek, hogy a második gép minél hamarabb munkába állhasson. A második gépen viszont azokkal jó kezdeni, amik ott sok időt fognak eltölteni.

Johnson sorrendje: Két halmazra lesz szükség, egy \mathcal{A} és egy \mathcal{B} halmazra. Az \mathcal{A} halmazban vannak azok a munkák, melyeknek az első gépen töltött ideje kisebb vagy egyenlő, mint amennyit a második gépen fog eltölteni. \mathcal{B} halmazban pedig azok, ahol az első gépen eltöltött idő mindig több, mint a második gépen. A sorrendben pedig a munkák úgy követik egymást, hogy p_{ij} szerint növekvő sorban először azok a munkák mennek az első gépen, melyek az \mathcal{A} halmazban vannak. Aztán azok a munkák jönnek, amelyek a \mathcal{B} halmazban találhatóak, p_{ij} szerint csökkenő sorrendben. Johnson sorrendje optimális erre a feladatra.

2.2.7. A job shop feladat

Adott n darab munka és m darab gép, valamint ugyan úgy a p_{ij} megmunkálási idők. A sorrend, ahogy a munkák a gépeket végigjárják, munkánként külön-külön adottak. Adott egy úgynevezett π_j permutáció is, minden munkához, ez jelenti a gépek sorrendjét.

J2 | | C_{max}

Visszavezethető az F2 | | C_{max} feladatra, mivel annak az általánosítása. **Johnson algoritmusával** megkapható az ütemezés, de a munkákat két halmazra kell bontani. Az egyik halmaz, ahol az első gépen kezdődik a munka, majd a másodikra megy (ez a \mathcal{B}_{12} halmaz), a másik halmazban pedig fordítva (ez a \mathcal{B}_{21} halmaz). Az optimális ütemezés kulcsa: az M_1 gépre kell vezényelni a \mathcal{B}_{12} halmaz első gépre eső részét, majd a \mathcal{B}_{21} halmaz szintén első gépre eső részét, ilyen sorrendben. Az M_2 gépen állásidő nélkül indulhat a \mathcal{B}_{21} halmaz második gépre eső része, majd jöhetnek a \mathcal{B}_{12} halmaz második gépre kiosztott részei. Tehát $\mathcal{B}_{12}(1) \rightarrow \mathcal{B}_{21}(1) \rightarrow \mathcal{B}_{21}(2) \rightarrow \mathcal{B}_{12}(2)$.

$$\mathcal{B}_{12} = \{j \in J : \pi_j = (1, 2)\},$$

$$\mathcal{B}_{21} = \{j \in J : \pi_j = (2, 1)\}$$

J | n = 2 | C_{max}

Fordított a helyzet, most két munkát, j_1 és j_2 -t kell ütemezni m gépen, valamint adott a p_{ij} megmunkálási idő, és π_1, π_2 sorrendek. Felvehető egy kétdimenziós koordináta rendszerben egy téglalap, ahol két szemben lévő csúcspont a $O(0,0)$ és az $F=(\sum_i p_{i1}, \sum_i p_{i2})$. A vízszintes tengely felosztható p_{i1} intervallumokra π_1 sorrend szerint, a függőleges pedig p_{i2} -re π_2 szerint. Tehát olyan hosszú egy intervallum, amennyit az adott gépen tölt az egyes vagy kettes munka. Az intervallum párokat, ahol a két gép áll párban egymással, tiltott zónának nevezzük. A téglalapon belül kell felvennünk egy görbét, amely vízszintes, függőleges, és 45 fokos

meredekségű vonalakkból áll. Megengedett lesz az ütemezés, ha a görbe nem érinti egyik tiltott mezőt sem, ha semelyik I_{i1} intervallum belső pontja fölött nem halad függőleges vonal, valamint ha semelyik I_{i2} intervallum belső pontjával egyvonalban nem halad vízszintes vonal. Az átfutási idő a görbe hossza lesz (meredek szakaszok hossza $=1/\sqrt{2}$).

2.3. Sztochasztikus modellek

„Egy modellt sztochasztikusnak nevezünk, ha a benne szereplő adatok vagy azoknak egy részhalmaza véletlentől függő mennyiség.”[9] A gazdasági számítások gyakran tartalmazzak a sztochasztikára épülő számításokat, mivel ott sok adat nagyban függ a véletlentől. A legfontosabb különbség a determinisztikus modellel működő rendszerekhez képest az, hogy a valószínűségeken alapuló egy adott feladatra mindig más megoldást ad, míg a determinisztikus mindig ugyan azt. Ezen modellekben a következő változók lesznek a véletlentől függőek, M. L. Pinedo jelöléseivel megadva:

- X_{ij} : az i -edik gépen lévő j -edik munka megmunkálási ideje.
- $1/\lambda_{ij}$: az X_{ij} várható értéke.
- R_j : a j -edik munka rendelkezésre állási ideje.
- D_j : a j -edik munka határideje.
- w_j : a j -edik munka súlya.”[10]

Szembetűnő, hogy a megmunkálási idő X_{ij} jelölést kapott, ez annak tudható be, hogy a p -re máshol lesz szükség, mivel az most valószínűséget fog jelölni. A w_j érték viszont változatlanul ugyan azt jelenti, mint a determinisztikus modelleknél.

2.3.1. Eloszlások és algoritmusok

Mostantól szükség lesz valószínűségi eloszlások és sűrűségfüggvények használatára, melyeknek igen sok formáját lenne lehetőség megvizsgálni. Nem lesz szó mindegyikről, ezekből csupán ízelítőt tartalmaz a dolgozat.

Egy folytonos eloszlású véletlen változó felvehet bármilyen nemnegatív valós számot egy intervallumon belül. A folytonos eloszlásfüggvényt $F(t)$ - vel, sűrűségfüggvényét pedig $f(t)$ - vel jelöljük, ahol a t egy időpillanatot jelöl. Vagyis:

$$F(t) = P(X \leq t) = \int_0^t f(t)dt, \text{ ahol}$$

$$f(t) = \frac{dF(t)}{dt} \text{ és } \bar{F}(t) = 1 - F(t) = P(X \geq t).$$

Az sztochasztikus ütemezési modelleknél fontos szerepet kap az exponenciális eloszlás, melynek eloszlás és sűrűségfüggvénye:

$$F(t) = 1 - e^{-\lambda t}$$

$$f(t) = \lambda e^{-\lambda t}$$

Az eloszlás függvény megmutatja, hogy mennyi annak a valószínűsége, hogy az X véletlen változó értéke kisebb, mint a t érték.

A következő fontos eloszlás, melyet ismernünk kell, a geometriai eloszlás, mely esetében annak a valószínűsége, hogy az X felveszi a t értéket (ha $t = 0, 1, 2, \dots$) a következő képlettel adható meg:

$$P(X = t) = (1-q)q^t$$

Eloszlásfüggvénye, vagyis annak a valószínűsége, hogy az X érték kisebb mint a t érték:

$$P(X \leq t) = \sum_{s=0}^t (1-q)q^s = 1 - \sum_{s=t+1}^{\infty} (1-q)q^s = 1 - q^{t+1}$$

Várható értéke pedig:

$$E(X) = \frac{q}{1-q}$$

Ütemezésméleti alkalmazásoknál bevezethető egy új fogalom, az elkészülési arány, melynek jele $c(t)$, kiszámítása módja pedig:

$$c(t) = \frac{f(t)}{1 - F(t)}$$

Fontos megjegyzés, hogy exponenciális eloszlású véletlen változó esetén a $c(t) = \lambda$ minden t esetén.

Ezután a rövid valószínűségi számítási áttekintő után említés szintjén következzen két algoritmust, mely már a sztochasztikus modellek témakörébe tartozik. Mivel a sztochasztikus feladatok témaköre sok száz oldalra rúg, ezért a konkrétumokról a későbbi fejezetekben lesz szó, és csak azon részek érintésével, melyet a téma megkövetel. Tehát a példaalgoritmusok:

WSEPT sorrend: minimalizálja a várható értékek összegét a súlyozott befejezési idő, egygépes problémák esetén, ha nem teljesül a megszakíthatósági feltétel. Ez úgy érhető el, hogy a munkákat csökkenő sorrendbe rakjuk a $w_j / E(X_j)$ szerint. Ez ekvivalens azzal, hogyha a $\lambda_j w_j$ szerint rendezzük csökkenő sorrendbe a munkákat. Ez az algoritmus más feladatok esetén is hatékony megoldást adhat.

