Miskolci Egyetem
Gépészmérnöki Kar

SZÁMITÓGÉPES TERMELÉSIRÁNYÍTÁS

(Saját fejlesztésű tananyag)

Készítette: **Dr. Tóth Tibor**
egyetemi tanár

Miskolc
2002.

Tartalomjegyzék

1. Bevezetés. Alapfogalmak …… 3
1.1. A termelésirányítás tágabb és szűkebb értelmezése …….. 3
1.2. A termelésirányítás elméleti háttere …….. 4
2. A számítógépes termelésirányítási rendszerek kialakulása és első változatai: az IBM által kifejlesztett COPICS és MAPICS rendszer …… 11
3. Rugalmas gyártórendszerek számítógépes termelésprogramozása …… 18
3.1. Milyen gyártóberendezés-csoport tekinthető rugalmas gyártórendszernek? 18
3.2. Rugalmas gyártórendszerek típusai (fizikai komponensek alapján) …… 20
3.3. Fizikai rugalmas gyártórendszer típusok (Buzacott, Groover) …… 20
3.4. Rugalmas gyártórendszerek termelésprogramozási alapesetei …… 21
3.5. Off-line irányítási környezetű FMS-ek termelésprogramozási rendszere 25
3.6. A termelésprogramozási rendszer felépítése off-line irányítási környezetben 27
3.7. A termelésprogramozási rendszer működése *on-line* üzemmód esetén .. 29
4. A termelésirányítás optimálási problémái …….. 31
4.1. A termelésirányítás feladata …… 31
4.2. A termelésirányítás célja, az optimális irányítás fogalma …… 31
4.3. A termelésirányítás hierarchikus szintjei ……. 32
4.4. Vezetési struktúra …… 32
4.5. Modellezés …… 32
4.6. A számítógép és az ember viszonya …… 33
4.7. Termeléstervezés ……. 33
4.8. Termelésütemezés …… 34
4.9. Termelésprogramozás …….. 35
4.10. Nagyméretű optimálási feladatok megoldása …….. 36
4.11. Termelésirányítási rendszerek ipari bevezetése ……. 38
4.12. Gépipari gyártásformák, irányítási feladatok …… 38
4.12.1. Általános szempontok …….. 38
4.12.2. A gyártásformákat jellemző tényezők …….. 39
4.12.3. Néhány gyakran előforduló gyártásforma …… 41
4.13. Integrált gyártórendszerek termelésirányítása (IGYR) …….. 44
4.14. Matematikai modellek a termelésirányításban …….. 44
4.14.1. Lineáris programozás …… 45
4.14.2. Diszkrét programozási problémák …… 46
4.14.3. Ütemezési problémák és megoldási lehetőségeik …… 51

1. Bevezetés. Alapfogalmak

A "termelésirányítás" szakkifejezés, elterjedtsége ellenére, nem egyértelmű és sokféle értelmezésre ad lehetőséget. Nincs szándékunkban az ismert eltérő felfogások értékelő áttekintésébe bocsátkozni, de szükségesnek látjuk saját álláspontunk kifejtését, hogy a későbbiekben a PPS/CAxx kapcsolatok vizsgálatát konzisztens elvi alapokra építhessük.

A "termelésirányítás" kifejezés pontatlan és nem szerencsés, mert az "irányítás" szó a *végrehajtást* emeli ki és legfeljebb áttételesen utal arra, hogy ezt a tevékenységláncot különféle időhorizontú *tervezési* fázisoknak kell megelőznie. E pontatlanságot újabban a "termelés tervezése és irányítása" terminológiával próbálják a szakirodalomban kiküszöbölni. A hazai szakmai köznyelv - praktikus rövidsége miatt - általában megmarad a *termelésirányítás* kifejezés mellett, hallgatólagosan tudomásul véve annak többértelműségét.

A külföldi szakirodalomban a *termelésirányítás különféle részterületeit* az alábbi terminológiák fedik:

- Management Information System (MIS),
- Production Management (PM),
- Production Planning and Scheduling (PPS),
- Manufacturing Resource Planning (MRP),
- Computer Aided Production Planning (CAPP).

Ezek közül az ESPRIT javaslatban szereplő utolsó elnevezést a CAPP rövidítés más irányú lekötöttsége miatt egyáltalán nem használhatjuk. Arra is felhívjuk a figyelmet, hogy a német szakirodalomban a PPS rövidítés a "Produktionsplanung und Steuerung" kifejezést fedi. Ez szó szerint megfelel a korábban említett, a "termelés tervezése és irányítása" terminológiának.

A további *négy* kifejezés közül az általunk vizsgált szakterületet legjobban a PPS fedi, ha deklaráljuk, hogy

- a tervezést elvileg tetszőleges mértékig számítógép támogathatja, továbbá
- a gyártási folyamat irányítása (különösen az *on-line*, *real time* irányítás) nem tartozik az általunk elemzett témakörökhöz.

Az előadásvázlatban ezért használjuk következetesen a PPS rövidítést.

1.1. A termelésirányítás tágabb és szűkebb értelmezése

Tágabb értelemben a termelésirányítás a termeléssel kapcsolatos döntési feladatok megfogalmazásával és megoldásával foglalkozik, figyelembe véve a termelés főbb tényezőit (eszközök, tárgy, feltételek és célok) és ellenőrizve a döntések végrehajtását. Ez az értelmezés tartalmát tekintve kiterjed a termelés tervezésére is, a PPS elnevezés éppen a *tervezést* hangsúlyozza.

Szűkebb értelemben termelésirányításon termelési feladatok meghatározását és végrehajtásuk megszervezését értjük. A szűkebb értelemben vett termelésirányítás feladatainak csoportosításakor az *időhorizont* nagysága jelenti a rendező elvet, azaz felülről-lefelé haladva egyre kisebb időhorizontú tevékenységek tervezéséről és/vagy irányításáról van szó.

Az időhorizontok *relatív* nagysága a termékek bonyolultságától és egyéb tulajdonságaitól, a gyártandó darabszámtól és a gyártás rendszerétől függ.

Átlagos bonyolultságú termékek kis- és közép-sorozatgyártására berendezkedett gépipari vállalat esetében a szokásos időhorizontok:

- termeléstervezés: *1-3 hónap*,
- termelésütemezés: *5-10 nap*,
- termelésprogramozás: *8-24 óra*.

Nagy bonyolultságú, súlyos termékek esetén (pl. repülőgépek, tolóhajók, úszódaruk) a három fenti időhorizont rendre például *1-3 év*, *3-6 hónap*, *10-30 nap* lehet.

A gyártási folyamatok irányítása mindkét esetben lehet részben valós idejű is (pl. CNC/DNC gyártóberendezésekre alapozott FMC-k, FMS-k *on-line, real time* folyamatirányítása).

1.2. A termelésirányítás elméleti háttere

Figyelembe véve a piacgazdaság alapvető követelményeit, a termelés tervezését, ütemezését és programozását oly módon kell megoldani, hogy

- a vevők (megrendelők) igényeinek megfelelő *szállítókészséget* (általában minél rövidebb termékszállítási határidőket),
- alacsony *készletszintekkel* (nyersanyagra, félkész- és késztermékre, pótalkatrészre, stb. egyaránt kiterjedően) és
- a termelő-berendezések és más homogén munkahelyek *kapacitásának* megfelelő mértékű kihasználásával

valósítsuk meg.

Ezt a feladatot egy adott vállalati környezetre adaptálva, nagyméretű - igen bonyolult és számításigényes - *optimumproblémaként* lehet felfogni.

A nagyon korszerű szemlélettel készült, osztott intelligenciájú, többtelephelyes hazai termelésirányítási rendszer, a KYBERNOS fejlesztői - elméleti vizsgálatok, heurisztikus megfontolások és üzemi termelésirányítási tapasztalatok alapján - az alábbi három, egymással szoros összefüggésben álló következtetést vonták le:

(1) Az előzőekben említett három fő termelési jellemző (szállítókészség, készletszint, kapacitáskihasználás) bármelyike a másik kettő rovására könnyen javítható, ezért a *három jellemzőt* (termelési tényezőt*) csak együtt lehet kezelni*;

(2) A körvonalazott optimumprobléma *logikailag a teljes termelés minden részletét* átfogja. Ez azt is jelenti, hogy a termelésirányítás optimálásához a három alapvető termelési jellemző vizsgálata *nemcsak szükséges*, hanem egyben *elégséges* feltétel is.

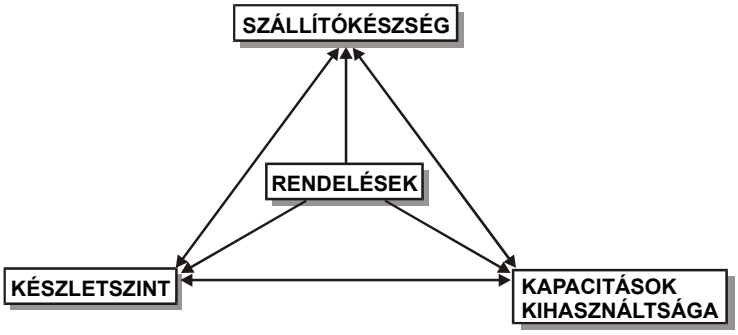
(3) A megoldási módszer *logikailag a feldolgozóipar összes vállalatára* kiterjeszthető.

Matematikai szempontból modellezve a feladatot, egyszerűen belátható, hogy a szállítókészség, a készletszint és a kapacitáskihasználás egyaránt függvénye a *gyártási* és *beszerzési rendeléseknek*.

A termelésirányítási rendszer alapfeladata tehát az, hogy olymódon határozza meg a gyártási és beszerzési rendeléseket az aktuális termelési környezet figyelembevételével, hogy az adott - alkatrészgyártó és szerelő típusú - iparvállalat a szállítókészség, a készletszint és a kapacitások kihasználása szemszögéből az "optimális munkapont" környezetében működjön. Ezt érzékelteti szemléletesen, bár nagyon leegyszerűsítve az 1.1. ábra. (Valójában a *rendelés* szó mögött tetszőleges véges - sok gyártási és beszerzési rendelés összes lényeges

információja áll jól strukturált és aktualizált formában, a szállítókészség, készletszint és a kapacitások kihasználása pedig a mindenkori összes rendelésre vetítve, szintén strukturáltan és aktualizáltan értelmezett).

Az 1.1. ábrát a KYBERNOS fejlesztői "termelési háromszögnek" nevezik.



1.1. ábra. A "termelési háromszög" modell

A *termelési folyamat* matematikailag általánosan modellezhető a három, egymással is szoros kölcsönkapcsolatban lévő *termelési tényező* részmodelljével. Ezek a tényezők a következők:

- A *szállítókészség* (delivery capability; readiness for delivery) definíciószerűen annak az időtartamnak a *reciprok* értéke (1/nap), amely egy adott rendelés elfogadásának visszaigazolásától a kiszállítás szerződésben rögzített időpontjáig eltelik. Ezt az időtartamot szimulációs módszerekkel, tapasztalati adatokra támaszkodva csak közelítőleg lehet meghatározni, megtervezve a szállítás időpontját, amely egyaránt függ a pillanatnyi raktárkészletektől, a kapacitások aktuális kihasználtságától és a beszállítók megbízhatóságától. (Ez utóbbi, nehezen számszerűsíthető hatástól első közelítésben eltekintenek azáltal, hogy a beszállítók viselkedését *racionálisnak* és teljes mértékben *kiszámíthatónak* tételezik fel.)
- A *készlet* (stock) a gyártási tételek, vagyis a raktárakban, valamint a gyártó és szerelő munkahelyeken lévő *összes* termék, részegység, alkatrész, nyersdarab és egyéb szükséges anyag *mennyiségét* jelenti egy adott időpontban.

A készletgazdálkodási alapfogalmak közül itt *négyet*, fontosságára való tekintettel, kiemelünk:

- *Készletnorma*: a termeléshez szükséges egyes anyagi eszközök legkisebb, legnagyobb vagy általános mennyiségének előírása (pl. anyagkészlet, gyártóeszközkészlet, stb.) [Storage standard].
- *Készlet-időnorma*: olyan anyagkészletnorma, amely egy bizonyos anyagból a termelés folyamatosságának biztosítására szükséges anyagkészletet a napi anyagfelhasználás többszöröseként, napokban fejezi ki. [Standard stock].
- *Törzskészlet* (minimális készlet, "vastartalék"): az a készletnagyság, amely alá a készlet a termelés folyamatos ellátásának veszélyeztetése nélkül nem eshet. Ez a készlet hivatott pótolni az utánpótlásban bekövetkezett váratlan fennakadás által okozott kiesést. Készlet-időnormában írják elő. [Minimal stock, base stock].

- *Folyókészlet*: egy adott időpontban meglévő, vagy tervezett készlet és a törzskészlet különbsége abban az esetben, ha a készlet nagyobb a törzskészletnél. A folyókészlet készlet-időnormája a napokban kifejezett utánpótlási idővel azonos, tehát a folyókészlet az utánpótlás beérkezése pillanatában a legnagyobb, majd napról-napra csökken és a következő szállítás időpontja előtt éri el a zérus szintet (törzskészlet). [Current stock].

- A *kapacitáskihasználás* a gyártóhelyek, vagyis azon összes üzemek, csoportok, munkahelyek és mozgóegységek időkapacitásának százalékos lekötöttségét jelenti egy adott időszakban, amelyek a munkaadagolás szempontjából önálló egységet képeznek.

E tényezők bármelyik szintű gyártási tételre és bármely szintű gyártóhelyre összegezve is létező realitások. A termelésirányítás céljainak megfogalmazásához ezek szükségesek és elégségesek is.

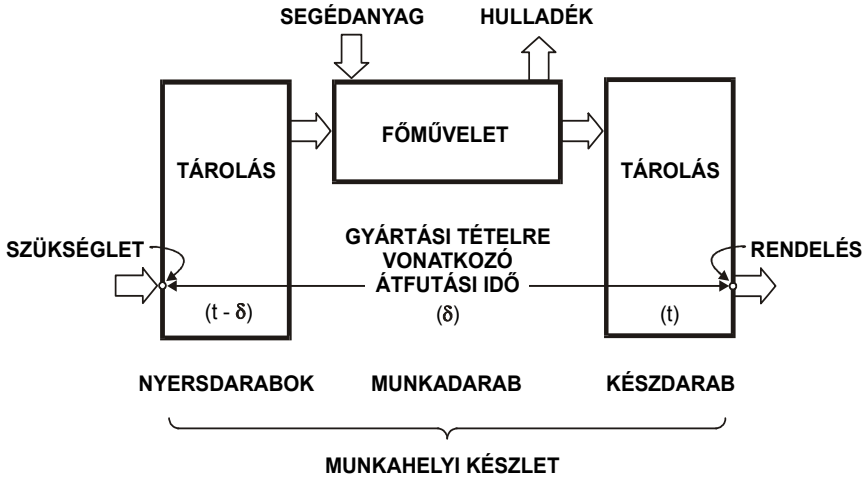
A meghatározásokban szereplő fogalmakat ismertnek tételezhetjük fel. Az anyagáramokat követve gyártmánycsaládfán lehet kimutatni a gyártási tételek (állapotok) és a gyártási folyamatok (műveletek, tevékenységek) egymásbaépülését. A folyamatrészek gyártóhelyekhez kötődnek, amelyek között tárolóhelyek (átmenő, készletező raktárak, ütemtárolók, munkahelyi rakterületek) vannak. Kereskedő vállalat esetén a gyártás egyszerű készletezésre rendukálódik (esetleg kiszerelő műveletekkel). Magas vertikumú gépgyártás esetén a struktúra tagoltsága a konstrukció és a technológia bonyolultságától függően növekszik.

A gyártási tétel *rendelését* az előkészülés (teljesítés) időpontjával kell megadni (ütemezni). Ez azt jelenti, hogy az adott gyártóhelyen a gyártást előfeltételező *szükséglet* a gyártás *átfutási idejével* korábban jelenik meg. A *rendelés-kibocsátást* tehát ez elé kell ütemezni. A rendelés így termelésirányítási *döntés* következtében jön létre (mennyiség, idő); a szükséglet a rendelés teljesítésének tárgyi *feltétele*.

A rendelések tehát lehetnek:

- *külső rendelések* (a vevői rendelések), amelyek függetlenek a belső irányítástól és így a beérkezésük révén létrejövő gyártmányszükséglet is *független szükséglet*, továbbá
- *belső rendelések* (a gyártási és beszerzési rendelések), amelyeket a belső irányítás bocsát ki a külső rendelés tárgyát képező gyártmány szerkezeti, ill. technológiai lebontása alapján és így ezek a gyártóegységek előtt szerelvény-, alkatrész-, ill. anyagszükségleteket, vagyis *függőszükségleteket* okoznak.

Az 1.2. ábrán feltüntetett *munkahelyi készleteket* (nyers-, munka- és készdarabokat) szintén a készletgazdálkodás tárgyaként kell kezelni. Ha az ábrából elhagynánk a megmunkálási blokkot, akkor ez egy olyan raktár sémája lenne, amelyben a kezelési és az átvétel-átadási műveletek miatt tranzit esetben is van átfutási idő, és ez esetleg külön programozható. Ilyen elvont modellben programozási szempontból nincs különbség a raktározó és a gyártóhely között.



1.2. ábra: A gyártási helyek általános leírásához szükséges fogalmak vázlata.

Egy raktár *anyagkezelés-technológiai* szempontból munkahely, amelynek időt igénylő műveletei éppúgy tervezhetők, mint a *megmunkálás-technológiai* műveletek (kezelés = mozgatás + rakodás + tárolás technikai eszközökkel, gépekkel). Jelen témában ettől az absztrakciótól eltekintünk és a raktárakat minőségileg megkülönböztetjük a gyártóhelyektől. A raktárak kapcsolata az irányítási fogalmakkal az 1.3. ábrán látható.

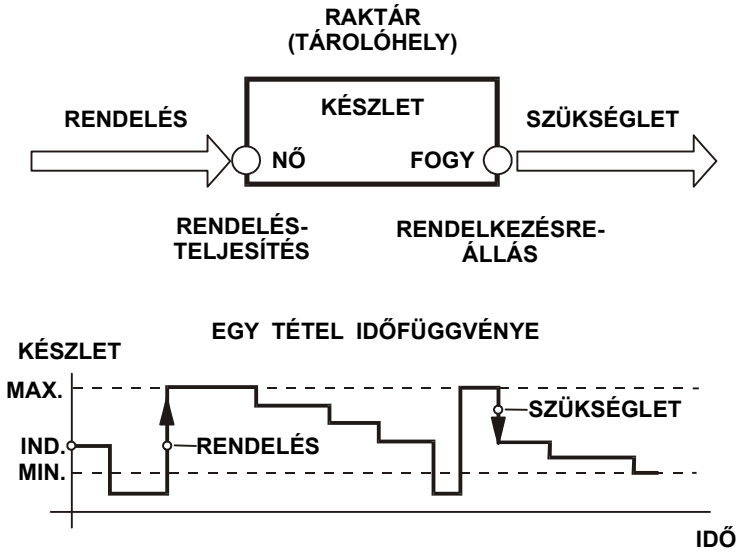
A termelés tényezői beláthatóan attól függenek, hogy miként változik a külső rendelésállomány és hogy a termelés irányítása ezt követően miként ütemezi (adagolja) a belső rendeléseket. Ennek alapján elvileg *a termelésirányítás általános feladata a termelési folyamat optimálása*, amelyben az

- *optimálás változója* a kibocsátott belső rendelések időben is változó halmaza, és az
- *optimálás célfüggvénye* a termelési tényezők halmazainak összefüggő együttese.

Nyilvánvaló azonban, hogy a célfüggvény gyakorlatilag kifejezhetetlen a változóhalmazon végzett matematikai műveletekkel és még inkább megvalósíthatatlan az, hogy a célfüggvény optimális értékéből visszafordítva (invertálva) kifejezzük az optimumhoz tartozó rendeléshalmazt.

Az optimálási "célfüggvényben" szereplő tényezők kapcsolataiból az is adódik, hogy bármelyiket előnyben részesítjük a belső rendelések kiosztásában, akkor az előnytelenül befolyásolhatja a másik két tényezőt (pl. a szállítókészség növelése károsan növelheti a készleteket és túlzó igényekkel léphet fel a kapacitások kihasználását illetően. Ez azt jelenti, hogy újabb műszak beállítása válhat szükségessé, ami nagyobb bér- és járulékos költségekkel, valamint gyorsabb berendezés-elhasználódással, kopással jár, így a ráfordítások növekednek.). Szükség van tehát egy *termelési politikára*, amely meghatározza a termelési tényezők *súlyát* az együttes termelési célban.

A termelésirányítási probléma szimbolikus megfogalmazása például a következő lehet. Vezessük be a *rendeléshalmaz* egy tetszőleges, de választás után rögzített időpontban vett jelölésére az $R(t)$ szimbólumot.



1.3. ábra: Raktárak kapcsolata a termelésirányítási fogalmakkal.

A rendeléshalmaztól függő termelési tényezőket jelöljék a következők:

$K(R(t))$ - készlethalmaz,

$C(R(t))$ - kapacitáskihasználás-halmaz,

$S(R(t))$ - szállítókészség-halmaz.

Mivel ezek különböző dimenziójú mennyiségek, csak akkor vonhatók össze egy függvénybe, ha azonos mértékre konvertálhatjuk őket a termeléspolitikát kifejező súlyfaktorokkal. Így: $\Phi(K,C,S)$ mérték-független számérték, amely a rendeléshalmaznak és ezúton az időnek is *közvetett* függvénye. A közvetett korlátfeltételek a tényezők esetében részben fizikai alapon (pl. kapacitáskihasználás esetében), részben üzleti megfontolásokkal (pl. az összkészletállomány esetében) elvileg adottak. A rendelésállományra azonban hagyományos értelemben nehéz közvetlen korlátfeltételeket elképzelni. A termelésirányítási probléma tehát a következőképpen szimbolizálható.

A $t=0$ időpontban előre meghatározandó az a $t \le T$ időpontra vonatkoztatott $R_0(T)$ élő rendelésállomány, amelyre nézve

$$\Phi(R_0(T)) \rightarrow opt.,$$

és amelyre nézve a

$$\begin{Bmatrix} K(R_0(T)) \\ C(R_0(T)) \\ S(R_0(T)) \end{Bmatrix}$$

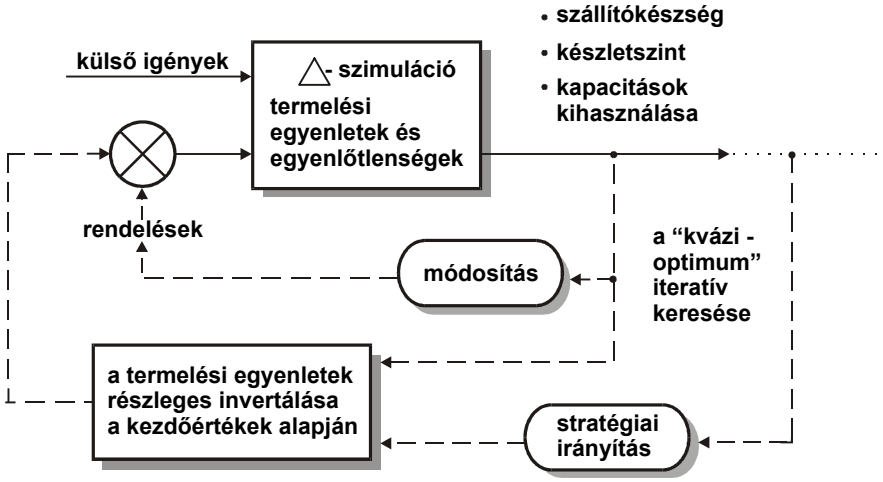
együttes az előírt korlátok közt marad. A $(0\text{-}T)$ időintervallum nem más, mint a termelés reagálási ideje a rendelések kibocsátása után. Fel kell tételeznünk, hogy a termelési szituáció (a valóságos állapot) a $t=0$ időpontban az optimáló művelet végrehajtásakor ismert, továbbá azt is, hogy a $t \le T$ időpontokra a korábbi optimálások alapján az $R_0(t)$ időfüggvényt már kiszámították (szimulálták). Ez az irányítási mód megfelelne a szabályozáselméleti "anticipatív" szabályozásnak, ha realizálható lenne.

Mivel azonban magának az optimálási műveletnek is van késleltetése (τ-holtideje), az előzetes (*0-T*) intervallumon belül is csak bizonyos diszkrét, szakaszos szabályozás képzelhető el (vagyis nem folytonos), amelyre még nincs kidolgozott matematikai modell. Következőleg az $R_0(t)$ rendeléshalmaz elemei a (*0-T*) intervallumban egy diszkrét idősorozatot képeznek, amely minden optimálás után önmagában is módosul. A valóságos teljesülések ugyanis a τ időközben eltérhetnek a (*0-T*) intervallumra előzetesen szimulált helyzetképtől (nemteljesülés, késés, sztornó, készlethiány, kapacitás-kiesés, prioritásváltozás, stb.).

Az iménti optimálási modell megközelítése csak úgy lehetséges, hogy a *termelési tényezők* halmazainak *elemeire* írjuk fel azokat az egyenleteket, amelyekkel a termelési célok részletenként érvényesíthetők és feltételezzük, hogy így az együttes optimum iterálva megközelíthető.

A "termelési háromszög" $R(t)$ független változója, valamint a háromszög három csúcsának $K(R(t))$, $C(R(t))$ és $S(R(t))$ változói között fennálló összefüggéseket a KYBERNOS fejlesztői a *termelési egyenletek* gyűjtőnév alatt matematikai formában is megfogalmazták. Hangsúlyozzák, hogy a termelésirányítás alapproblémája a gyártási és beszerzési rendelések, vagyis az $R(t)$ meghatározása, ehhez viszont az egyenletrendszer *invertálására* lenne szükség. Ez rendkívül nehéz feladatot jelent, mivel egy vállalatnál több tízezer féle anyag, illetve alkatrész is lehet, amelyek bonyolult módon épülnek egymásba; a homogén kapacitások száma pedig meghaladhatja a százat. Nyilvánvaló tehát, hogy a termelési egyenleteket csak iteratív úton lehet meghatározni. A megoldás lényegét az 1.4. ábrán látható *anticipatív* (javító) szabályozókör szemlélteti.

A szabályozókör felső ága, amely a rendelésekből az ismeretlen $K(R(t))$, $C(R(t))$ és $S(R(t))$ paraméterek egy induló kombinációját határozza meg ("termelési háromszög"-szimuláció), a termelési egyenletek alapján, egzakt módon kialakítható. A visszacsatoló ág azonban bizonytalan.



1.3. ábra: A KYBERNOS rendszerben alkalmazott megoldásjavító szabályozókör.

Mivel a termelésirányítási probléma, mint láttuk, analitikusan megoldhatatlan, heurisztikus eljárásokat alkalmaznak a következő két lépés szerint:

(1) A matematikai egyenletek többsége részlegesen invertálható. Ez azt jelenti, hogy a keresett rendelésvektor (a rendelések, mint független változók egy indexelt, rendezett halmaza) *első közelítésben* meghatározható úgy, hogy bizonyos változók értékeit konstansnak tekintik, illetve tapasztalati átlagértékeket alkalmaznak.

(2) A felhasználó döntéseinek és heurisztikus megfontolásokra alapuló beavatkozásainak kiértékelése - a "szétvágás és próbálkozás" módszerét alkalmazva - szimulációs megoldásokat képes létrehozni, amelyek egymás között összehasonlíthatók.

Az elmondottaknak megfelelően a KYBERNOS rendszer rendelkezik egy *automatikus ággal*, amely részleges invertálást hajt végre, valamint *egy módosító ággal*, amely alkalmas a felhasználó által irányított szimulációra.

A számítógépes reprezentáció kifejlesztése során a KYBERNOS itt körvonalazott *funkcionális modellje* mellett egy speciális szerkezetű adatmodell kidolgozására is szükség volt. Az *adatmodell* létrehozásakor az alábbi alapelvekből indultak ki:

(1) Meg kell határozni azoknak az adategységeknek (adatelemeknek, entitásoknak) a készletét, amelyek a rendszer működtetéséhez nélkülözhetetlenek (minimálisan szükséges adatok készlete).

(2) A minimálisan szükséges entitásokból egy olyan felépítésű *relációs adatmodellt* kell kialakítani, amelynek elemei *funkcionálisan függetlenek* a *Codd* szerinti ún. "harmadik normál alak"-nak megfelelően.