LEPT sorrend: ha a munkákat várható befejezési idők szerint csökkenő sorrendbe ütemezzük, akkor minimalizálhatjuk a várható teljes átfutási időt egygépes feladat esetén, ha nem teljesen a megszakíthatósági feltétel. Jó megoldást adhat egyéb problémák esetén is.

3. Az Anton Kft. szerszámgyártási folyamata

Ebben a fejezetben kerül sor az Anton Kft. bemutatására, valamint a szerszámgyártási tevékenység ismertetésére. A cél a vállalat megismerése, hogy pontos kép legyen a munkájáról, valamint, hogy a szerszámgyártási tevékenységgel kapcsolatban minden szükséges információ rendelkezésre álljon ütemezési feladat végrehajtásához.

3.1. A vállalat bemutatása

Az Anton Kft. profilját tekintve szerszámgyártással, műanyag alkatrészek gyártásával és speciális fémmegmunkálással foglalkozó, zalaegerszegi telephelyű magyar tulajdonban lévő vállalat.

A cég mint egyszemélyes magánvállalkozás jött létre 1990-ben, melynek fő tevékenysége kisebb szerszámok készítése és javítása volt. A cég 1992-ben alakult Kft.-vé, majd az ezt követő évben üzembe helyezte az első fröccsöntő gépet. Első gyártócsarnokát, melynek mérete 2200m² 1999-ben építi meg, jelenlegi, zalaegerszegi telephelyén. 2002-ben bővült a cég a speciális megmunkálási üzletággal, melynek feladata nikkel bázisú alapanyagokból szerszámok gyártása az energiaipar részére. Az újabb mérföldkönek tekinthető fejlődés 2004-ben következett be, amikor a műanyaggyártás is önálló iparággá fejlődött. E célból épült meg a második 4800m²-es gyártócsarnok, ahol több mint 1500 tonna műszaki műanyagot dolgoznak fel fröccsöntő gépekkel. 2008-ban tovább növekedett a gyártóterület 12000m²-re. A cég üzletágai:

- Szerszámgyártási tevékenység. A dolgozat szempontjából a legfontosabb üzletág, így ezzel a 3.3 fejezetben foglalkozunk.
- Műanyag fröccsöntés: az Anton Kft. 1993 óta folytat fröccsöntő tevékenységet, a műanyaggyártás pedig 2004-ben alakult önálló üzletággá, főként az elektromos-, háztartási- és autóiipari szegmensnek szállít. A létesítmény műanyaggyártással foglalkozó része képes automatizált termelésre az alapanyag ellátástól a termék csomagolásáig. Az üzletág gyártáshelyes terméktervezési tevékenységet is folytat.
- Speciális megmunkálás: az SMD üzletág 2004-ben alapult, céljának a nemzetközi vevőkör kiszolgálását tűzte ki. Az üzletág fontos jellemzője, hogy a berendezések egyedi fejlesztésűek, és a termékekhez egyedileg tervezett technológiákkal történik a turbina alkatrészek speciális megmunkálása. A folyamatos fejlesztésekre és az egyedi megoldások használatára főként azért van szükség, mert az alkatrészek speciális anyagúak,

és bonyolult geometriával rendelkeznek. Évente több mint 12000 alkatrész megmunkálásra kerül itt sor.

Fontos megjegyezni, hogy a vállalat korszerű, integrált vállalatirányítási (ERP) rendszert üzemeltet. Minőségbiztosítási szabványai a következők:

- MSZ EN ISO9001:2009: műanyag fröccsöntő szerszámok és egyéb gépipar eszközök tervezésére, gyártására vonatkozó szabvány, tehát a szerszámgyártás tanúsítványa.
- MSZ ISO/TS 16949:2010: a műanyag alkatrészek gyártásának minőségét biztosító szabvány.
- AS 9100 Rev C: a speciális megmunkálásra vonatkozó minőségbiztosítási szabvány, a repülőgépiparban is használt szabvány.
- MSZ EN ISO14001:2005: a teljes megmunkálására vonatkozó előírás.

3.2. Termelésütemezés az Anton Kft. üzemeiben

	Termelés típus	Jelenlegi ütemező eljárás
Szerszámgyártás	<i>Egyedi gyártás</i>	<i>Kézi</i>
Műanyag fröccsöntés	<i>Nagy széria</i>	<i>Kézi</i>
Speciális megmunkálás	<i>Kötegetelt gyártás</i>	<i>Szoftver/kézi</i>

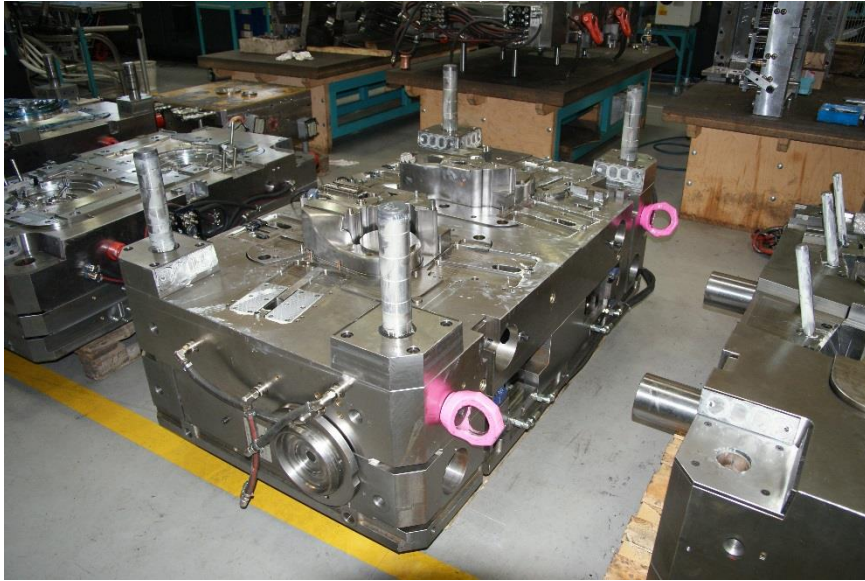
1. táblázat

A jelenlegi ütemező módszerek az egyes üzemekben

A táblázat adataiból látszik, hogy a szerszámgyártási tevékenység ütemezése kézi módszerrel történik, a célkitűzés ezen változtatni, és egy korszerű, szoftveres ütemező rendszert bevezetni.

3.3. Szerszámgyártási tevékenység

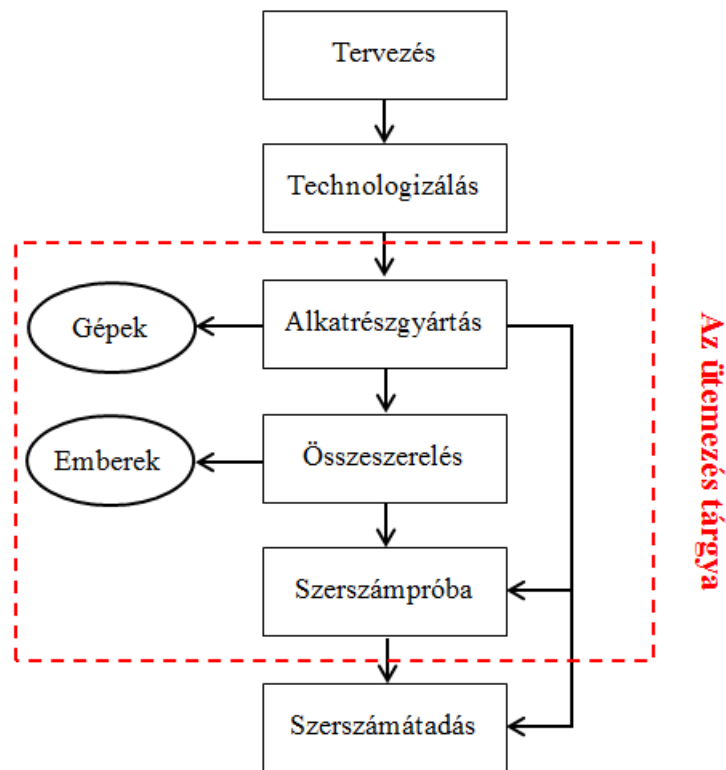
Az Anton Kft. 1990-ben kezdte tevékenységét szerszámgyártással, ebből lett 1999-re egy 900m²-es gyártóterület a hozzá szükséges tervező csapattal együtt. 2002 óta gyárt fém alkatrészek megmunkálásához szükséges készülékeket. Jelenleg évente több mint 150db fröccsöntő szerszám készül el évente. Fontos megjegyezni, hogy a gyártási tevékenység a globális nagyvállalatok igényeinek kiszolgálására lett felépítve.



2. ábra
Fröccsöntő szerszám
Forrás: Anton Kft.