Ez az adatmodell a KYBERNOS eddigi ipari alkalmazásai alapján a következő meglepő tapasztalatokhoz vezetett:

(1) A termelési folyamatnak és irányításának leírására felhasznált adategységek (entitások) összesített száma *40-nél kevesebbnek* bizonyult, ellentétben a szakirodalomban sokfelé található, *entitások százairól* tudósító állításokkal.

(2) A funkcionálisan független adatmodell lehetővé teszi, hogy a rendszer fejlesztését olymódon hajtsák végre, hogy ne legyen szükséges a kommunikáció az egyes programmodulokat kifejlesztő szakemberek között.

A szakirodalomban a termelésirányítás komplex problémakörét általában három nézőpontból szokás megközelíteni. Ezeket, tartalmai jellemzői alapján, célszerűen *funkcionális*, *algoritmikus* és *adatmodell*-bázisú megközelítésnek nevezhetjük.

A *funkcionális megközelítés* a termelésirányítás legelterjedtebb leíró formája, amelyet már a legelső termelésirányítási rendszerek (pl. PICS, COPICS, MAPICS, HOSKYNS) esetében is alkalmaztak. Ez a megközelítés egy működő vállalat legfontosabb, indokoltan megkülönböztethető szakterületeiből indul ki és azokhoz rendel különböző funkciókat. A termelésirányítás funkciója az így definiált részfunkciók együttese (uniója) lesz. A szakterületek száma ésszerű korlátok között marad, így pl. a PICS esetében 8, a COPICS és MAPICS esetében 12-12 a megkülönböztetett területek (alrendszerek) száma. A legújabb szoftvertechnológiai eszközrendszerre és informatikai infrastruktúrára épülő komplex vállalatirányítási rendszerek, amelyekben a termelésirányítás csak az egyik összetett alrendszerként szerepel, természetesen ennél jóval több szakterületi funkcionális alrendszert is magukba foglalnak (a BPCS vállalatirányítási rendszer esetében a funkcionális modulok száma, pl. meghaladja az 50-et).

A szakterületi felosztás nagy előnye, hogy a szakemberek túlnyomó többsége számára jól értelmezhető, nemzetközi szinten is megfelelő konzisztenciájú fogalmakra épül, ezért azt a termelésirányítás alapvető megközelítésének kell tekinteni.

Az *algoritmikus megközelítés* a termelésirányítási rendszerek fejlesztőinek számára fontos, a felhasználó számára gyakorlatilag érdektelen. A termelésirányítás algoritmusai elválaszthatatlanok az egyes szakterületek igényei szerint kifejlesztett vagy adaptált *matematikai modellektől*. Ezek túlnyomó többségét az *operációkutatás*, mint alkalmazott matematikai diszciplína szolgáltatja.

Az *adatmodell-bázisú megközelítés* egyaránt fontos a termelésirányítási rendszer fejlesztői és adaptálói számára, a végfelhasználó szempontjából kisebb jelentőségű. Az adott rendszer adatmodelljében használatos entitásokat természetesen a felhasználónak is pontosan kell ismernie, mivel az ezekre való közvetlen vagy közvetett utalások a felhasználói felületen is megjelennek (pl. cikkszám, gyártókapacitás, független szükséglet).

A KYBERNOS rendszer fejlesztőinek alapvető érdeme az, hogy a világon tudomásunk szerint elsőként tettek sikeres kísérletet arra, hogy a termelésirányítás *egészét* egységes elméleti modellel, nevezetesen a "termelési háromszög" koncepciójára épülő diszkrét matematikai modellel írják le. Ez a matematikai modell a bázisa a KYBERNOS esetében a funkcionális-, algoritmikus- és adatmodell-megközelítésnek is, a rendszer összes hozzáférhető dokumentációja tükrözi ezt a komplex szemléletet. Ennek előnyei nemcsak az ár/teljesítmény viszony kedvező értékében mutatkoznak meg, hanem abban is, hogy a műszaki felsőoktatásban a KYBERNOS kiválóan alkalmazható egy olyan reprezentatív mintarendszerként, amelyen a számítógépes termelésirányítás elméleti háttere és gyakorlati megvalósítása egyaránt szemléltethető és igény szerint mélyebben is tanulmányozható.

Amennyiben a termelésirányítás komplex feladatkörét egy adott vállalaton belül működő, csak bizonyos termékek csoportját érintő CIM rendszerre lokalizáljuk, olyan PPS rendszert kapunk, amely a CIM rendszer egyik ún. *elsőszintű alrendszere*.

2. A számítógépes termelésirányítási rendszerek kialakulása és első változatai: az IBM által kifejlesztett COPICS és MAPICS rendszer

Az 1960-as évek végén kezdett foglalkozni az IBM egész iparvállalat igényeit kielégítő termelésirányítási rendszer – egyes megfogalmazások szerint termelési információrendszer – kifejlesztésével. Elsőként a 8 modulból (alrendszerből) felépített PICS (Production Information & Control System) rendszert dolgozták ki IBM 360/370-es gépekre, kötegelt feldolgozásokhoz. Kezdetben 4 súlypont alakult ki a fejlesztés során, amelyek a megváltozott hw+sw környezet és az újabb igények mellett is őrzik fontosságukat:

(1) Alapadatok, központi adatbázis struktúra: gyártmányszerkezet; művelettervek; termelő (gyártó-) berendezések; tétel, termék;
(2) Készletgazdálkodás;
(3) Anyagszükséglet számítása;
(4) Kapacitás-terhelés.

Az (1) szerinti központi adatbázis-koncepció az IBM által fejlesztett termelésirányítási rendszerek filozófiájában ma is sarkalatos pont: minden parciális felhasználó a központi adatbázisból (komplexebb, teljesebb kiépítettségű verzióknál az adatbankból) dolgozik és ez képezi a vállalati feltételekhez való illesztés alapját is.

A PICS rendszerből fejlődött ki az IBM cégnek a 80-as évek elejéig legsikeresebb termelésirányítási rendszere, a COPICS (Communication Oriented Production Information & Control System). Ez a rendszer – mint elnevezése is utal rá – *interaktív* felhasználási környezetre készült, vagyis *on-line* rendszer, amely kikerült a számítóközpont közvetlen kezeléséből, terminálokon keresztül használták. Koncepciója 1970/72 körül alakult ki, implementálása két irányban történt meg:

- System/370 és 43xx típusú *main-frame* gépekre és terminálhálózatra (eredeti COPICS koncepció);
- System/34 kisgépre és terminálhálózatra (szűkített rendeltetésű, kisebb teljesítőképességű COPICS „leképzés": MAPICS-I és MAPICS-II; MAPICS = Manufacturing & Accounting Production Information & Control System), de úgy, hogy nagyszámítógéppel való összekapcsolás is lehetséges legyen.

Történeti érdekesség, hogy eleinte az IBM csak a koncepciót adta el, a programok fejlesztésében csupán támogatásra vállalkozott. A nagy érdeklődésre való tekintettel a hetvenes évek középtől már számítógépes változatok is készültek az IBM-nél, amelyek jellemzői (egyebek között):

- Interaktív felhasználási környezet;
- Teljeskörű vállalati adatbázis (központi adatbank) a következőkre:
    - tételek,
    - termelőkapacitások,
    - technológiai folyamatok,
    - anyagtervek,
    - rendelések,
    - előrejelzések,
    - szállítók, vevők

  adatai;
- Közvetlen hozzáférés;
- Nagyszámú felhasználói funkció támogatása.

A COPICS-koncepció 12 felhasználói területet definiál az iparvállalaton belül, amelyek a következők:

(1) Műszaki és termelési (tervezési) adatok kezelése
A központi adatbázis(adatbank-) koncepció szerint az alapvető műszaki adatok strukturált előállítása, kezelése, karbantartása.

(2) Beérkező megrendelések kezelése
Az értékesítési információkat a gyártással köti össze. Ide tartozik a vevőktől beérkező megrendelések kezelése, irányítása és követése egészen a kiszállításig.

(3) Piaci előrejelzések
A vállalatvezetés számára lehetővé teszi a késztermékekre vonatkozó igények meghatározását, mennyiségi előrejelzését, egyfelől becslési algoritmusok, másfelől értékesítési tényadatok kombinált felhasználásával.

(4) Nagyvonalú gyártási főprogram (Master Schedule)
Késztermék-kibocsátás (termelés) tervezése, amely azt vizsgálja, hogy a különböző eladási tervek milyen mértékben veszik igénybe a vállalat kapacitásait, erőforrásait. Az eredmény egy nagyvonalú termelési terv, amely kiindulási alap a finomprogram készítéséhez.

(5) Készletgazdálkodás és anyagszükséglet-tervezés
A késztermékekre vonatkozó tervek alapján meghatározza a saját gyártású és a kooperációs tételek megrendelési mennyiségét.

(6) Gyártási tevékenységek tervezése (kapacitástervezés)
A finomprogram megvalósításához szükséges kapacitásokat kiszámítja és összehasonlítja a rendelkezésre álló kapacitásokkal, azaz finomítja a (4) szerint betervezett rendelések ütemezését. Célja a gyártásban lévő készletek és az átfutási idő csökkentése is.

(7) Műhely-megrendelések kibocsátása
A tervezés és a gyártás közötti csatlakozási felületet jelenti. Egy-egy rendelés gyártásba adásának időpontjában biztosítja a gyártáshoz szükséges dokumentációkat és a szükséges anyagokat.

(8) Műhelyszintű irányítás
Ez a funkció végigköveti az összes műhelymegrendelést a gyártás folyamán. Koordinálja megfelelő prioritások szerint az egymáshoz kapcsolódó fő-, segéd-, fenntartó- és környezeti folyamatelemeket, így az anyagmozgatást, szerszámellátását, a gyártóberendezéseket, stb. Mivel ide tartozik az ipari folyamatok közvetlen számítógépes irányítása is ,a gyártásautomatizálás fejlődése révén – különösen CIM rendszerekben – ez a terület kiemelkedő jelentőségű a PPS/CAPP, PPS/CAM és természetesen a CAM/CAST interfészek szempontjából is.

(9) Tervszerű megelőző karbantartás (TMK) tervezés
A TMK munkaerő- és anyagszükségletének, valamint költségének tervezésére, a meghibásodások figyelemmel kisérésére és azok elhárítására terjed ki.

(10) Beszerzés és anyagfogadás
Ez a funkció kezeli az érvényes ajánlatokat, előállítja a beszerzési megrendeléseket, követi a folyamatot a beszerzés indításától a visszaigazoláson keresztül a beérkezésig, valamint a raktárban való elhelyezésig.

(11) Raktárirányítás, fizikai raktárfunkciók
Ehhez a területhez tartozik a raktárhelyek figyelése és az anyagok tárolási helyének meghatározása, beleértve az automatizált raktárirányítás fizikai funkcióit is. A cél a raktár befogadóképességének minél jobb kihasználása, az anyagkezelési hibák és idők csökkentése.

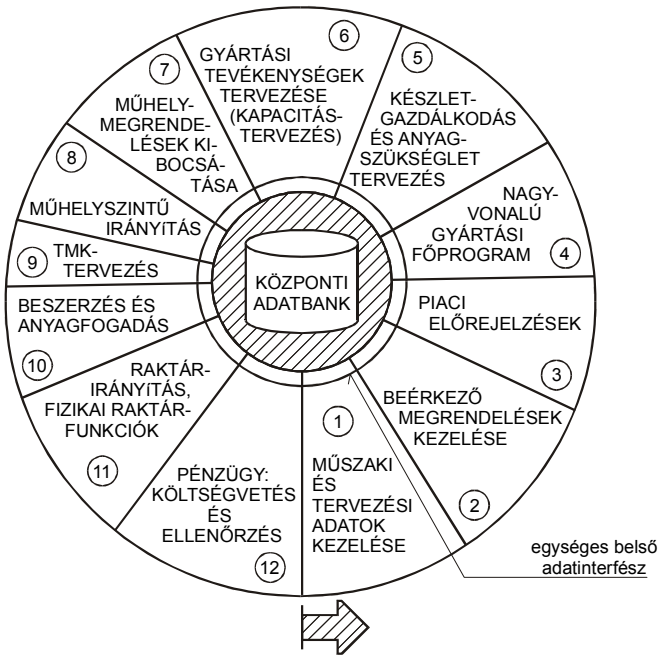
(12) Pénzügy: költségvetés és ellenőrzés
A költségvetési (költségtervezési) és elszámolási feladatok megoldásához az idetartozó információk összeállítását, kezelését és feldolgozását végzi a vállalat gazdasági vezetése számára.

A felsorolt 12 fő funkciót a 2.1. ábra, az egyes területek kapcsolódását és a taglalt funkciók időhorizontok szerinti csoportosítását pedig a 2.2. ábra szemlélteti.

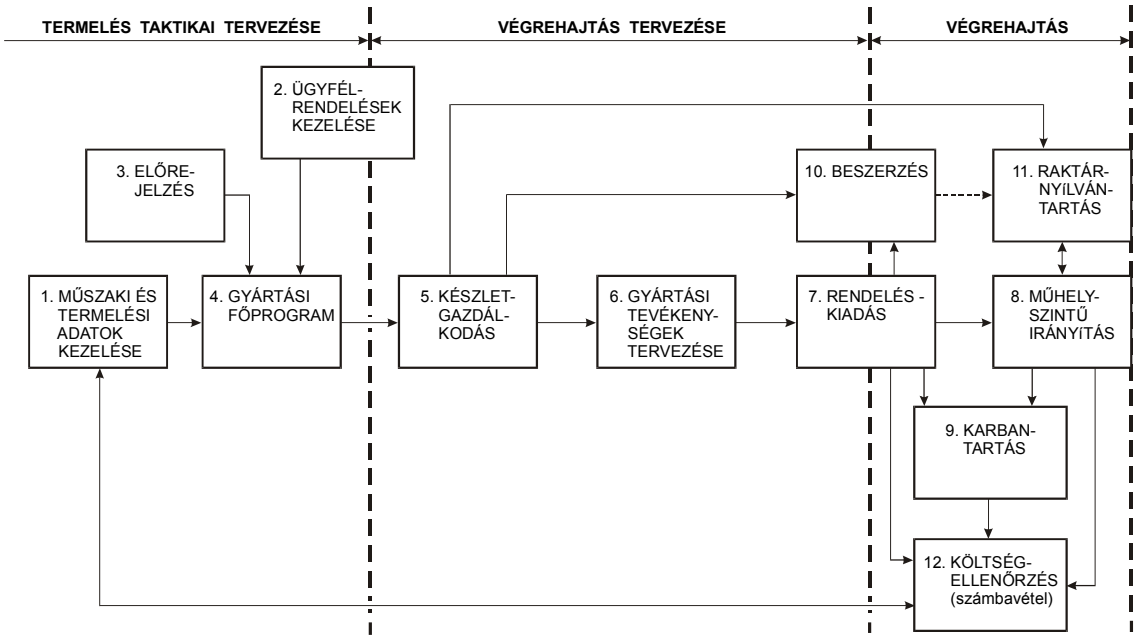
A COPICS-koncepció hatékony ipari bevezetését elősegítendő, az IBM kifejlesztette az ún. COPICS Implementation-t, amelynek 4 fő komponense van: *standard adatbáziskezelő architektúra, alkalmazási programrendszerek, gyártóipari adatbázis és tervezési, alkalmazásfejlesztési kézikönyvek.*

A *standard adatbáziskezelő architektúra* a DL/1 ill. az IMS/VS adatbáziskezelő rendszereket, valamint a CICS/VS adatkommunikációs rendszert foglalta magába. Ezek

használatával egyrészt csökkentek a rendszer bevezetéséhez szükséges ráfordítások, másrészt lehetővé vált a mindenkori vállalati termelésirányítási rendszer folyamatos bővítése.



2.1. ábra: A COPICS és MAPICS rendszerek közös funkcionális modul készlete



2.2. ábra: A COPICS/MAPICS modulok kapcsolatai az időhorizontok figyelembevételével

Az alkalmazási programrendszerek az előzőekben felsorolt fő funkciókat támogatták. A számítógépes támogatás mértékét illetően az IBM bizonyos tapasztalati prioritások szerint járt el. A vállalati tapasztalatok alapján nyilvánvaló, hogy minden COPICS-bevezetés egyedi igényeket, feltételeket és körülményeket is jelentett és a bevezetést modulonként kellett végezni. Az IBM mind COPICS, mind MAPICS implementációk esetében alapelvként deklarálta, hogy a bevezetést megelőzően gondos elemző vizsgálattal kell eldönteni, hogy mely modulokat kell először bevezetni

Az alkalmazási programrendszerek a legerősebb támogatást az alábbi területeken nyújtják (zárójelben az 5. ábrán jelzett modul-sorszámok): műszaki és termelési adatok kezelése (1); vevői rendelések kezelése (2); előrejelzés (3); készletgazdálkodás, szükséglettervezés (5); műhelyszintű irányítás (8); beszerzés, áruátvétel (10); raktárirányítás, raktárgazdálkodás (11); költségtervezés és –ellenőrzés (12).

Az alrendszerek közötti automatikus kommunikációt, üzenetközvetítést a CORMES (Communication Oriented Message System) végezte.

A *gyártóipari adatbázist* elvileg csupán a mindenkori felhasználói adatokkal kell feltölteni. Gyakorlatilag módosításra, bővítésre minden vállalatnál szükség van.

A *tervezési, alkalmazásfejlesztési kézikönyvek* a konkrét vállalat tényleges és valós igényeinek definiálásában, a programrendszerek és az adatbázisok illesztésében nyújtottak nélkülözhetetlen segítséget. Közülük a Rendszerdefiniálás-Felhasználói kézikönyv és a Rendszerdefiniálás-Adatfeldolgozási kézikönyv a legfontosabb, amelyek minden egyes alrendszerhez külön-külön rendelkezésre álltak.

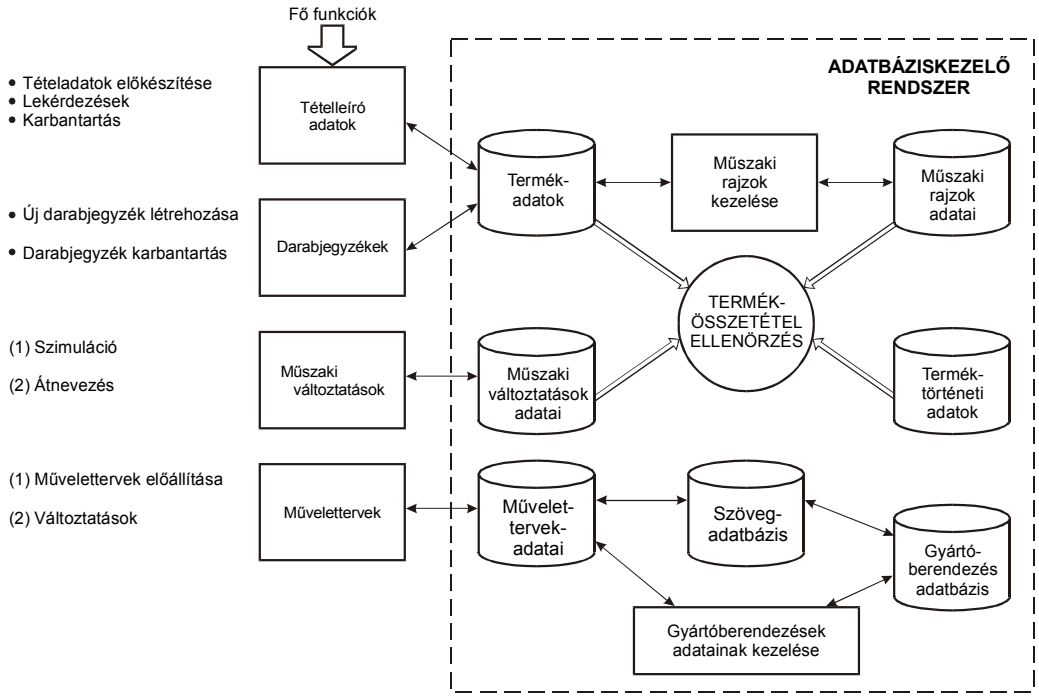
Természetesen nem minden vállalatnál volt szükséges a COPICS teljes szolgáltatási spektruma. A moduláris felépítés lehetőséget adott a lépésenkénti bevezetésre. Miután az alkalmazás sikere szempontjából a közös adatbank, valamint a műszaki és termelési alapadatok kezelését végző modul implementálása elsődleges fontosságú, ezért a bevezetést ezekkel kellett kezdeni. Mindez a MAPICS-re is érvényes.

A COPICS/MAPICS rendszerek műszaki-ügyviteli alrendszerének (1. Műszaki és termelési/tervezési adatok kezelése) teljesítőképességét egy kissé részletesebben is megvilágítjuk (2.3. ábra). Például a műveletterv adatok létrehozását (felvitelét), karbantartását és lekérdezését az alábbiak jellemzik:

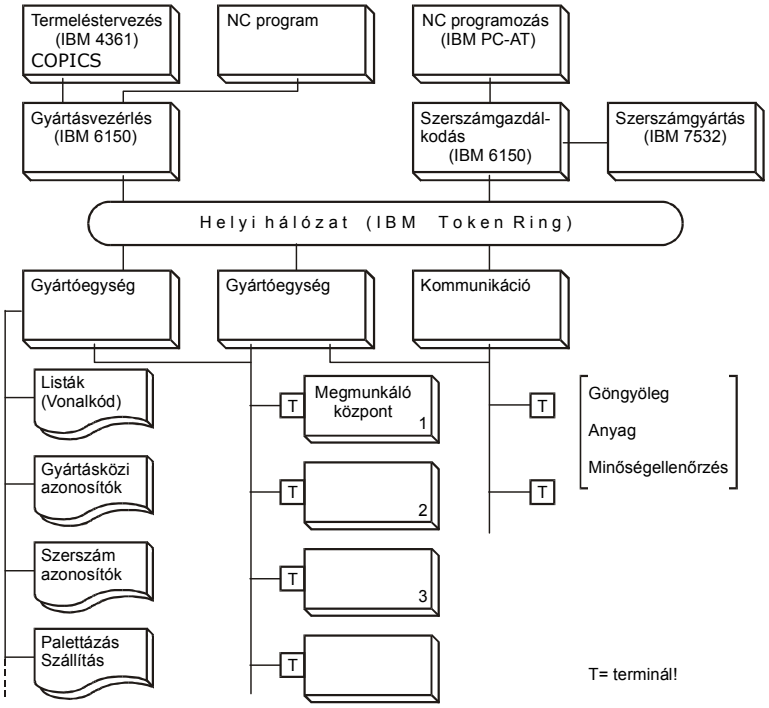
- kétféle feltöltési módszer (lassú, gyors);
- műveleti részletek másolása, műveletek újrasorolása;
- műveletterv-családok megjelenítése;
- helyettesítő (alternatív) műveletek megkeresése;
- több tétel által használt műveletterv megkeresése;
- tételhez, munkahelyhez, szerszámhoz tartozó műveletek megkeresése.

Az IBM nagy erőket koncentrált a nyolcvanas években a számítógéppel integrált gyártás (CIM) különböző részterületein saját kutatási-fejlesztési eredmények (modulok, integrált részrendszerek és rendszerek) elérésére. Az erőfeszítéseket főként a rugalmas gyártórendszerek gyártáselőkészítésének támogatására, valamint a meglévő korszerű programrendszerek közötti kommunikáció megteremtésére fordították (2.4. ábra).

Külön említést érdemel a rendkívül nagy teljesítőképességű CAD/CAM rendszer, a 17 alrendszerből felépülő CATIA, amelynek a COPICS/MAPICS rendszerekkel is vannak csatlakozási felületei (2.5. ábra).

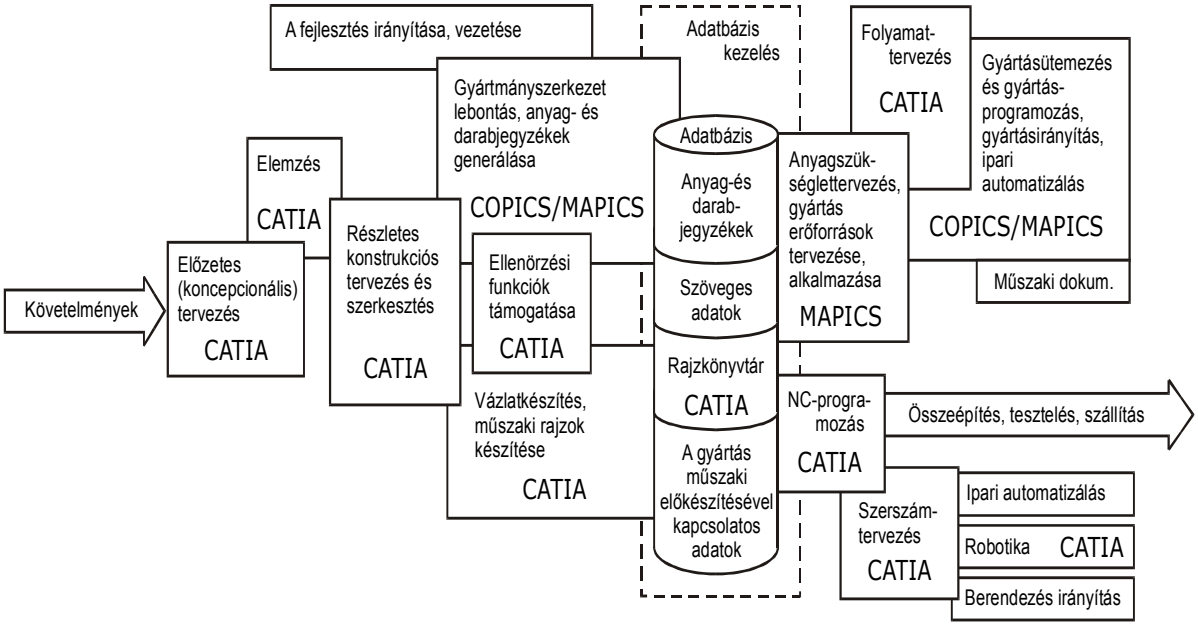


2.3. ábra: A COPICS/MAPICS rendszerek műszaki-ügyviteli alrendszerének funkcióvázlata



2.4. ábra: IBM informatikai eszközök rugalmas gyártórendszer irányításában

A COPICS rendszer rendelkezett egy-egy természetes CAD és CAM csatlakozási felülettel. A gyártmánytervezés egyik eredményét a gyártmányszerkezeti adatok, a többszintű darabjegyzékek jelentették, amelyek közvetlenül bekerültek a COPICS műszaki és termelési adatokat tartalmazó adatbázisába. Ugyanakkor a műhelyszintű termelésirányításnak közvetlenül a CAM-hoz volt output jellegű interfésze (2.5. ábra).



2.5. ábra: A CATIA és a COPICS/MAPICS rendszerek integrációja az IBM cég CIM koncepciójában

A COPICS alkalmazásokról néhány statisztikai adat:

- A nyolcvanas évek végén világszerte kb. 2500 alkalmazást tartottak nyilván, amelyek egyes részterületektől a teljeskörű implementációig a legkülönfélébb szintűek.
- A vállalati projekt-tervezések átlagos adatai a következők voltak:
  kb. 400 emberhét (programozó, szervező);
  + kb. 25 % felhasználói részvétel;
  + kb. 15 % projektvezetés;
  Összesen: 10-15 emberév ráfordítás.

(A fentiek feltételezik a megfelelő vállalati információs infrastruktúra meglétét.)

3. Rugalmas gyártórendszerek számítógépes termelésprogramozása

3.1. Milyen gyártóberendezés-csoport tekinthető rugalmas gyártó-rendszernek? (*Buzacott, Groover, Kusiak* munkái alapján)

Szempontok:

- legyen meg benne a 8 flexibilitás-típusból legalább egy;
- az automatizálás bizonyos mértéke, kiterjedtsége;
- a gyártandó alkatrészféleségek elég széles skálája, amelyet tipikusan kis- és közép-sorozatokban állit elő a rendszer.

Flexibilitás típusok

(1) *Gép (gyártóberendezés) flexibilitás*, amely megkönnyíti a gépnél (berendezésnél) való változtatásokat (pl. szerszámcsere, NC program csere, stb.), amelyek egy adott alkatrésztípusból álló halmaz legyártásához szükségesek; ez részben fejlett technológiával, részben a gépre való megfelelő műveletkijelöléssel érhető el.

(2) *Gyártási folyamat flexibilitás*: különböző alkatrésztípusok különféle műveleteket igénylő és anyagokat felhasználó változatos halmazát tudja gyártani a rendszer egyidejűleg: ez elérhető a gépek flexibilitása, ill. többcélú, adaptálható CNC megmunkáló központokkal.