3. ábra
Az Anton Kft. szerszám összeszerelő műhelye
Forrás: Anton Kft.



4. ábra

A szerszámgyártási tevékenység folyamatábrája az Anton Kft-nél
 Forrás: saját szerkesztés

A szerszámgyártási tevékenység első lépése a tervezése, ahol egy új szerszám születik, valamint elkészül a darabjegyzék is. A technológizálás szakaszában születnek meg a művelettervek. Az Anton Kft-nél egyedi gyártási folyamat történik, mely azt jelenti, hogy minden egyes rendeléshez egyedülálló szerszámot készítenek, mely csak az adott gyártást szolgálja ki.

3.4. Egyedi gyártási folyamat sajátosságai

A cél az egyedi gyártási folyamat üzemzése a job- és az open shop modellből kialakított hibrid modell segítségével, a szerszámgyártási üzletágra. Ehhez először is tudni kell ezen egyedi folyamat sajátosságait.

3.4.1. Egyediség

Az egyedi gyártás a szériagyártástól sok dologban különbözik. Az alapvető különbség az, hogy itt egy gyártott termékből csak egy, vagy nagyon kis darabszám (2-8 db) készül, általában nincs ismétlődés. Tehát a szériagyártástól eltérően kiegyensúlyozó puffer mennyiségekkel nem számolhatunk a gyártásban.

3.4.2. Projekt prioritások











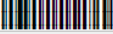

A gyártási tevékenységek projektek formájában vannak jelen, melyeknek prioritásai vannak. A fontossági sorrendek, prioritások meghatározása a vezetőség feladata. Az egy projekthez tartozó alkatrészeket együtt kell gyártani, közel egy időben. Előfordulhat, hogy a termelési folyamatba befér más munka is, de törekedni kell arra, hogy ezen alkatrészek együtt mozogjanak. A projektek mindig teljesen egyediek.

3.4.3. Darabjegyzék, művelettervek

A darabjegyzék meghatározza, hogy egy szerszám elkészüléséhez milyen további beépülő alkatrészekre van szükség. A műveletterv pedig meghatározza, hogy az adott alkatrészen milyen sorrendben és milyen megmunkálásokat kell elvégezni. Az egyedi gyártás esetén a darabjegyzékek, művelettervek folyamatosan készülnek, előbbi 2-6 hét alatt, utóbbi ezután 1-2 hét alatt lesz kész. Ebből kifolyólag a gyártás már azelőtt elindul, mielőtt a teljes tervdokumentum rendelkezésre állna.

3.4.4. Műveletterv alternatívák

Lehetőség van alternatív művelettervek illetve gépcsoportok hozzárendelésére. A gyártás egyes szakaszaiban lehetőség van egy adott megmunkálást több különböző technológiával (géptípussal) is végrehajtani. A terv mindig úgy születik, hogy optimális, költséghatékony legyen, azonban aki tervezi az ütemezést, nem mindig tudja, hogy mikor lesz pontosan beütemezve a munka, ugyanis előfordulhat az, hogy akkor szűk keresztmetszet lesz az a gépcsoport, amelyre az adott műveletet tervezi. Például itt jön képbe a műveletterv alternatíva, mellyel az ilyen helyzetek hatékonyan megoldhatók.

ANION		TECHNOLÓGIAI UTASÍTÁS					2014.10.06. 10:45			
Gyártási az.: 1097023		Rajzszám : sA1097/023	Megnevezés : Álló Betét 03		Mennyiség: 5					
Gyártási rend.: s1097	Kibocsátás: 2014.06.13.	Esedékesség: 2014.06.13.		Technológus: XXXXXXXXXX						
Technológiai előírás										
	Műv.	Gép.	Státusz	Me.	Norma/db	Norma	Ráf. idő	Dátum	Aláírás *	Megjegyzés
	Darabolás ▶ Műv.:1010 Gép:SG037		Ráhagyással darabol 5x	Lezárt	0	0,5ó	0,0ó	0,8ó		
	Esztorgálás ▶ Műv.:1030 Gép:SG007		Esztorgál illesztett átmérő és hossz méreteket 0,3oldi kösz ráhagyással. Vízhorony, és °O gyúró hornyokat készre.	Lezárt	0	0,5ó	0,0ó	5,5ó		
	Marás ▶ Műv.:1040 Gép:SG009		Vízterelő hornyot CNC nagyolás után hőkezelés előtt lapolással párhuzamosan rajzi méretre mar!		0	1,0ó	0,0ó	0,0ó		
	NC Marás ▶ Műv.:1060 Gép:SG003		Alulnézetben metszetben méretezett, jelölt fúvóka férőhelyet készre, lapolást 0,5 mm/oldal simítási ráhagyással mar. Fazon oldalon lapolás figyelembe vételével 3d fazont nagyol prg.1097-023-R00.	Lezárt	0	1,5ó	0,0ó	15,0ó		
	Mérőgépes mérés ▶ Műv.:1076 Gép:SG020		Betét illeszkedő méreteirő mérési jkv. készit.		0	0,0ó	0,0ó	0,0ó		
	Hőkezelés ▶ Műv.:1080 Gép:SG106		Kooperáció! Vákumezés és 2x megeresztés! Anyag 1.2343 ESU 50+2 HRC Bizonylatolni!		0	0,1ó	0,0ó	0,0ó		
	Síkköszörű ▶ Műv.:1090 Gép:SG023		Betétmagasságot beállit, lapolásokat 2x rajzi méretre köszörül.	Lezárt	0	2,0ó	0,0ó	3,0ó		
	Palatköszörű ▶ Műv.:1110 Gép:SG022		Illesztett átmérőket, és válméreteket beállit.	Lezárt	0	1,5ó	0,0ó	2,5ó		
	Tömbszikra ▶ Műv.:1120 Gép:SG030		Beömlő helyet, fozott eletróda térkép és meglévő minta szerint szikráz!	Lezárt	0	16,0ó	0,0ó	7,5ó		
	Illesztés ▶ Műv.:1150 Gép:SG102		Betéteket javítandó szerszámba illeszt(Nolato)		0	8,0ó	0,0ó	0,0ó		
				Összesen:	31,1ó	0,0ó				

5. ábra
Műveletterv az Anton Kft.-nél
Forrás: Anton Kft.

3.4.5. Alvállalkozói műveletek

Ezek azok a műveletek, melyek nem végezhetőek el házon belül, hanem egy másik cég végzi és juttatja vissza a gyártási folyamatba a megmunkált termékeket. Az Anton Kft. esetében ez a művelet a hőkezelés, melyet a vállalaton kívül végeznek. Itt átfutási idővel kell számolni (általában munkanapok), ami nem adja meg, hogy pontosan mikor készül el a munka, hanem azt, hogy mennyi idő múlva lesz újra házon belül.

3.4.6. Szakaszos anyagellátás

Szériagyártás esetén lehet raktározni, mennyiségeket összevonni, hiszen az alapanyagokra folyamatos igény van. Egyedi gyártásnál ez az anyagellátási módszer nem alkalmazható. Tehát minden alapanyagot az adott gyártáshoz kell megrendelni, aminek az átfutási ideje zavarhatja az ütemtervet.

3.4.7. Kötött műszakrend

Fontos megemlíteni az egyedi gyártásban dolgozó emberi erőforrást is. A szériagyártásban a munkaerő létszáma és a műszakrend szabadabban skálázható. Az egyedi gyártásnál viszont ez nem megvalósítható, mert speciális tudással rendelkező szakemberek kellene a termeléshez. A szakképzett munkások hiánya sok esetben nehezen pótolható.

3.4.8. Zavaró tényezők, javítások

Az Anton Kft. szerszámüzeme a projektmunkán kívül a szériagyártást is támogatja. Ez a termelő eszközök javítását és karbantartását foglalja magában. A javítások ugyanazon az erőforrásokon történnek. A meghibásodott részegységeket, alkatrészeket, vissza kell helyezni a gyártásba. Ez általában azonnali ad-hoc gyártási igényt jelent. Ez felborítja a prioritási sorrendeket, és módosítja a rövidtávú ütemtervet is.

4. Az ütemező rendszer specifikációja

A továbbiakban az Anton Kft. tevékenységét elemezzük gyártásütemezése gyártásütemezési szempontból. Az egyedi gyártásból kifolyólag most projektek vannak, melyek gyártási rendelésekből állnak. A gyártási rendelés úgynevezett műveletekből áll (pl. marás, szikraforgácsolás, stb.) melyeket a műveletterv tartalmaz. A keresett optimális ütemezésekben ezeket műveleteket kell majd a gépekhez rendelni, úgy, hogy az egyes korlátozásokat be kell tartani.

Két alapvető adatbázis áll rendelkezésre, az egyik az Anton Kft. vállalatirányítási rendszerének adatbázisa, a másik pedig a kialakított ütemező rendszeré, melyek egymással egy vagy kétirányú kommunikációt képesek folytatni. A vállalatirányítási rendszerből az ütemező rendszerbe történő kommunikáció kötelező jellegű, visszafelé pedig döntés kérdése, hogy akarunk-e visszaírni. A visszaírás akkor kaphat nagy jelentőséget, ha a létrejövő ütemterv adatai más vállalati folyamatokra lehetnek hatással (például beépülő alapanyagok beérkezésének esedékessége).

4.1. Gépek modellezése

A gépek hierarchiákba szervezve adóttak. A géphierarchia mindig egy gyökérrel rendelkező erdő, melynek levelei az egyedi gépek. Az erdő mindig három szintű lesz, melynek a gyökere az üzemet jelenti, a következő szint a gépcsoportokat, míg a levelek pedig az egyedi gépek lesznek. A gépekre vonatkozó egyes paraméterek az általános elv szerint az erdő bármelyik szintjén megadhatóak. Bármelyik gépre vonatkozó konkrét paraméter érték a gép szülőjétől függ, tőle örökli az értéket, ahol ez meg van adva.

Például: legyen adott egy M_1 és M_2 gép, valamint az L_1 gépcsoport. A két gép az L_1 gépcsoport tagja. Ha az L_1 -ben a géphatékonyság 0,85-el adott, viszont az M_2 gép hatékonysága 0,9, akkor az M_1 gép hatékonyságára vonatkozó érték a 0,85.

Az ütemezésről fontos tudni, hogy gépenként történik, ezért L_l -el jelöljük a géphierarchia egy l elemét, az M_i pedig továbbra is az i -edik gépet jelent. Az M_i jelölés esetén feltételezzük, hogy a az egyes paraméterek a hierarchia alapján specifikusan ki vannak számolva az i -edik gépre.

A shop modell tulajdonságaiból kifolyólag egy gép egyszerre csak egy gyártási rendelés egy munkáján dolgozhat, és addig dolgozik vele, míg az el nem készül, tehát nem szakítható meg másik munkával. Vannak olyan gépek, ahol az állásidő nem felel meg a modellben látott

értékeknek, ezért a ξ_1 logikai változóval meg lehet adni, hogy az l hierarchielem állásideje beleszámít-e a célfüggvénybe.

4.1.1. Kalendáriumok

A gépek előre meghatározott kalendárium szerint dolgoznak. Az M_i géphez számított kalendárium azt adja meg, hogy az i -edik gépen milyen időintervallumokban végezhető munka. Minden gépnek a saját kalendárium határozza meg, hogy mikor dolgozhat, tehát a műveletek csak olyan periódusokban végezhetők az adott gépen, amikor az a kalendárium szerint dolgozik. A munkát csak szünettel lehet megszakítani, és a szünet vége után kötelezően a félbehagyott munkát kell folytatni. Egy c kalendárium bejegyzés három adatot tartalmaz, az érvényesség kezdetét ($start_c$), az érvényesség végét (end_c), és az *avail* értéket, mely érvényességi időben munkát vagy szünetet határoz meg (logikai változó).

Egy helyszínhez tartozó Cal_l kalendáriumot az érvényesség kezdete, vége, valamint a szünetek egy olyan halmaza határozza meg, ahol az érvényességi időintervallumok nem lapolhatók át. Tehát a géphierarchia alapján egyre mélyebbre kell mennünk az egyre specifikusabb kalendáriumok felé, melyek érvényességi időintervalluma felülírja a szülő által meghatározott munkarendet. Ennek könnyű megértésére következzen egy példa:

Az **üzem kalendárium**a 2014. július 2. 8:00-tól 16:00-ig rendelkezésre állást ír elő. Az üzemben az **L1 gépcsoport kalendárium**a azt mondja, hogy ezen a napon 7:00-tól 9:00-ig szünet van, míg 17:00-tól 20:00-ig munka. Az ebbe a gépcsoportba tartozó **M1 gép kalendárium**a ugyanezen a napon 15:30 és 17:30 között szünetet ír elő. Ebből kifolyólag az M1 gép kalendárium a 2014. július 2. 9:00-tól 15:30-ig, valamint 17:30-tól 20:00-ig munkát fog előírni.

Ha egy folyamat valamilyen okból eltér a normál periódustól, akkor az az ütemező rendszer adatbázisában definiált kivételnaptár segítségével téríthető el.

4.2. Munkák modellezése

A fő fogalom a projekt fogalma, mely a vevőhöz való kiszállítandó termékkel együtt jelenti az ehhez szükséges szerszámot, és annak alkatrészeit. Egy S_j projekthez tartozik:

- tételkód ($product_j$)
- betartandó dátum(d_j)
- típus ($type_j$)

A projekt lehet valós, vagy tervezett rendelés. A valós projekt valódi vevői rendelést jelent, a tervezett rendelések pedig az MRP futás által számított jövőbeli igényeknek felelnek meg.

4.2.1. Gyártási rendelések

Egy projekt több gyártási rendelésből áll, egy gyártási rendelés vagy egy darab alkatrésznek, vagy többnek felel meg. Az S_j projekthez a $PO_{j,k}$ gyártási rendelések tartoznak. A $PO_{j,k}$ gyártási rendeléshez tartozó adatok:

- a gyártási rendelésben szereplő alkatrészek darabszáma ($qty_{j,k}$)
- legkorábbi kezdési idő ($r_{j,k}$)
- műveleti terv ($OpPlan_{j,k}$)

4.2.2. Műveleti tervek

A gyártási rendelések elkészítésének a módját a műveletterv tartalmazza. Megmondja, hogy az adott gyártási rendelésen milyen műveleteket kell elvégezni, milyen sorrendben, melyik gépen, valamint tartalmazhat egyéb feltételeket, megkötéseket. Minden gyártási rendeléshez tartozik műveletterv, a műveletterv minden műveletéhez pedig egy azonosító, tehát két művelet akkor azonos, ha azonosítójuk megegyezik. A p műveleti tervhez tartozó adatok:

- műveletek halmaza (ops_p)
- műveletek közötti precedenciák ($Precs_p$)

Az egyes műveletek sorrendje minden esetben kötött a megmunkálás folyamán.

A $PO_{j,k}$ gyártási rendelés egy o műveletéhez tartozó adatokat a jelölésjegyzékben tüntetem fel, későbbi előfordulások esetén arra fogok hivatkozni.

A gyakorlatban az egyes műveletekhez gépcsoportok vannak megadva, melyek bármelyikén végezhető az adott művelet, de a modell ettől függetlenül megengedi, hogy tetszőleges géphierarchia-elem legyen megadva, tehát lehet egyedi gép is, és gépcsoport feletti egység is. Ez azért jó, mert a modellt általánosabbá és könnyebben bővíthetővé teszi, miközben a hatékonyságot nem befolyásolja.

Egy művelethez tartozhat darabjegyzék is, melynek szerepe megadni, hogy az adott munkadarabon a megmunkálás elvégzéséhez melyik anyagból mennyire van szükség. A darab-

jegyzék tartalmaz minden szükséges beépülő anyagot, az alapanyagok és a saját gyártású beépülő anyagok közül is.

A különböző műveleti idők fajtái a következők:

- Sorbanállási idő (t^{queue}): ha az adat meg van adva, akkor a művelet kezdését késlelteti a megadott idővel. Ha a befejezési idő ígéret ($\tilde{C}_{j,k,o}$) ki van töltve, akkor nem kell figyelembe venni a sorbanállási idő értékét.
- Beállítási idő (t^{stp}): ha az adat megadott, akkor a gép által meghatározott beállítási időt írja felül.
- Műveleti idő (t^{proc}): az adott művelet egy munkadarabon való elvégzésének időigénye.
- Alvállalkozói idő (t^{extern}): külső alvállalkozó által elvégzett munka, melynek időtartama nem függ a darabszámtól. Az Anton Kft. számára az alvállalkozói művelet a hőkezelés, egy időintervallumot jelent, amíg a termékek az alvállalkozónál vannak, és nem elérhetőek a gyártás számára.
- Anyagmozgatási idő (t^{delay}): amennyiben ki van töltve, a művelet késlelteti a következő műveletet a megadott idővel. Ha nincs kitöltve, akkora teljes rendszerre megadott érték lép életbe. Ez az érték azt reprezentálja, hogy a következő művelet csak egy bizonyos idő múlva kezdődhet el, mivel szükség van a munkadarab elmozgatására a következő művelet elvégzésének helyére. Ez az érték használható arra is, hogy az ütemterv ne legyen olyan szoros, így a modell és a valós gyártás közötti különbségből adódó időbeli problémákat kiküszöbölje.