(3) *Termék vagy gyártmány flexibilitás* az a képesség, hogy a rendszer gazdaságosan és gyorsan tudjon átállni új termék vagy termékcsoport gyártására; ezt a gyors reakcióképességet segíti egy hatékony és automatizált termeléstervezési és irányítási rendszer (pl. automatikus művelet hozzárendelés a gépekhez, automatikus paletta szétosztási képesség, stb.), valamint a gép flexibilitás megvalósítása.

(4) *Technológiai útvonal flexibilitás*, amely lehetővé teszi, hogy egy-egy gép kiesése (pl. meghibásodása) esetén is folytatni tudja a rendszer az adott alkatrészhalmaz gyártását előre adott technológiai változatok vagy adott helyettesítő gépek révén. Ez elérhető a technológiailag ekvivalens gépek homogén gépcsoportokba való összevonásával, műveleti technológiai variánsok felkínálásával, esetleg egy-egy műveletnek duplikált géphez rendelésével s az összes ilyen lehetőség kihasználásával a termelésprogramozás során.

(5) *Gyártási volumen flexibilitás*: képesség a különböző (esetleg kis) volumenek melletti gazdaságos gyártásra; ezt elősegítik a többcélú gépek, a gyártási folyamathoz nem kötött műhelyelrendezés, az anyagmozgató-, kezelő rendszer automatizáltsága és rugalmassága, továbbá a technológiai útvonalak rugalmassága.

(6) *A gyártórendszer kiterjeszthetőség flexibilitása*: lehetőség a gyártórendszer fokozatos, moduláris kiépítésére és kiterjesztésére; ezt elősegíti a gyártási folyamathoz nem kötött műhelyelrendezés, rugalmas anyagmozgató rendszer, moduláris gyártócellákból való építkezés és a technológiai útvonalak rugalmassága.

(7) *A gyártórendszer működési flexibilitása*

- felcserélhetőség bizonyos műveletek között (precedencia-struktúrán belül);
- nyitott technológiai útvonal.

(8) *Termelési flexibilitás*: a meglévő technológiai színvonallal függ össze és a rendszerben egyáltalán gyártható alkatrésztípusok által meghatározott.

<table>
<tr><td>GÉP
FLEXIBILITÁS</td><td>• TERMÉK FLEXIBILITÁS
• GYÁRT.FOLY. FLEXIBILITÁS
• MŰKÖDÉSI FLEXIBILITÁS</td><td rowspan="2">TERMELÉSI FLEXIBILITÁS</td></tr>
<tr><td>TECHNOLÓGIAI ÚTVONAL FLEXIBILITÁS</td><td>• VOLUMEN FLEXIBILITÁS
• KITERJESZT-HETŐSÉG FLEXIBILITÁS</td></tr>
</table>

3.2. Rugalmas gyártórendszerek típusai (fizikai komponensek alapján)

(a) szerszámgépek

- általános vagy speciális célú gépek;
- automatikus szerszámcsere képesség, szerszámtár kapacitás, szerszámcsere-szükséglet, stb.;

(b) anyagmozgató és szállító rendszerek

- változatos típusúak (pl. felrakógép, konveyor);
- az alkatrészek mozgatása (pl. raklapon, palettán);
- szerszámszállítás megoldása (pl. kézi, automatikus);

(c) műveletközi tárolási helyek

- központi puffertár;
- decentralizált puffer minden egyes gépnél;
- lokális tárolás;

(d) számítógépes irányítás

- elosztott döntési rendszer;
- információs rendszerre építkezés, centralizált döntés;
- a gyártandó alkatrészek összetételét egyszerű periodikus input szabályozza, vagy visszacsatoláson alapuló prioritási szabály.

3.3. Fizikai rugalmas gyártórendszer típusok (Buzacott, Groover)

I. Rugalmas gyártócella (FMC)

A legegyszerűbb és ezért legrugalmasabb *autonóm gyártóegység*, amely egy CNC szerszámgépből, automatizált anyagkezelő berendezésből (pl. robot) és input-output puffer-tárolókból tevődik össze.

Lényegében megvannak benne egy rugalmas gyártórendszer összes komponensei.

II. Rugalmas gyártórendszer (FMS)

Döntően *on-line, real-time* irányítású rendszer, amely általános célú szerszámgépekből, vagy ilyeneket tartalmazó gyártócellákból áll. A rendszerben kis sorozatban gyártandó munkadaraboknak számos különböző technológiai utat kell megtenniük a készregyártásig.

Az ilyen FMS az anyagmozgató és szállító berendezések változatos fajtáival lehet felszerelve. Jellemző rá a *gép-gyártási folyamat-* és *termék-flexibilitás*, valamint a *technológiai útvonal flexibilitás*, aminek következtében könnyen és automatikusan áthidalja a gép-meghibásodásokat. *Job-shop* típusú műhelynek felel meg.

III. Rugalmas transzfersor (FML)

Az alkatrésztípusok technológiai útvonala azonos, a gyártórendszer elrendezése a gyártási folyamat által meghatározott.

3.4. Rugalmas gyártórendszerek termelésprogramozási alapesetei

(1) Hagyományos gyártási környezetbe telepített FMS esetén

- egységesen *off-line* termelésirányítás és gyártási folyamatirányítás valósul meg;
- az FMS-nek a befogadó műhely szerves részeként kell működnie;
- az FMS termelésprogramozási rendszerének a befogadó műhely termelés-programozásával összhangban kell működnie, mivel *a termelésütemezés szintjéről kapott, ugyanazon tervperiódusra szóló gyártási feladaton osztoznak*.

(2) Önmagában funkcionáló, egységes egészként kezelendő FMS esetén

- az FMS, önálló műhelyhez hasonlóan, készregyárt munkadarab-sorozatokat;
- *on-line, real-time gyártási folyamat-irányítással* rendelkezik, amely lehetővé teszi az ugyancsak *on-line termelésprogramozást*;
- a gyáron belül rá kiterjedő felsőbb szintű termelésirányítás döntéseit és a rá kiszabott gyártási feladatokat önállóan valósítja meg.
- Az automatizáltsági szintje magasabb, mint az (1) típusé.

A termelésprogramozási rendszer feladata, hogy a magasabb időhorizontú termelésirányítási szinttől egy rövidebb tervidőszakra (pl. 1 hét, 1 dekád) kapott termelési feladat *gyártási műveleteit időben és térben* konkrét gépekre (munkahelyekre) *ütemezze* és ezzel előállítsa az operatív műhelyszintű irányítás alapját képező *finomprogramokat*. Mindezt adott gazdasági célok és korlátozó feltételek, valamint a termelés aktuális állapotának figyelembevételével kell ellátnia.

Követelmények:

(1) A termelésprogramozási rendszernek figyelembe kell vennie a termelésirányítás magasabb szintjén előírt, az *adott szintre aktualizált gazdasági célokat*. Ilyenek:
- a termelésütemezés szintjén deklarált határidők betartása;
- a gyártórendszer erőforrásainak hatékony kihasználása;
- a sorozatok átfutási időinek csökkentése;
- a műveletközi készletekben lekötött érték csökkentése.

(2) A termelésprogramozási rendszernek az *aktuálisan adott kapacitás- és gyártóeszköz korlátok között* kell működnie, vagyis:
- a termelőkapacitások (gépek) korlátos kapacitására *általában*, a kialakuló szűk keresztmetszetekre vagy gépkiesésekre *aktuálisan* felkészültnek kell lennie;
- a műveletközi szállítás módjától függően tekintettel kell lennie a sorozatok *egyszerre mozgatható* adagjaira (pl. raklapos továbbításnál az egység-rakományokra, palettás rendszernél a palettán mozgatott darabok számára);
- a gyártóeszköz-korlátok (pl. palettás rendszereknél a véges palettaszám) figyelembevételével kell kialakítania az *optimális sorozatnagyságokat*;

(3) A finomprogramok összeállítása során összhangot kell teremtenie a *befogadó gyártási környezet* termelésprogramozási döntéseivel;

(4) A finomprogram-készítést ajánlati jelleggel (különösen *on-line* folyamatirányítás estén) ki kell terjeszteni a *gyártási segédfolyamatokra* (pl. gyártáselőkészítésre) is;

(5) Az FMS flexibilitás-típusainak megvalósítását *szoftver-oldalról* támogatnia kell
- a munkadarab-sorozatok gyártásánál választható *technológiai útvonal variánsok*,
- *műveletenként redundáns gépek* (pl. homogén gépcsoport gépei)

közül való választással, figyelembe véve a műveletek egymás közötti felcserélhetőségét;

(6) *Real-time, on-line* irányítású rendszereknél lehetővé kell tennie az ún. *elosztott döntést* a finomprogramozás során (*dinamikus* ütemezés);

(7) Figyelembe kell vennie a döntések előkészítésekor a *termelés mindenkori aktuális állapotát* (státuszát), ugyanis csak a termelésből visszacsatolt információn alapuló döntések eredményezhetnek *végrehajtható* finomprogramokat. *On-line* folyamat-irányítás megvalósulása esetén a visszacsatolt információt egy *real-time termeléskövetési nyilvántartásnak* kell szolgáltatnia.

(8) A termelésprogramozási rendszernek tovább kell adnia a gyártásba vett munkadarab-sorozatok *készültségi fokára* vonatkozóan a termelésből visszacsatolt információt a termelésirányítás felettes szintjének;

(9) A gyártási folyamatban bekövetkező esetleges *zavarok, váratlan események áthidalásában* a termelésprogramozási rendszernek támogatnia kell a gyártási folyamatirányítást;

(10) A termelésprogramozási rendszer *szoftver* realizációjának *moduláris felépítésűnek* kell lennie a könnyű adaptálhatóság érdekében.

A követelmények kielégítésére javasolt elvi megoldások:

ad(1): *Prioritási szabály-bázis* szükséges az ütemezési eljáráshoz, amelyből a gyártási finomprogramok generálásakor az aktuális gazdasági cél és az adott termelési helyzet elemzése alapján a megfelelő prioritási szabály kiválasztható egy gép aktuális terheléséhez. *Az egyes prioritási szabályok szubrutinok formájában lehívhatók a döntésnél*. A prioritási szabályok több cél egyidejű figyelembevétele esetén kiegészíthetők bizonyos *heurisztikákkal* (Például: átszerszámozások számának minimalizálása, műveletek átlapolt végrehajtása).

ad(2): A termelőkapacitások (napi munkaidő) *norma adatai* a termelésprogramozási rendszer alapadatai közé sorolandók és ezeket a termelésből való vissza-csatolással *aktualizálni* kell. A termelésprogramozási rendszernek a kialakuló *szűk keresztmetszetű* gépeket (bottleneck) kiemelten kell kezelnie: az ütemezési eljárás döntései során mindig először ezekre történik döntés a szabad gépek közül. Az ütemezés során megfelelő *szimulációs eljárás* gondoskodik a paletták vissza-cirkuláltatásáról a gyártásba és csak elégtelen számú paletta esetén kezdeményez *sorozatbontást*.

ad(3): Hagyományos gyártási környezetbe telepített FMS esetén a termelés-programozási rendszer összhangban működik a hagyományos műhely termelésprogramozásával: ugyanazon gyártási feladat egy-egy munkadarab sorozatán osztoznak.

ad(4): Az FMS-ek termelésprogramozási rendszere az operatív gyártáselőkészítéssel való összehangolt működést a *gördülő*-(*csúszó*-) tervezés elvének alkalmazásával teszi lehetővé (mellék- és segédfolyamatok). Ennek elsősorban a gyors termékátfutásokra és termékváltásokra felkészített *on-line*, *real-time* FMS-ek esetében van jelentősége, amelyeknél a termelésprogramozási rendszer napi gyakorisággal való futtatására az igény és lehetőség megvan.

A termelésprogramozási rendszer a felettes irányítás által adott *T* tervperióduson (pl. egy dekádon) belül minden egyes napra (vagy tetszés szerinti *T1<T* időszakra) ún. "*éles*" *finomprogramot* szolgáltat az esedékes végrehajtható gyártási műveletre. Az "éles" finomprogram folytatásaként a felhasználó kívánsága szerinti további néhány napra (*T2* időperiódusra, ahol $T1+T2 \leq T$) ún. *ajánlati finomprogramot* állít elő (ha $T2 \neq 0$) a gyártási feltételekkel még nem rendelkező további gyártási tevékenységekre. Az operatív gyártáselőkészítés így megteremtheti a következő *T1* periódus gyártási feltételeit.

ad(5): A termelésprogramozás szintjén hozott ütemezési döntéseknél érvényesített *technológiai útvonal flexibilitás* szoftver oldalról végső soron a *termelési flexibilitáshoz* járul hozzá. Ehhez a termelésprogramozási rendszer bemenetén meg kell adni a technológiai útvonal variánsokat leíró gráfokat vagy fákat a műveleti törzsadatállományban. Az ütemezési eljárás során a technológiai variánsok figyelembevételével tehermentesíthetők a szűk keresztmetszetű gépek, azáltal, hogy egyes munkák számára az ilyen gépet megkerülő útvonal lesz választva.

Ugyancsak felhasználhatók a szűk keresztmetszetek feloldására a műveletekhez megadott *duplikált gépek* vagy *homogén gépcsoportok*, amelyek az ütemezési döntés számára meghagyják azt a *szabadsági fokot*, hogy a kevésbé terhelt gépre lehessen ütemezni a munkákat.

Szokás használni az ún. *előretekintési heurisztikákat* is az üres gépre vonatkozó ütemezési döntéseknél, figyelembe véve a gépre később beérkező munkákat is.

ad(6): *On-line*, *real-time* folyamatirányítású FMS-eknél a folyamatirányításban megvalósítható a *real-time termeléskövetés*. A termeléskövetés nyilvántartásában minden, a valóságos gyártásban elkezdett vagy befejezett szignifikáns gyártási tevékenység vagy esemény azonnal naplózható. Ennek alapján egy *real-time* döntésekre alkalmas *dinamikus ütemező modullal* - amely a termelés-programo-zási rendszer része - kivitelezhetők magából a gyártási folyamatból az egyes gépek felszabadulása által kiváltott, *real-time* és gépekre elosztott döntések.