4.2.3. Precedenciák

A precedencia egy kötött sorrendet határoz meg, melyek a műveleti azonosító alapján tervez meg a technológus. Ez az alapértelmezett műveleti sorrend. A felhasználónak lehetősége van osztályokba sorolni a műveleteket, mely osztályokon belül a műveleti sorrend tetszőleges. A különböző osztályokba sorolt műveletek sorrendje kötött.

Például: egy gyártási rendelés A műveletosztályának elemei A_1, A_2 , B műveletosztály elemei B_1, B_2 és B_3 , valamint a C műveletosztály C_1 műveletből áll. Adott a $B_1 \rightarrow B_2$ precedencia is. Ez alapján példák a megengedett sorrendekre:

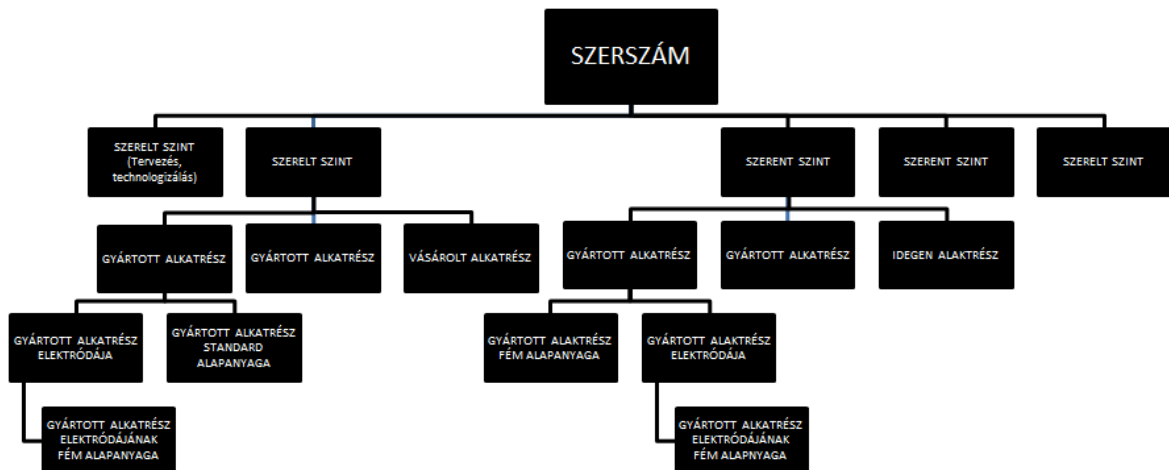
- $A_2 \rightarrow A_1 \rightarrow B_3 \rightarrow B_1 \rightarrow B_2 \rightarrow C_1$
- $A_2 \rightarrow A_1 \rightarrow B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow C_1$

Példa nem megengedett sorrendre:

- $A_1 \rightarrow B_1 \rightarrow A_2 \rightarrow B_3 \rightarrow B_2 \rightarrow C_1$

4.3. Anyagok modellezése

A BOM tartalmazza a beépülő anyagokat, mely alapján a műveletek igényelhetik ezeket.



6. ábra

Az anyagok modellezése

Forrás: saját szerkesztés

Az ábrán látható, hogy miből épül fel egy szerszám. Az építkezésnek mindig a legalsó szinttől kell kezdődnie, az alapanyagoktól. Ehhez a szinthez tartoznak a gyártott alkatrészek alapanyagai, valamint az alkatrészekhez tartozó elektródák alapanyagai. Az efölött lévő szint a félkész termékeké, melyhez az elektródák és az alkatrészek tartoznak. A legfelső a késztermék szint, mely magát a szerszámot jelenti.

A P_p anyag paraméterei a $level_p$, mely azt jelenti, hogy a hierarchia melyik szintjén foglal helyet az adott termék, és a Cal_p , mely a termék rendelkezésre állását mutatja az idő függvényében.

4.4. Szerszámok modellezése

Itt a „szerszámok” segédeszközöket jelentenek, például a gépekhez rendelt satu. Az egyedi gyártásban ezt korlátlan erőforrásként kezeljük. Ezzel a gyakorlatban a termelésvezető foglalkozik, olyan módon, hogy a szükséges eszközöket szükség esetén, párhuzamosan előre tervezi. A szerszámgyártási tevékenységnél ilyen ütközések nagyon ritkán fordulnak elő. A T_r segédeszköz paramétere a belőle rendelkezésre álló num_r mennyiség, mely jelen esetben végtelennek, és mindig rendelkezésre állónak tekintendő.

4.5. Emberi erőforrás modellezése

Az emberi erőforrás rendelkezésre állását a kalendáriumban meghatározott időablakokhoz kell igazítani. Tehát az időablakokon belül esetleges munkaerő korláttal nem számolunk. Ilyen esetekben az adott erőforráshoz tartozó naptárt kell megváltoztatni, kézi beavatkozással

4.6. Korlátozások

Korlátozásnak tekintjük a shop modellek alap megkötéseit, például hogy egy gép egyszerre egy munkán dolgozik, de ezeken felül egyéb speciális megszorítások is életbe léphetnek.

4.6.1. Csoportos végrehajtás

Az egy projekthez tartozó alkatrészeknek egyszerre, közel egy időpontban kell mozogniuk. Az alkatrészeknek az egyes szerelt szintekhez közel egy időben kell érkezniük, a gyártás optimális folyása érdekében.

4.6.2. Alapanyagok

Az ütemtervben az alapanyagokból nem léphet fel hiány, mivel a gyártás akkor kezdődik, amikor az alapanyag az igényelt mennyiségben rendelkezésre áll. Az alapanyag igényekre beszerzési rendelés, illetve készlet jelenti a fedezetet. Előbbi esetben a rendelés esedékességi dátumát kell figyelembe venni, utóbbinál pedig azonnal megkezdhető a gyártás.

4.7. Célfüggvények

Mint már az előzetes elméleti ismertetőből is kiderült, a célfüggvény többféle lehet. Két ütemterv között különbséget tudunk tenni úgy, hogy azt nevezzük ki jobbnak, melynek a legfontosabb mérőszáma jobb. Ha ezek egyenlők, akkor a második legfontosabb mérőszám vizsgálendő, hogy melyik terv esetében jobb. Egyenlőség esetén át kell térni a harmadikra, és így tovább. Az ütemező a következő célfüggvény-komponenseket kezeli:

- késés a határidőhöz képest ($\sum_j \max(C_j - d_j, 0)$);
- teljes projekt átfutási ideje ($\sum_j C_j$ ahol j valós gyártási rendelés).

Ezekon felül megadható még célfüggvényként az állásidő minimalizálása, a projekt prioritás (felhasználó által definiált változó), és gépkihasználat maximalizálása.

4.8. Globális paraméterek

4.8.1. Alapértelmezett késleltetés

Ez a paraméter azt adja meg, hogy az egy gyártási rendeléshez tartozó, egymást következő műveletek között mennyi időnek kell eltelnie. Ezt azonban felülírhatja az egyes műveletekhez tartozó anyagmozgatási idő (t^{delay}). Ezen adatokat az ütemező rendszer adatbázisában tárolandóak.

4.8.2. Időhorizontok

Befagyasztott időszak: az ütemterv kezdőnapjától számított dátum, amiben az terv már nem módosítható, kivéve anyagihiány esetén. Ez az egész ütemtervre érvényes, és ezen adatok szintén az ütemező adatbázisában tárolandóak. Egy művelet befagyasztott, ha

- beleesik a befagyasztott időszakba;
- olyan batch-be tartozik, melynek van befagyasztott eleme;
- van olyan rákövetkező művelete, ami befagyasztott.

Állásidő számításának időszaka: mivel a gyártási rendelések még nem ismertek a teljes időhorizontra, az ütemtervben egy idő után a rendszerben szereplő gyártási rendelések nem töltik ki a kapacitásokat, vagyis az ütemterv jelentős részében a gépeken mindenképpen állásidő jelenik meg. Emiatt az állásidőt itt nincsen értelme minimalizálni. Ezzel a paraméterrel megadható, hogy az állásidőt meddig kell kiértékelni (pl. a kezdőnaptól számított négy hét). A paraméter az egész ütemtervre vonatkozik. Tárolás: az ütemező adatbázisban.

Az alapanyag-korlátok ellenőrzésének időszaka is megadható, meghatározza, hogy az ütemező meddig vizsgálja az anyag rendelkezésre állásokat. A megadott időszakon túl bármely gyártási rendelés esetén megengedett anyagihiány fellépése, mivel a feltételezés szerint az adott időtávon minden anyag rendelkezésre állása garantáltan biztosítható lesz. Tárolás: az ütemező adatbázisban

4.9. Ütemterv

Egy ütemterv megadása a következő hozzárendelésekkel és sorrendezésekkel történik:

- a műveletek gépekhez rendelése;
- az egyes gépeken a hozzájuk rendelt műveletek sorrendezése;
- a gyártási rendelések felcserélhető műveletein egy sorrendezés megadása.

Vagyis egy művelet ütemezésénél meg kell adni, hogy melyik gépen kell végrehajtani, ott melyik pozícióban, valamint melyik művelet után (amennyiben felcserélhető műveletről van szó, vagyis ez nem egyértelmű). Az ütemterv minden más paramétere – pl. a kezdése és befejezési idők – számított érték, nem részei az ütemterv megadásának. Egy ütemterv pontosan akkor megengedett, ha nem tartalmaz irányított kört.

4.9.1. Végrehajtási idők számítása

Egy művelet végrehajtási ideje a beállítási időből és megmunkálási időből áll. Az M_i gépre ütemezett S_j projekt $PO_{j,k}$ gyártási rendeléséhez tartozó o művelet $s_{j,k,o}$ beállítási idejének és $p_{j,k,o}$ végrehajtási idejének számítása:

$$s_{j,k,o} = \begin{cases} 0 & \text{ha } id_o = id_{o'}, \text{ ahol az } o' \text{ az } i \text{ gépen az } o\text{-t megelőző művelet} \\ stp_o & \text{ha az előző feltétel nem teljesül, és } stp_o \text{ ki van töltve} \\ stp_i & \text{egyébként} \end{cases}$$

$$p_{j,k,o} = \begin{cases} t_o^{extern} & \text{amennyiben ez ki van töltve (alvállalkozóknál végzett művelet)} \\ \eta_i \cdot qty_{j,k} \cdot t_o^{proc} & \text{egyébként} \end{cases}$$

4.9.2. Kezdési és befejezési idők számítása

Az egyes műveletek kezdési- és befejezési idejének számítása az ütemterv egy topologikus sorrendje szerint történik műveletenként. Egy művelet legkorábbi olyan időpontban lesz elkezdve, amikor

- minden megelőző művelete a megfelelő késleltetések figyelembevételével már befejeződött;
- a gépen a megelőző művelet befejeződött;
- a végrehajtás alatt minden igényelt szerszám rendelkezésre áll;
- minden igényelt alapanyag rendelkezésre áll ($r_{j,k,o}$ alapján);
- a kezdési idő nem kerül $\bar{r}_{j,k,o}$ kézzel megadott legkorábbi kezdési idő elé, amennyiben az meg van adva;
- a befejezési idő nem kerül $\bar{C}_{j,k,o}$ befejezési idő ígéretnél hamarabbra, amennyiben az meg van adva.

A már készre jelentett műveletekre nincsen kezdési- és befejezési idő számítva, ezeknél befejezési időnként az ütemező $\hat{C}_{j,k,o}$ lejelentett befejezési időt veszi figyelembe.

Projektek szintjén is értelmezett a kezdési és befejezési idő. A j projekt kezdési ideje (S_j) az összes gyártási rendelésének összes művelete közül a legkorábban kezdődő kezdési ideje. A j projekt befejezési ideje (C_j) az összes gyártási rendelésnek összes művelete közül a legkésőbbben végződő befejezési ideje.

4.9.3. Félkész termékek raktárkészleteinek nyilvántartása

A gyártási rendelések leírása valamint a BOM-ok alapján adott, hogy mely gyártási rendelések milyen félkész termékeket gyártanak, és az egyes műveletek elvégzéséhez milyen félkész termékekre van szükség. Így a műveletek kezdési- és befejezési ideje alapján kiszámítható a félkész termékek raktárkészleteinek alakulása. Ez azért is fontos, mert a félkész alkatrészeket gyártásközi tároló polcokon tárolják. Ezek segítségével fizikailag is ellenőrizhető az ütemező által számított készletállapot.

5. Termelésütemező szoftverek

Ezen fejezetben a termelésütemező szoftverekről lesz szó, melyeket a későbbiek során különböző szempontok szerint lesznek összehasonlítva. Ezen programoknak két fajtája van:

- Dobozos: a már kész programot veszi meg a vállalat, és paraméterezés segítségével testre szabják a cég tulajdonságainak megfelelően;
- Fejlesztett: egyedi igények alapján, csak az adott vállalat számára megtervezett és létrehozott rendszer.

Az Anton Kft. vállalatirányítási rendszere a QAD. A cég esetében a szoftverválasztásnál nagy a jelentősége van annak, hogy a program rendelkezik-e QAD interfésszel. Az interfész (általában az ütemező) szoftver azon része, ami értelmezi, áttölti, konvertálja a QAD adatait a termelésütemező rendszerbe. Ennek az együttműködésnek az eredményeképpen jöhet létre az, hogy az ütemező rendszer számára ezen nyers adatok feldolgozhatóak egy ütemtervvé.

Az összehasonlító táblázat tartalmaz fejlesztett, és dobozos programokat is.

5.1. A termelésütemező szoftverek összehasonlítása

Ez a fejezet egy összehasonlítást tartalmaz azokról a termelésütemező szoftverekről, melyeket az Anton Kft. esélyesnek minősített, mint bevezetendő rendszert. Ezek:

- QAD On-Plan (Isys-On Kft.);
- QAD Planner (QAD Inc.);
- Asprova (Asprova Corporation);
- SZTAKI ütemező (MTA Számítástechnikai és Automatizálási Kutatóintézet).

A következő táblázatban a vállalati döntéshozó számára nagyon fontos szempontok szerint értékeltem ezen szoftvereket.

	QAD On-Plan	QAD Planner	Asprova	SZTAKI ütemező
Dobozos/Fejlesztett	Fejlesztett	Dobozos	Dobozos	Fejlesztett
Kidolgozott interfészek	QAD interfész	QAD interfész	QAD interfész; COM interfész (tetszőleges adatmódosítás, kigyűjtés)	Programozás során fejlesztve (QAD)
Ár (becsült)	6M Ft	6M Ft	10M Ft	11M Ft

Matematikai megvalósítás	Nem optimalizált, lineáris algoritmus	Optimum kereső	Nem optimalizált, lineáris algoritmus	Sztochasztikus optimumkeresés.
Hardverigény	↓	↓	↑	↑
Grafikai megjelenítés	Átlátható, az események manuálisan, egér használatával átütemezhetőek a megjelenítési felületen	Átlátható, a vezérlőelemek kivitelezése nem korszerű	A megjelenítés kifogásolható, sok tevékenység esetén nehezen átlátható	Átlátható, egyszerű megjelenés; fejleszthetősége a Java környezet miatt korlátozott
Kezelhetőség	Kezelése viszonylag egyszerűen megtanulható, a felhasználói felület megfelelően testre szabható	Jól kezelhető, korlátozottan testre szabható	Átlagos felhasználó számára nehezen kezelhető, tetszőlegesen testre szabható	Egyszerű, de korlátozott kezelhetőség
Testre szabhatóság	Korlátozott	Korlátozott	Jól testreszabható	Tetszőlegesen a fejlesztés során
Kiegészítő szoftverek	SQL adatbázis szerver	Nem szükségesek	SQL adatbázis szerver	SQL adatbázis szerver
Naprakésztség, fejlesztések	A gyártó folyamatos fejlesztéseket ad ki	Nem fejlesztik	A gyártó folyamatos fejlesztéseket ad ki	Csak egyedi igény alapján
Várható bevezetési időtartam	3 hónap	3 hónap	6 hónap	9 hónap
Erősségek	Szép kivitelezés, egyszerű értelmezhetőség; erős QAD integráció	Kiforrott felhasználói felület	Hosszútávon használható, mivel szabadon paraméterezhető, így képes minden igényt kielégíteni	Egyedi fejlesztés, ezért az Anton Kft. igényeinek megfelelő
Gyengeségek	Ütemező algoritmus nem optimum kereső	Elavult programozási és adatbázis technológia	Bonyolult kezelés, kezdők számára átláthatatlan a program	A terméktámogatás bizonytalan