Ilyenkor a gyártási folyamatirányító rendszer aktiválja működésre a dinamikus ütemező modult, valahányszor egy gép felszabadul a fizikai termelési folyamatban. Ezért a dinamikus ütemező modulnak rendkívül gyorsnak kell lennie. *A döntésnek ezen a szinten már csak a felszabaduló (vagy meghibásodott) gép fizikai várakozó sorában lévő munkára kell kiterjednie.* Ezek már mind a szükséges gyártási feltételekkel rendelkező munkák, tehát a döntés eredményeként a gépre terhelt munka (a munkadarab-sorozat valamely művelete) biztosan végrehajtható.

ad(7): Csakis a gyártási folyamatból visszacsatolt információt figyelembevenni képes termelés-programozási rendszert tekinthetünk elfogadhatónak. A visszacsatolás módja *off-line* és *on-line*, *real-time* FMS-eknél nagyon eltérő (v.ö.: ad(6)).

ad(8): A termelésprogramozási rendszer működéséhez tehát minden újrafuttatásánál szükség van a gyártás valóságos előrehaladásáról visszacsatolt információra (*Off-line* esetben pl. a termelési bérbizonylatok adatállományából, *on-line* esetben pedig a gyártási folyamatirányítás termeléskövetési nyilvántartásából nyerhető).

Nyilvánvaló, hogy minden újrafuttatás kezdetén maga a termelésprogramozási rendszer módosítja a felettes irányítás (pl. dekádtervezés) valamely adatállományban rögzített és az adott tervidőszakra előirt gyártási feladatának műveleti rekordjait, leírva belőlük a már elkészült tételeket, illetve műveleteket (Visszacsatolás a műhelyszintű irányításból a termelésütemezéshez).

ad(9): *Off-line folyamatirányítás* esetén magát a folyamatirányítást az *intelligens műhely-diszpécser* látja el. Az ő megítélésétől függően aktiválódik újra a termelésprogramozási rendszer az új adatokkal.

On-line folyamatirányítás esetén az "éles" finomprogramban ütemezett (tehát előzetesen tervezett) folyamat bármikor összehasonlítható a dinamikus ütemezési döntések alapján megvalósuló folyamattal. Ha a zavarok mértéke következtében a valóságos folyamat ún. *fatális* mértékben eltér a tervezettől, akkor a folyamatirányítás a statikus (éles) újraütemezést kezdeményezi.

ad(10): A moduláris felépítés előnyei közismertek. Hármas követelményt jelent:

- kapcsolhatóság
- csatlakoztathatóság
- újra-felhasználhatóság.

Bevált gyakorlat az egyes modulok *funkciójuk* általi definícióját használni.

3.5. Off-line irányítási környezetű FMS-ek termelésprogramozási rendszere

Hagyományos gyártási környezetben az FMS-nek a befogadó műhely szerves részeként kell működnie.

Időhorizont: 8-24 óra (1-3 műszak).

Felettes szint: *termelésütemezés*

(max. időhorizont: dekád).

Az FMS az egész műhely szempontjából

- egy nagy termelékenységű gépcsoport;
- nem szabad, hogy szűk keresztmetszet legyen.

Az FMS a műhely feletti irányítási szinten homogén módon kezelendő a műhely hagyományos részével, egy-egy tervidőszakra a megadott gyártási feladatokon azzal osztozik.

Az FMS *termelésprogramozási rendszere* a műhely hagyományos részével *kooperatív*, de *autonóm* kapcsolatban van. (Indoklás: a fejlettebb, bonyolultabb, rugalmasabb, egyben drágább rendszer *prioritása*).

Az FMS-t magába foglaló gyártóműhelyben az adott időhorizontra vonatkozó gyártási feladat munkadarab-sorozatainak *technológiai útvonala* többféle lehet:

(1) "*BENT-KINT*" *sorozattípus*: az első néhány művelet az FMS-ben, a továbbiak a hagyományos műhelyrészben;

(2) "*KINT-BENT*" *sorozattípus*: az első néhány művelet a hagyományos műhelyrészben, a további (befejező) műveletek az FMS-ben;

(3) "*BENT-KINT-BENT*" *sorozattípus*;

(4) "*KINT-BENT-KINT*" *sorozattípus*.

Diszkusszió:

ad(1): Kiemelt prioritás az FMS finomprogramozási rendszerében: az adott munkadarab-sorozat összes *külső* műveletének is el kell készülnie a tervperiódus végére (dekád, hét).[Lásd: 3.1. ábra].

Az FMS nem lehet szűk keresztmetszet. Cél (kritérium): a sorozat átfutási (megmunkálási + várakozási) idejének csökkentése. Több sorozat: egymás közötti, relatív prioritások.

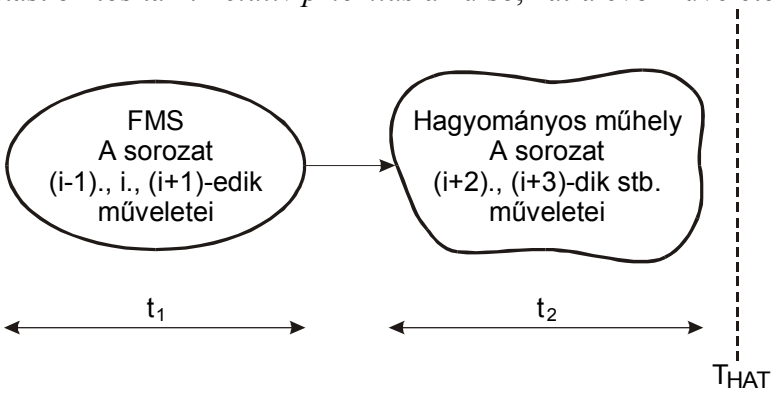
ad(2): Bizonytalan beérkezés veszélye miatt az FMS termelésprogramozásának a finomprogramokon kívüli *figyelmeztető listán* közölnie kell azon munkadarab-sorozatok jegyzékét és az FMS-be való legkésőbbi beérkezési idejét, amelyek befejező műveleteit a tervidőszak végéig az FMS-ben kell végrehajtani. [Lásd: 3.2. ábra].

Több sorozat esetén a beérkezési időket (T_{be}), a tervperiódus végét (T_{HAT}) és a hátralévő átfutási-idő igényeket kell figyelembe venni (relatív prioritások).

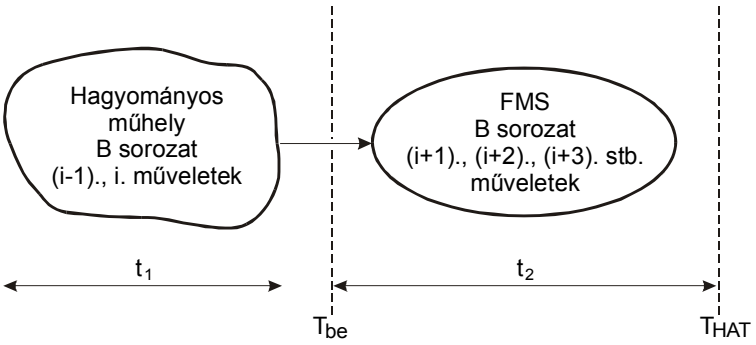
ad(3): Visszavezethető (1) és (2) kombinációjára, de a hagyományos műhelyrészbe távozott és onnan visszatérő sorozatok *új prioritást* kapnak visszatéréskor (figyelmeztető listák itt is!).

ad(4): A (2) és (1) alapesetre visszavezetve először az FMS-be a kellő időben történő beérkezésről kell gondoskodni a figyelmeztető listával. Ezután az FMS

termelésprogramozási rendszerének kell a beérkezett sorozatokra kiemelt prioritással rövid átfutást biztosítani. *Relatív prioritás* a külső, hátralévő műveletek szerint.



3.1. ábra



3.2. ábra

3.6. A termelésprogramozási rendszer felépítése off-line irányítási környezetben

Alapfeltevések:

(1) Ilyenkor a termelésprogramozási rendszer is *off-line*, *batch* (kötegelt) futtatásokkal működő rendszer, amelyet a műhelydiszpécser aktivál a szükség szerinti gyakorisággal.

(2) Az FMS termelésprogramozási rendszere számára:

- naprakészen hozzáférhetők az egész műhelyre vonatkozó rendelésállomány-ból az FMS gépeire előírt munkák (sorozatok műveletei) egy tervperiódusra;
- elérhetők az FMS gépein végzendő műveletek technológiai adatai és az FMS gépeinek normaidő (kapacitás) adatai az egész műhely feladatainak *műszaki* (technológiai) *törzsadatállományában*.

(3) Az előző tervidőszak gyártási bizonylatainak kiértékelése alapján elvégezhető a teljesítések és befejezetlen állapotok vizsgálata a gyártott tételekre.

Ez azt is jelenti, hogy

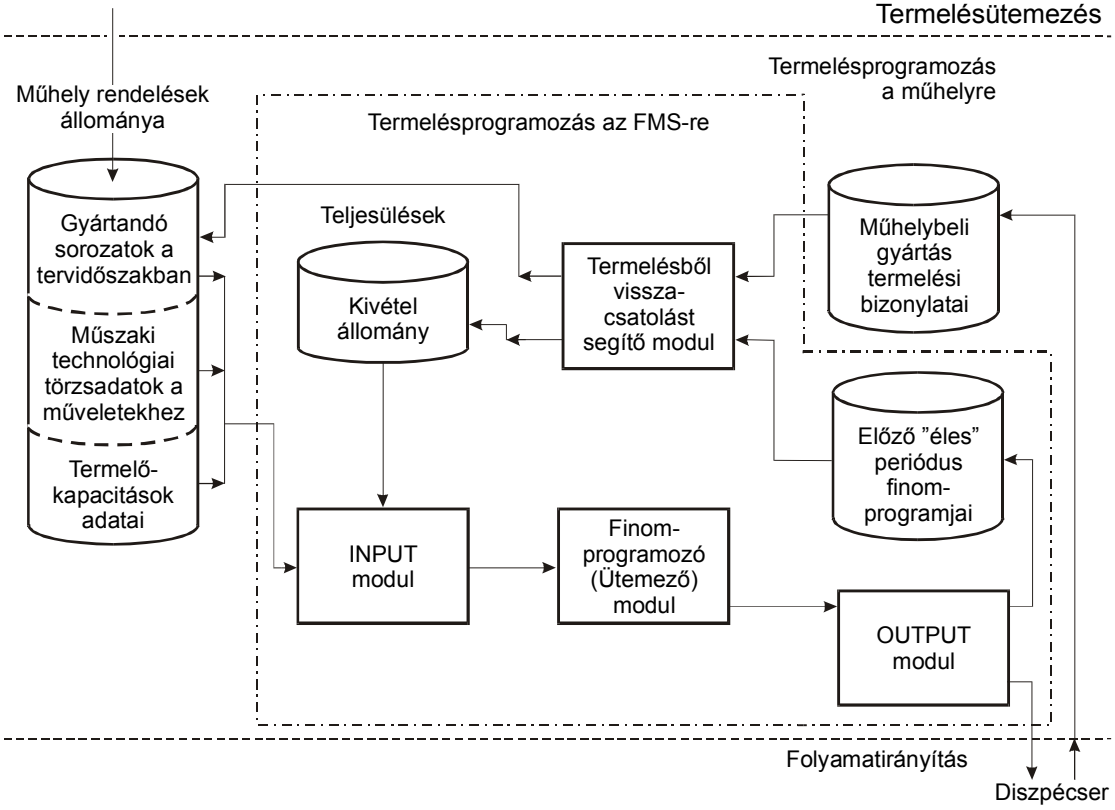
- a teljesítések leírhatók a műhely gyártandó tételeinek állományából;
- a *befejezetlen tételekből* a jelen tervidőszakra áthozott ún. *kivételállomány* képezhető, mint egyik input az aktuális finomprogram elkészítéséhez;
- megállapítható, hogy az FMS gépei milyen folyamatban lévő munkákkal és meddig vannak leterhelve.

E három lépés biztosítja a termelésből való visszacsatolást a termelés aktuális állapotáról.

A termelésprogramozási rendszer moduljai:

- az INPUT modul;
- a finomprogramozó (ütemező modul);
- a termelésből a visszacsatolást segítő modul és
- az OUTPUT modul.

[lásd: 3.3. ábra]



3.3. ábra

A termelésprogramozási rendszer interfészei és működése

Az *off-line* irányítási környezetű termelés-programozási rendszer működése a 3.4. ábrán látható.

A termelésprogramozási rendszernek 4 interfészét célszerű létrehozni (*osztott* interfész a teljes rendszerre nézve):

(1) A felettes termelésirányítási szint, a termelés-ütemezés felé az INPUT modul jelenti a csatlakoztatási felületet. Ezen keresztül az alábbi adatok hozzáférhetők az FMS termelés-programozási rendszere számára:
- *gyártandó tételek a tervperiódusban* (sorozatok azonosítói, határidői, darabszámai és gyártandó műveletei);
- *a műszaki (technológiai) törzsadatok* az előbbi műveletekhez (egy-egy művelet azonosítóihoz az előírt gép kódja, műveleti idő, előkészületi idő, gyártóeszközök kódjai);
- *a termelőkapacitások normaidő adatai*.

(2) A felhasználói interfész itt a művezető *diszpécseré*, aki aktiválja a rendszert, megadván az aktuális dátumadatokat:
- a *termelésből visszacsatoló modul* az aktuális futtatási nap dátumát kapja (a kivétel-állomány már ezzel képződik);
- az *INPUT* modul megkapja a finom-programozási tervidőszak hosszát (az éles és az ajánlati programozási napok számát) és dátumállományát.