2. táblázat

A termelésütemező szoftverek tulajdonságai

Forrás: saját szerkesztés

	QAD On-Plan	QAD Planner	Asprova	SZTAKI ütemező
Dobozos/Fejlesztett	-	-	-	-
Kidolgozott interfészek	10	8	7	10
Ár (becsült)	6	8	7	7
Matematikai megvalósítás	4	8	6	10
Hardverigény	8	5	7	7
Grafikai megjelenítés	8	8	8	9
Kezelhetőség	6	7	5	9
Testre szabhatóság	6	6	9	8
Kiegészítő szoftverek	7	5	7	7
Naprakésztség, fejlesztések	6	0	9	6
Várható bevezetési időtartam	8	8	7	6
Erősségek	-	-	-	-
Gyengeségek	-	-	-	-

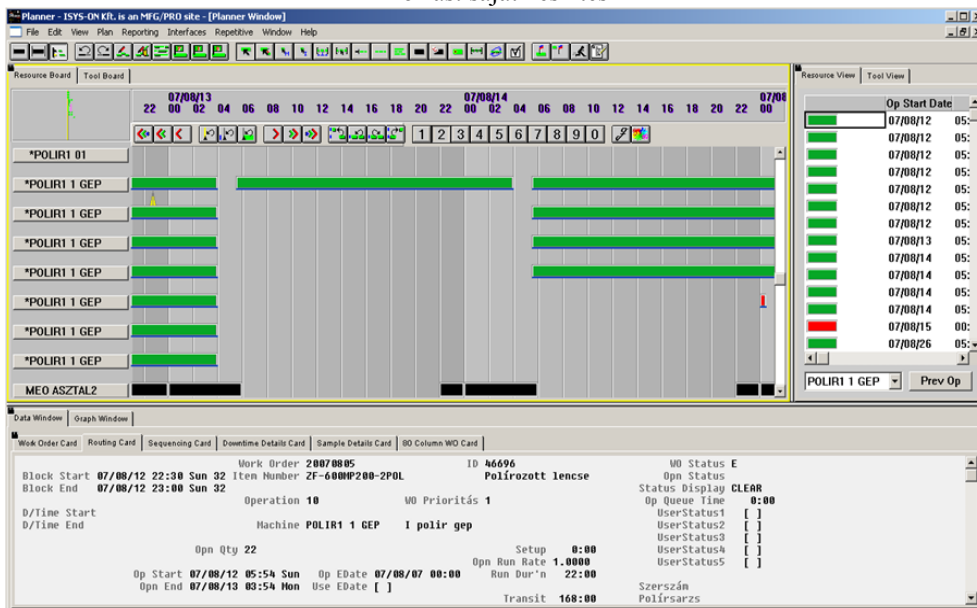
3. táblázat

A termelésütemező szoftverek pontozása a szempontok szerint

Forrás: saját szerkesztés



7. ábra
Tervezőasztal a QAD On-Plan programban
Forrás: saját készítés



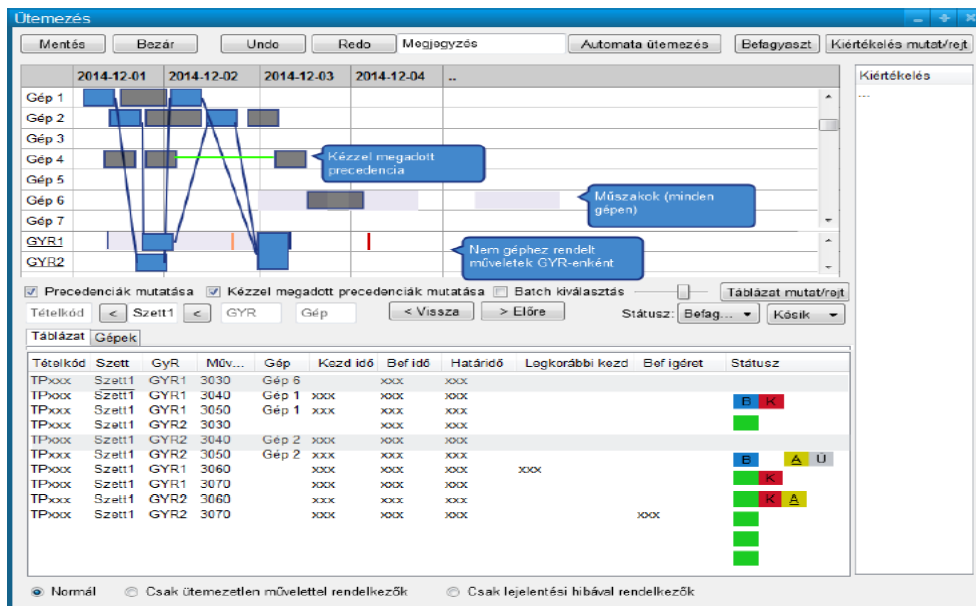
8. ábra
Tervezőasztal a QAD Planner programban
Forrás: saját készítés



9. ábra:

Tervezőasztal az Asprova programban

Forrás: ITA Group Company: Production Scheduling System Asprova. Hozzáférés(URL): <http://www.itaoffice.com/en/asprova.html>. Letöltés időpontja: 2014.11.15.



10. ábra

Tervezőasztal a SZTAKI által fejlesztett ütemező rendszerben

Forrás: saját készítés

A 3. táblázat alapján azt javaslom, hogy az Anton Kft. az ASPROVA és a SZTAKI ütemező rendszerek prototípus tesztelése után válassza ki az ütemező szoftvert.

Ennek eldöntésére javaslom a 3. táblázat értékelését újra elvégezni a tesztek eredményei alapján.

6. Bevezetési terv

Ahhoz, hogy egy vállalat sikeresen vezessen be egy új termelésütemező szoftvert, sok tényezőnek meg kell felelni.

- A célt úgy tűzzük ki, hogy félúton legyen a tökéletes és a reális között. Ez azért fontos, mert akármennyire is vonzó a tökéletesről álmodni, a valóságban sosem fog teljes mértékben megvalósulni ez az elképzelés, mivel a termelésütemezés egy nagyon nehéz terület. Ez a gondolkodásmód lesz a siker titka.
- Amennyire csak lehetséges, érdemes óvakodni a sztenderdtől nagyon eltérő egyedi igényektől. Egy esetlegesen céges kereten belül végbemenő fejlesztés rengeteg időt vesz igénybe, és a rendszerben nem várt hibák léphetnek fel.
- A cég számára hasznos lehet egy próbaverziós terméket kipróbálni. Ezek ingyenesek, hátrányuk, hogy csak korlátozott mennyiségű adattal működtethetőek, de a program látható működés közben.

6.1. A bevezetés folyamata

A bevezetés folyamata különböző lépésekből áll, melyek hosszú időt vehetnek igénybe egy ilyen összetett rendszer üzembe helyezése során. Ezen fázisokat egy egyedileg fejlesztett szoftver esetében mutatom be, ugyanis ez tartalmazza az összes lehetséges lépést, ami a fejlesztés során előjön.

		Hetek:																																												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	
Specifikáció elkészítése	<i>Vállalati specialitások</i>	3*																																												
	<i>Adatbázis egyeztetés</i>								2*																																					
	<i>IT infrastruktúra</i>													2*																																
	<i>Felhasználói felület</i>																2*																													
	<i>Programozás</i>																			-																										
Prototípus rendszer telepítés	<i>Szoftver telepítések</i>																													1*																
	<i>Paraméterezés</i>																													3*																
	<i>Tesztelés</i>																													1*																
Specifikáció aktualizálása																																					2*									
Program fejlesztés																																							-							
Éles rendszer telepítés	<i>Szoftver telepítések</i>																																								1*					
	<i>Paraméterezés</i>																																								1*					
	<i>Tesztelés</i>																																								1*					
Korrekciók																																												1*		

* jelentése: a bevezetés során a vállalatnál szükséges emberi erőforrás létszáma

11. ábra
A bevezetés tevékenységei
Forrás: saját szerkesztés

7. Összefoglalás

Egy termelésütemező rendszer kiválasztása és bevezetése komplex feladat. A dolgozatban bemutatott elméleti alapok és a vállalat sajátosságai alapján gondos előkészítés szükséges egy sikeres bevezetéshez.

A 2. fejezetben bemutatásra kerültek a termelésütemezés elméleti alapjai, mely minden eljárás és rendszer gyökerét képezik. Ezen ismeretek nélkül nehéz megérteni a különböző szoftverek közötti különbségeket, azok működésének alapelveit. Ebben a fejezetben szó volt ütemező eljárásokról, a legegyszerűbb matematikai modellekről, de mint később láttuk, egy valódi cégnél fellépő ütemezési probléma jóval túlmutat azon, hogy a munkákat megmunkálási idő szerint csökkenő, vagy növekvő sorrendben hajtjuk végre. Nagyobb hangsúlyt fektettem a determinisztikus modellekre, mivel ezeken keresztül könnyebben megérthetők az elméleti alapok. Külön alfejezetben bemutattam a sztochasztikus modell legfőbb tulajdonságait, mely sok termelésütemező szoftver matematikai alapját képezik.