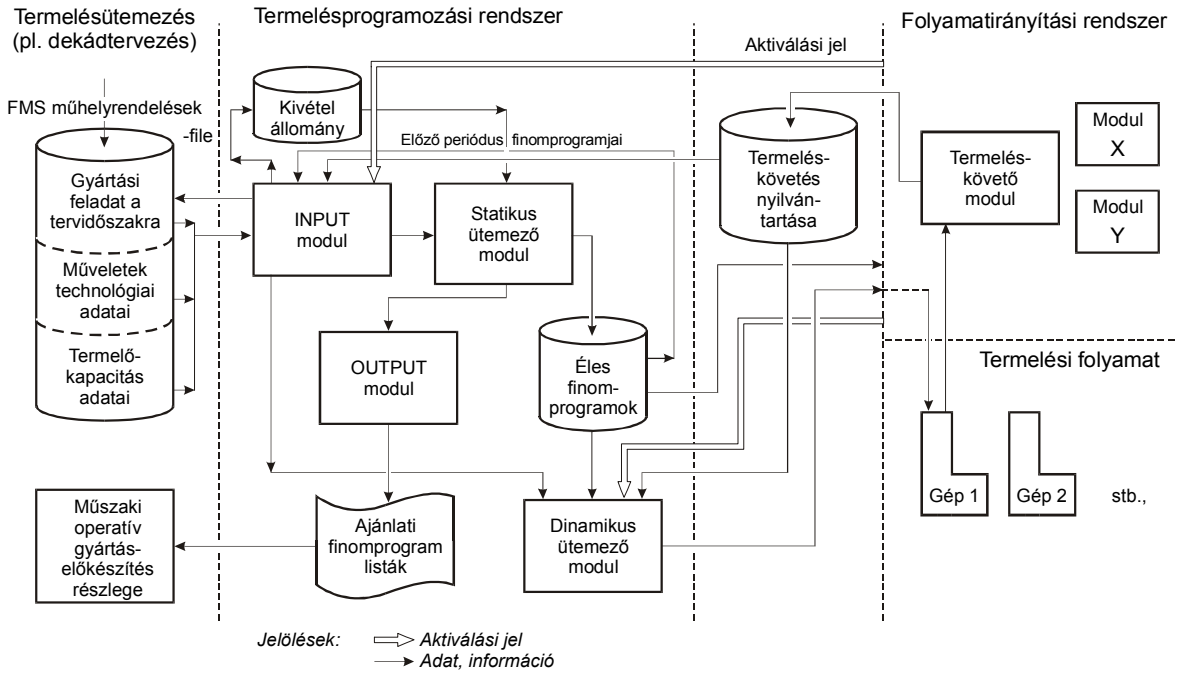
(3) A termelésből visszacsatoló modulnak nemcsak a művezető diszpécser (folyamatirányítás) felé van a (2)-ben leírt interfésze, hanem a *teljesült gyártási feladatok* leírásával a műhelyirányításon át a termelésütemezés felé is.

(4) A rendszer *OUTPUT* modulja jelenti az (*éles*) *finomprogramok tablóinak* és a *figyelmeztető listáknak* a generálásával az interfészt a folyamatirányítás (diszpécser) felé. Másrészt, az opcionálisan kérhető *ajánlati finomprogram listákon* keresztül a műszaki operatív gyártáselőkészítés (GYEK) felé ad interfészt.

3.7. A termelésprogramozási rendszer működése *on-line* üzemmód esetén

On-line, *real time* folyamatirányítású FMS esetén a rendszer rendkívül magas automatizáltsági fokú, önálló műhelyként működik, amelyben a diszpécser szerepét real-time termeléskövető program-modul veszi át. Ez azt jelenti, hogy a termeléskövetés háttértárolóra

írt nyilvántartásában minden befejezett lényeges gyártási tevékenység, valamint a valóságos gyártásban már elkezdett munkafázis gyakorlatilag azonnal megjelenik, naplózásra kerül. A rendszer része egy dinamikus ütemező modul, amely real-time döntésekre alkalmas. Szokványos, normális gyártási folyamat esetében ennek be-menetén a termeléskövetésből származó információk folyamatosan jelennek meg és kivitelezhetők magából a gyártási folyamatból az egyes gépek felszabadulása által kiváltott, real-time és gépekre elosztott döntések. Amikor egy gép felszabadul egy adott munkafázis alól, a gyártási folyamatirányító rendszer *aktiválja* működésre a dinamikus ütemező modult, amelynek emiatt nagyon gyorsnak kell lennie. Aktiválási jelként a gép meghibásodása esetén magas prioritású üzenet jelenik meg, amely azonnali intézkedésre készteti a dinamikus ütemező modult a meghibásodott gép fizikai várakozó sorának átütemezésére és a hibás gép helyettesítésének kezdeményezésére (3.4. ábra).



3.4. ábra

4. A termelésirányítás optimálási problémái

4.1. A termelésirányítás feladata

Tágabb értelemben a termelésirányítás a termeléssel kapcsolatos *döntési feladatok megfogalmazásával* és *megoldásával* foglalkozik, figyelembe véve a termelés különböző tényezőit; továbbá ellenőrzi a döntések végrehajtását.

A termelés *fő tényezőit* az alábbi csoportokba sorolhatjuk:

a) termelők (munkások, konstruktőrök, technológusok, vezetők, stb.);
b) a termelés eszközei (gépek, szerszámok, anyag, raktárak, stb.);
c) a termelés tárgya (amit termelünk);
d) a termelés feltételei (szervezettség, árviszonyok, eladhatóság, gazdasági szabályozók, egyéni érdekeltség, információáramlás, stb.);
e) A termelés céljai (nemzetgazdasági igény, nyereség, minőség, mennyiségi termelés, export, stb.).

Szűkebb értelemben *termelésirányításon termelési feladatok meghatározását és végrehajtásuk megszervezését értjük.* A továbbiakban a termelésirányításnak a technológiai folyamatok tervezésével legjobban összefüggő *három szintjére, a termeléstervezésre*, *termelésütemezésre* és *termelés-programozásra* szűkítjük vizsgálatainkat. (A termelésirányítás - mint komplex problémakör - az előző fejezetben szerepelt. A jelen fejezet terminológiája némileg eltérő a korábbitól, de ennek célszerűségét az optimálási feladatok kitűzése indokolja).

4.2. A termelésirányítás célja, az optimális irányítás fogalma

A termelésirányítás célja, hogy a rendelkezésre álló eszközök figyelembevételével minél jobban törekedjék a kitűzött komplex gazdaságpolitikai célok elérésére. A termelési feladatok megválasztásától kezdve egészen a késztermék kiszállításáig bonyolult irányítási rendszer látja el ezt a feladatot és az irányítási rendszer feladatai szintenként nagyon eltérőek.

Természetesen minden hierarchikus szinten a legjobb megoldást keressük, ez azonban nem lehet egyetlen szűken értelmezett cél (pl. max. termelési érték előállítása). Ehelyett optimális irányításról akkor beszélünk, ha a rendelkezésünkre álló információk alapján olyan irányítási döntéseket hozunk, amelyek

- nagy biztonsággal megvalósíthatók,
- sokoldalú céljainkat a lehető legjobban szolgálják,
- figyelembe veszik az egyes célok fontosságát.

4.3. A termelésirányítás hierarchikus szintjei

Még abban az esetben is elképzelhetetlen lenne a termelés minden mozzanatának egy évre előre való lerögzítése, ha a teljes évre kiterjedően ismernénk az összes termelési feladatot. Egyrészt ilyen nagy mennyiségű adatot tárolni és használni sem lehetne, másrészt teljesen áttekinthetetlen volna az egész rendszer. Így az egy időpontban fontos és lényegtelen dolgok összekeverednének.

Természetesen a szükséges adatok sem állnának rendelkezésünkre és az adódó feladatokat sem tudnánk megoldani. Minderre azonban nincs is szükség. *A termelésirányítás több, hierarchikusan egymásba illeszkedő szinten megy végbe*. A magasabb szint hosszabb időszakot fog át és csak a fontosabb szempontokat képes áttekinteni. Az alacsonyabb szinten a felsőbb szintről kapott irányítási döntések alapján készítik el az egyre részletesebb, de egyre rövidebb időperiódusra vonatkozó irányítási döntéseket. Ugyanakkor azonban, ha az alsóbb hierarchikus szint megoldhatatlan feladatot kap (akár azért, mert a magasabb szint nem tudott mindent figyelembe venni, akár azért, mert a körülmények közben megváltoztak), akkor visszajelez az eggyel feljebb lévő szintre, ahol módosítást kell végrehajtani.

A követező három alapvető irányítási szintet különböztetjük meg:

- termeléstervezés;
- termelésütemezés;
- termelésprogramozás.

4.4. Vezetési struktúra

Minél magasabb a termelés technikai színvonala egy gyárban, annál magasabb vezetési színvonal szükséges annak irányítására. A magas színvonalra jellemző egyebek között az is, hogy a vezetők összehangoltan és megfelelő előkészítés után hozzák meg döntéseiket. Az összehangoltság feltétele, hogy mindenki tudja, milyen információk alapján, milyen további eszközök segítségével és miről kell döntenie, majd az erre vonatkozó utasításait kivel kell közölnie, továbbá gondoskodik a végrehajtás ellenőrzéséről. Mindezek nélkül a legjobb irányítási rendszer is működésképtelen.

4.5. Modellezés

A termelésirányítási döntések között vannak olyanok is, amelyekhez csak megfelelő adatok szükségesek, azokból egyszerű szabályok szerint az eredmény megkapható.

Az ilyen típusú tevékenység gépies, könnyen az adatbázist hordozó számítógépre lehet bízni. Azonban a bonyolult környezeti feltételek és célok nem láthatók át. Ilyenkor célszerű a feladatot *modellezni*, azaz a legfontosabb feltételeket és célokat kiemelve optimálási feladatot kitűzni és megoldani. Ebben az esetben az illetékes vezető megalapozottabb döntésekhez juthat. Természetesen, gondosan ügyelni kell arra, hogy a modellben figyelmen kívül hagyott feltételek és szempontok nem okoznak-e nehézséget.

Gyakran problémát jelent az is, hogy a megfogalmazott matematikai feladatok egzakt megoldása reménytelenül nehéz. Ebben az esetben *közelítő megoldásokkal* vagy *szimulációval* kísérletezhetünk. Mindenképpen fennáll azonban az, hogy bonyolult feltételek és nagyszámú adat elemzéséhez a számítógép különböző módokon nagy segítséget nyújthat.

4.6. A számítógép és az ember viszonya

Egyformán nagy hiba a számítógépet (és a számítógépes módszereket) mértéktelenül túlbecsülni, vagy azt állítani, hogy a tapasztalt vezetőnek nincs rá szüksége. A számítástechnika ugyanis alapvetően megváltoztatja a modern vezetést, mivel

a) rendkívül gyorsan szolgáltatja a kívánt pontosságú adatokat, igy gyors és megfelelő beavatkozást tesz lehetővé;
b) az adatok kumulálásával (halmozásával), szelektálásával és más egyszerű átalakításával nagyobb területek működésére és a változás tendenciájára vonatkozó információhoz jutunk, következtetéseket vonhatunk le, a helyzetet elemezhetjük;
c) bonyolult feltételek mellett sokoldalú céloknak megfelelő modellek és megoldásuk segítségével hatékonyabb és eredményesebb irányítási módszerekhez juthatunk. Döntéseink realitását és annak következményeit felmérhetjük, majd újabb döntéseinket ennek tudatában hozhatjuk meg.

Lényegében tehát a számítógépek segítségével a vezetők is megszabadulhatnak gépies munkájuktól és figyelmüket az érdemi döntések meghozatalára fordíthatják, ismerve azok további kihatását is.

4.7. Termeléstervezés

Egy hosszabb időszakra (1-3 hónap) a rendelkezésre álló erőforrásokat (ember, gép, anyag, energia, raktár, stb.) figyelembe véve a vállalati céloknak legjobban megfelelő termelési tervet kívánunk létrehozni. A vállalati célok rendszerint sokrétűek: gyártási kötelezettségek; vállalási határidők; gazdaságos termelés; önköltség-csökkentés;

minőségjavítás; a termelékenység növelése; a berendezések, gépek viszonylag egyenletes terhelése; a nyereség növelése; az export növelése, stb.

A felsorolt célok összeegyeztetése igen nehéz. A termeléstervezés matematikai modelljében az *erőforrások korlátozó feltételként* szerepelnek.

Ezen túlmenően a célok többségét ugyancsak korlátozó feltételként tekinthetjük, ugyanis csak olyan tervet fogadunk el, amely ezeknek a céloknak legalább valamilyen előre megadott mértékben eleget tesz. *A legfontosabb cél lesz az optimálási feladat célfüggvénye.* Amennyiben egyetlen cél kiemelése nem reális, akkor néhány célfüggvény súlyozott összegével is dolgozhatunk. A feltételek közé helyezett célok a feladat paraméterei.

A termeléstervezés az éves, vagy annál is hosszabb távú munkavállaláson alapszik és annak kedvezőbb alakítását is lehetővé teszi. Ezenkívül a már elvállalt gyártási feladatokat igyekszik a legmegfelelőbben csoportosítani és 1-3 hónapos időközökre betervezni. Feladata tehát nemcsak a következő időszak lehető legjobb tervének elkészítése, hanem gondoskodnia kell arról is, hogy az egymást követő időperiódusok egyformán jól tervezhetők legyenek. Továbbá már az eggyel alacsonyabb irányítási szint, a termelésütemezés legfontosabb szempontjait is szem előtt kell tartania.

Minthogy minden komplex gyártási folyamatban vannak véletlenszerű tényezők, a terv ellenőrzéséről és esetleges korrekciójáról is gondoskodni kell. Erre szolgál a csúszótervezés, amelyet dinamikus tervezésnek is nevezhetünk. A *csúszótervezés* lényege az, hogy a tervperiódus (pl. 1 hónap) egy részének (pl. első hetének) elteltével újabb négyhetes tervet készítünk, amelynek első három hete az előző terven alapul, de figyelembe veszi a bekövetkezett változásokat. A negyedik hétre új terv készül, amely illeszkedik az előzőekhez, a feladatokhoz, a lehetőségekhez és figyelembe veszi a célokat. Célszerű a tervezésen belül is elkerülni a túl nagy időbeli ugrásokat, igy az éves tervet negyedévesekre, a negyedéveseket havi tervekre bonthatjuk le. Mind a három havi, mind a négyhetes tervek csúszótervezéssel készülnek.

Amennyiben egyes termékek gyártási ideje hosszabb, mint egy tervperiódus, akkor a hosszabb idejű tervnek a részfeladatok végső határidejét ki kell jelölnie. Ennél - indokolt esetben - előbb elkészülhet a részfeladat (pl. alkatrész), később nem.

4.8. Termelésütemezés

A termeléstervezés során többszáz vagy többezer munkadarab gyártási idejéről döntünk. (Melyik hónapban, melyik héten kerülnek gyártásra.) Nyilvánvaló, hogy a tervezés

szintjén nem foglalkozhatunk sem a pontos időbeosztással, sem egyéb mikrokörnyezeti problémákkal, amelyek az egyes munkadarabok egyidejű gyártásakor felmerülnek.

A termelésütemezés feladata, hogy rövidebb időszakra (**5-10** napra) az előzőleg elfogadott havi terv alapján elkészítse a munkák elvégzésének részletes ütemtervét. Ez magába foglalja az esetleges alternatív megmunkálási sorrendek közötti választást, az egyes munkák vagy az azokon végzett műveletek részére a megfelelő gépek és berendezések kiválasztását és mindenekelőtt a munkák elvégzésének időbeli ütemezését. Ez az utóbbi feladat elsősorban az egyes munkák és műveleteik elvégzésének sorrendjét jelenti.