A 3. fejezetben az Anton Kft. bemutatására, tevékenységének meghatározására és minőségbiztosítása szabványainak felsorolására került sor. A három tevékenység közül a szerszámgyártást jelölte ki a vezetőség, mint fejlesztendő területet, melyben jelenleg „kézi” ütemezést alkalmaznak. Ez számos problémát okoz a határidők teljesítésében és sok esetben szervezeti feszültségekhez is vezet. A fejezet fontos része volt az egyedi gyártási folyamat definiálása, és tulajdonságainak ismertetése, mely jelentősen különbözik a szériagyártási modellektől. Mindezek figyelembevételével a vezetőség igénye indokolt és érdemes egy korszerű ütemező rendszert bevezetni a szerszámgyártási tevékenységre.

A 4. fejezet az ütemező rendszer specifikációját tartalmazta, melynek fő feladata definiálni minden szükséges változót, erőforrást, melyekre a program épülni fog. A modellezések során meghatározásra kerültek a gépek, munkák, szerszámok tulajdonságai, változóik definiálása. Láthatóvá vált, hogy az ütemezési probléma mennyire bonyolult, és milyen sok pilléren nyugszik. Ennek automatizáláshoz segít hozzá az ütemterv, mely segítségével a szerszámgyártás ütemterve automatizálható, egyértelmű sorrendekben. Az anyagok modellezése egy fát mutatott be, mely reprezentálja az alapanyagok és félkész termékek helyét a hierarchiában. Szükséges betartani bizonyos korlátozásokat, melyek alapvetően meghatározzák az ütemterv használhatóságát. Az ütemező rendszer adataihoz felírt célfüggvények teszik lehetővé a rendelés kielégítését az aktuális gyártási prioritások és feladatok figyelembe vételével.

Az 5. fejezet tartalmazza a 4 termelésütemező szoftvert, melyek alkalmasak lehetnek a bevezetésre. Az Anton Kft. vezetésével a SZTAKI által fejlesztett, és az Asprova szoftvert

tartottuk a legmegfelelőbbnek, így a választás kettejük között fog eldőlni. Az egyes rendszerek között különbséget tenni nagyon nehéz, mindnek van erőssége és gyengesége. A végső döntést prototípus rendszerek tesztelésével lehet megalapozottan meghozni.

A rendszer építés fontos része a bevezetési projekt megtervezése. A 6. fejezet egy fejlesztett szoftver bevezetési folyamatát, mint projektet szemlélteti. Ennek segítségével kell a vállalatnak az ehhez rendelt erőforrásokat megtervezni.

Mindezek után, várhatóan jelentősen javítható az Anton Kft. szerszámgyártási tevékenységének ütemezési folyamata.

Ábrajegyzék

1. Ábra: Egyszerűsített számítógépes TIR modell	14
2. Ábra: Fröccsöntő szerszám	27
3. Ábra: Az Anton Kft. szerszám összeszerelő műhelye	27
4. Ábra: A szerszámgyártási tevékenység folyamatábrája az Anton Kft.-nél	28
5. Ábra: Műveletterv az Anton Kft.-nél	30
6. Ábra: Az anyagok modellezése	36
7. Ábra: Tervezőasztal a QAD On-Plan programban	44
8. Ábra: Tervezőasztal a QAD Planner programban	44
9. Ábra: Tervezőasztal az Asprova programban	45
10. Ábra: Tervezőasztal a SZTAKI által fejlesztett ütemező rendszerben	45
11. Ábra: A bevezetés tevékenységei	47

Táblázatjegyzék

1. Táblázat: A jelenlegi ütemező módszerek az egyes üzemekben	26
2. Táblázat: A termelésütemező szoftverek tulajdonságai	42
3. Táblázat: A termelésütemező szoftverek pontozása a szempontok szerint	43

8. Irodalomjegyzék

- [1] Jordán Tibor: Ütemezés, Eötvös Lóránd Tudományegyetem, Operációkutatási tanszék, 2007. ősz, 5. oldal
- [2] Jordán Tibor: Ütemezés, Eötvös Lóránd Tudományegyetem, Operációkutatási tanszék, 2007. ősz, 6. oldal
- [3] Egyetemes Lexikon, Officina Nova kiadó, Belgium, 1996, ISBN szám: 963 8185 87 2, 58. oldal
- [4] Krekó Béla: Optimumszámítás – nemlineáris programozás Közgazdasági és Jogi Könyvkiadó, Budapest, 1972. KG-1802-k-7275
- [5] Jordán Tibor: Ütemezés, Eötvös Lóránd Tudományegyetem, Operációkutatási tanszék, 2007. ősz, 14. oldal
- [6] Jordán Tibor: Ütemezés, Eötvös Lóránd Tudományegyetem, Operációkutatási tanszék, 2007. ősz, 33. oldal
- [7] Tasnádi Attila: Számítástudomány gazdaságinformatikusoknak, Bologna tankönyvsorozat, Aula kiadó, Budapest 2006.
- [8] Katona-Recski-Szabó: A számítástudomány alapjai, 5. kiadás, Typotex kiadó, 2006. 56. oldal
- [9] Krekó Béla: Optimumszámítás – nemlineáris programozás Közgazdasági és Jogi Könyvkiadó, Budapest, 1972. KG-1802-k-7275
- [10] Michael L. Pinedo, Stern School of Business, New York University, Third Edition, 2008. ISBN: 978-0-387-78934-7

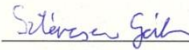
SZERZŐI NYILATKOZAT

Alulírott, Sztárcevecs Gábor büntetőjogi felelősségem tudatában nyilatkozom, hogy a szakdolgozatomban foglalt tények és adatok a valóságnak megfelelnek, és az abban leírtak a saját, önálló munkám eredményei.

A szakdolgozatban felhasznált adatokat a szerzői jogvédelem figyelembevételével alkalmaztam.

Ezen szakdolgozat semmilyen része nem került felhasználásra korábban oktatási intézmény más képzésén diplomaszerzés során.

Zalaegerszeg, 2014.12.22.



hallgató aláírása



ÖSSZEFOGLALÁS

Vállalatirányítási rendszerbe integrált termelésütemező szoftver specifikációja az Anton Kft. szerszámgyártási tevékenységére.

Sztárcsevics Gábor

Nappali tagozat/Gazdaságinformatika/Logisztikai informatikus szakirány

Diplomamunkám keretében egy gyártási tevékenységére szabott termelésütemező rendszer specifikációját készítettem el. Ennek célja a későbbiekben bevezetendő rendszer kiválasztásának és bevezetésének előkészítése. Ennek segítségével a vállalat vezetősége döntést tud hozni a megfelelő ütemező rendszer kiválasztásában.

Ezen okokból választottam szakdolgozatom témájaként azon gyártásütemező rendszer ismertetését, mely a zalaegerszegi székhelyű Anton Kft. szerszámgyártási tevékenységére épül. A jó ütemező rendszer magas fokon képes szinkronizálni a gyártást, amely hatékony erőforrás felhasználást eredményez.

Munkám során megvizsgáltam a probléma megértéséhez és megoldásához szükséges tényezőket, mint az egyedi gyártási folyamat sajátosságait, a különböző erőforrások tulajdonságait, a gyakorlati megvalósítás mikéntjét. Ahhoz, hogy teljes képet kaphassak a feladatról, szükségem volt a matematikai alapok elsajátításához, melyből a 2. fejezet tartalmaz egy nagyon fontos részt. Ezután született meg a 4. fejezet, mely már a konkrét specifikációt tartalmazza testre szabva az Anton Kft. tevékenységére.

A szakdolgozat 5. fejezete a piacon elérhető, potenciális termelésütemező szoftverek összehasonlítását tartalmazza különböző - a döntéshozó számára fontos - szempontok szerint. Az ezt követő részben pedig a bevezetési folyamat lépéseit írtam le, időbeli és emberi erőforrás igényel együtt.

Munkám során arra a megállapításra jutottam, hogy egy adott cégnél egy új rendszer bevezetése szinte mindig az emberi erőforrás függvénye. A mindenkori tudás potenciál határozza meg, hogy megéri-e új szoftvert fejleszteni, vagy egy dobozos program megvétele a megfelelő választás. A dolgozatban, értékelés alapján javaslatot tettem két különböző típusú szoftver prototípusának tesztelésére.