A termelésütemezés kapcsán a terven és a berendezések kapacitásán túlmenően a következő szempontokat kell figyelembe venni:

- az egyes gyártási sorrendek anyagi és minőségi kihatásait,
- a raktárkészletek alacsony szinten tartását,
- az egyes gépek, berendezések zökkenőmentes együttműködésének megvalósítását (azaz, hogy a technológiai sorban előbb álló berendezések kellő időre anyagot, félkészterméket szolgáltassanak a következőknek),
- a berendezések viszonylag egyenletes terhelését,
- az azonos, vagy részben azonos funkciót ellátó berendezések között a munka optimális megosztását,
- a szerszámok, alap- és segédanyagok gazdaságos kihasználását.

4.9. Termelésprogramozás

A termelésirányítás felsőbb szintjei mindig valamilyen *tervezett* vagy *feltételezett* helyzetből indulnak ki. A termelésprogramozás azonban a *tényeket tükröző adatok* alapján rövid időre (8-24 óra) részletesen rögzíti a termelés menetét. A kiindulópont itt is az eggyel magasabb irányítási szintnek, a termelésütemezésnek a szóbanforgó időre (1 műszakra, ..., 1 napra) eső része. Egyrészt ellenőrizni kell, hogy a szükséges erőforrások és anyagok rendelkezésre állnak-e, továbbá az esetleg módosult körülmények között nem célszerű-e jelentős módosítást végrehajtani. Másrészt a termelés-programozás kevesebb anyaggal és rövidebb időperiódussal foglalkozik, így lehetővé válik, hogy a termelés teljes folyamatát kézbentartása. Magasabb szinten bizonyos szerelési, anyagmozgatási, szerszámelőkészítési és sok egyéb tevékenységet még nem lehet figyelembe venni. Ennek csak egyik oka az, hogy ez igen nagyszámú tevékenység és összefüggésük áttekinthetetlenné, kezelhetetlenné válik.

Legalább ennyire fontos, hogy a legpontosabban végrehajtott gyártás esetén is vannak *időbeli bizonytalanságok* és *váratlan események. Ezek áthidalása a termelésprogramozás*

feladata. Az elkészült termelési programnak természetesen bizonyos határig rugalmasnak kell lennie. Ha a termelés ennél nagyobb mértékben tér el az előírttól, akkor kellő időben új, módosított termelési programot kell készíteni.

A termelésprogramozás fő feladata tehát, hogy a zökkenőmentes termelést valósítsa meg az irányítás magasabb szintjein megadott elvek és tervek szerint. Ha módosítás szükséges, mert a tények a tervezettől lényegesen eltérnek, akkor a termelésprogramozásnak *át kell hidalnia* azt az időt, amely az ütemezés és esetleg a termeléstervezés feladatainak újbóli megoldásáig eltelik.

4.10. Nagyméretű optimálási feladatok megoldása

Gyakran előfordul, hogy egyszerű típusú és az igényekhez mérten nem nagyméretű optimálási feladatok egzakt megoldása meglepően hosszú ideig tart.

Ebből azonban nem szabad azt a következtetést levonni, hogy nagyobb termelésirányítási feladatok számítógépen úgyis megoldhatatlanok, tehát marad a régi módszer (ami gyakran találgatást jelent).

Az alábbiakban megemlítünk néhány módot, amelyekkel a matematikai modellek használhatósága kiterjeszthető.

a) ***Vertikális dekompozíció***

Ugyanúgy, ahogy a termelésirányítást három hierarchikus szintre és igy különböző típusú feladatokra osztottuk, e szintek számát - legalábbis a számítógépen belüli technikában - még tovább növelhetjük. Ennek egyik megvalósítása, hogy egy bonyolult termelésprogramozási feladatot időben egymást követő részekre bontunk. A részfeladatok megoldása után megkíséreljük a teljes probléma megoldását javítani, vagy ennek lehetetlenségét kimutatni.

b) ***Horizontális dekompozíció***

Viszonylag lazán összefüggő termelő egységeket időszakosan külön vizsgálunk, majd ismét az együttes problémával foglalkozunk.

c) ***Aggregálás***

Ha a termékek viszonylag hasonló elemeket tartalmazó csoportokra bontható, akkor - első közelítésben - ezeket a csoportokat homogénnek képzelhetjük és igy egyszerűbb feladathoz jutunk.

d) ***Több cél egyidejű figyelembevétele***

Ez sokféleképpen megoldható. A célokat kifejező függvények súlyozott összege egyetlen célfüggvényt ad. Más megközelítésben az egyes függvényekre külön korlátok

is előírhatók, amelyeket külső beavatkozással vagy automatikusan különböző módon meg lehet változtatni: így az egyes célok különböző szintű elérését valósíthatjuk meg. A *variánsok közötti döntés* -az értékelések figyelembevételével - *az ember feladata*.

e) ***Heurisztikus megoldási módszerek alkalmazása***

A matematikai modellekkel szemben két, egymásnak ellentmondó fő követelményt támasztunk: (1) minél hűebben tükrözzék a valóságot;
(2) minél egyszerűbbek legyenek.

A valósághűségre való törekvés bonyolítja a matematikai struktúrát és növeli a számításigényességet. Következésképpen a matematikai módszerek használatát végső soron a matematika és a számítógépi technika pillanatnyi fejlettsége korlátozza.

A matematikai módszerek használatának feltételei:

- a probléma numerikus formában is felírható legyen;
- az elérendő cél kifejezhető legyen jól definiált célfüggvényként;
- álljon rendelkezésre valamilyen számítási eljárás, amelynek számítás-igényessége nem lépi túl a rendelkezésre álló számítógép kapacitását, továbbá elfogadható időn belül elvezet a probléma megoldásához.

Ha a feladat a fenti feltételeknek *nem tesz eleget*, akkor az ún. *heurisztikus módszerek* alkalmazása válik szükségessé.

A heurisztikus módszer az adott probléma megoldását egy terv konstruálása útján keresi. Ez a konstruálási folyamat *lépésről-lépésre dönt a következő tervezési elemről*. Minél gondosabban, célszerűbben készítjük el előrehaladási stratégiánkat, annál jobb megoldást kapunk, de ezek az eljárások a matematikai eljárásokkal szemben az esetek többségében csak *szuboptimális* (vagy ún. kvázioptimális) megoldásokhoz vezetnek.

A heurisztikus eljárások sokkal egyszerűbbek és kevésbé igényesek, mint a matematikai optimálások. *Jelentőségük gyakorlati alkalmazhatóságukban van*. Mindegyik heurisztikus eljárással az emberi tervezési mechanizmust utánozzuk, amelynél a tervezés folyamata a gondos körültekintés ellenére is sokszor megakad. Ezekben az esetekben a tervező a tervkonstrukció egy részét lebontja, majd kijavítva újra felépíti. Sokszor az eredeti feltételeken, korlátokon is változtatva alakul ki a végleges terv.

A heurisztikus eljárás mindig tartalmaz szubjektív tényezőt, ezért az eljárás jóságát mindig gyakorlati alkalmazhatóságán lehet lemérni.

A heurisztikus eljárásokat sok változót tartalmazó, bonyolult problémák megoldására alkalmazzuk, amikor az eljárás kiépítése magában foglalja egy számítógépes program elkészítését is. Ehhez *a manuális tervezési technikát addig kell módosítani, ameddig a terv automatikus felépítése lehetségessé nem válik* (algoritmizálhatósági feltétel kielégítése).

A heurisztikus megoldó módszereknek olyan változatai is ismeretesek, amelyek jól kombinálhatók egzakt eljárásokkal. Ennek eredményeként létrehozható olyan eszköz, amely nagyobb számítógépi időigénnyel pontosabb eredményt szolgáltat. Ennek a megközelítésnek az az előnye is megvan, hogy egy gyors első megoldás rossz adatokat vagy bizonyos koncepcionális hibákat is kiszűrhet.

4.11. Termelésirányítási rendszerek ipari bevezetése

Gyakran felmerül a kérdés, hogy miért nem vesz át egy gyár kész termelésirányítási rendszert, hiszen ha saját maga fejleszt vagy fejlesztet, az igen sok pénzbe és munkába kerül. Ennek fő oka az, hogy az általános célú termelésirányítási rendszerek éppen általánosságuk miatt túl nagy felesleges terhet hordoznak (Nagyobb számítógép szükséges hozzá, nehezebb az implementálás, stb.). Ugyanakkor gyakran még ez a nagyobb rendszer sem tud bizonyos igen fontos speciális igényeket kielégíteni.

Ha egy nagyvállalat - külső segítséggel - TIR kifejlesztésére határozza el magát, ez egyben az új termelésirányítási koncepció bizonyos mértékig kollektív kialakítását is jelenti. Egy kész TIR üzemszerű használata igen sok gonddal jár, amit még a próbaüzem alatt sem észleltek. A bevezetést ezért gondosan meg kell tervezni és alaposan kell előkészíteni.

4.12. Gépipari gyártásformák, irányítási feladatok

4.12.1. Általános szempontok

A gépgyártás folyamata két, egymástól jól elkülöníthető részre oszlik: *alkatrészgyártásra* és *szerelésre*. Bár az irányítás során a kettő csak együtt kezelhető, a tantárgy keretei között csak az alkatrészgyártással, ezen belül a forgácsolási műveletek irányításával foglalkozunk. Csak megemlítjük, hogy a szerelési feladatok ütemezésére jól bevált módszerek léteznek (pl. PERT, CPM).

A TIR kimenő információi közvetlenül vagy közvetve előírják a gyártásban résztvevő minden munkadarabra, hogy azon MIKOR, HOL, MIT, és MIVEL végezzenek el. A MIKOR jelentése egy tényleges időpont, ha az irányítás közvetlen, de lehet valamilyen feltételek által meghatározott döntések eredménye is, ha az irányítás közvetett. A HOL munkahelyet, gépet jelent, a MIT technológiai műveletsorozatot, a MIVEL pedig szerszámokat, készülékeket, segédanyagokat.

Az irányítás célja: a fenti kérdéscsoport olyan megválaszolása, amely - kielégítve a környezeti, technológiai és gazdasági feltételeket - valamilyen cél(-ok) szerinti optimális termelést ad.

Az optimalitási kritérium megfogalmazása, az egzakt megoldás megkeresése rendkívül nehéz, lévén minden bonyolult rendszer eleme egy nála még nagyobb és bonyolultabb rendszernek. Elméletileg már a nemzetgazdaság irányításában kellene figyelembe venni minden kis részletet, de ez természetesen lehetetlen.

Közel optimális megoldáshoz juthatunk azonban vertikális dekompozíció segítségével, amelynek lényege, hogy a feladatot különböző szintekre bontjuk, minden szinten egyetlen optimálási feladatot oldunk meg bizonyos célok és feltételek mellett és az egyes szintek optimális megoldásai szerepelnek a következő, eggyel alacsonyabb szint feltételeinek kialakításában.

Egy gyakori példán szemléltetve:

Szint	Feltételek	Optimálandó változók
Éves terv	eszközök technológia	MIT
Havi terv	eszközök technológia MIT	HOL
Napi terv	eszközök technológia MIT HOL	MIKOR

4.12.2. A gyártásformákat jellemző tényezők

Az optimálási feladatok megszokott formája, hogy vannak bizonyos feltételek, amelyeket az optimálandó változóknak ki kell elégíteniük és van egy célfüggvény, amelynek szélsőértékét kell meghatározni az említett feltételek kielégítése mellett. Ilyen feladatok megoldásával a Matematikai Programozás tudománya foglalkozik. A feltételek és célok matematikai formái azonban olyan sokfélék, hogy egy általános termelésirányítási feladat megoldásához szükséges apparátus felöleli a Matematikai Programozás teljes eszköztárát. A modellezés során a megoldhatóság figyelembevétele korlátként jelentkezik: *egy általános célú programcsomag csak kisméretű konkrét feladatokat tud megoldani*.

A nagyméretű feladatok megoldására írt algoritmusoknak alkalmazkodniuk kell az adott, *tényleges adatstrukturákhoz*, így lehet megsokszorozni az eredeti program(-rendszer) teljesítőképességét.

A gépipari termelésirányítási rendszerek jellemző tényezői:

Feltételek:

a) Környezeti feltételek:

- gépállomány;
- raktárkapacitások;
- munkaerő.

b) Technológiai feltételek:

- az egyes gyártási fázisok idősorrendje;
- időigények és időkorlátok;
- minőségi feltételek.

c) Üzemi és gazdasági feltételek:

- vállalati terv;
- értékesítési viszonyok;
- gyártási határidők;
- raktárkészletek;
- importkorlátozások;
- exporttámogatások;
- a termelés illesztése a gyári környezethez.

Célok:

- maximális nyereség;
- minimális önköltség;
- legrövidebb átfutási idő;
- maximális termelékenység;
- <ezek valamilyen kombinációja>;
- maximális fedezet;
- minimális készlet;
- határidők betartása;
- ütemes szállítás;
- maximális bruttó kibocsátás;
- a berendezések egyenletesen maximális leterhelése;
- piacok megtartása, stb.

Az egyes feltételek lehetnek *egyenlőség*, *egyenlőtlenség* vagy *logikai* típusúak, a függvények *lineárisak*, *nemlineárisak*, a változók *folytonosak*, *diszkrétek* vagy *valószínűségi* változók.

4.12.3. Néhány gyakran előforduló gyártásforma

A következőkben néhány, hazai viszonyok között alkalmazható modellt említünk.
Idevágó *alapfogalmak*:

Sorozatnak vagy *tételnek* nevezzük az azonos, egymást minden szempontból helyettesítő és a gyártás során együtt haladó munkadarabok halmazát. Egy sorozat vagy egy tétel speciálisan egyetlen munkadarab is lehet.

A munkadarabon *műveleteket* kell elvégezni, amelyek sorrendjét a technológiai dokumentáció (műveletirányítási lap, műveletterv) írja elő. Ez az ún. *technológiai sorrend.*

Egy *elvégzendő munka* az egy sorozaton (tételen) végrehajtandó műveletek együttesét jelenti, amelyek sorrendjét a technológiai sorrend rögzíti.
Alkalmanként a *munka* szó magát a munkadarabot (vagy a sorozatot) is jelölheti, ami a szöveg összefüggéseiből kitűnik.

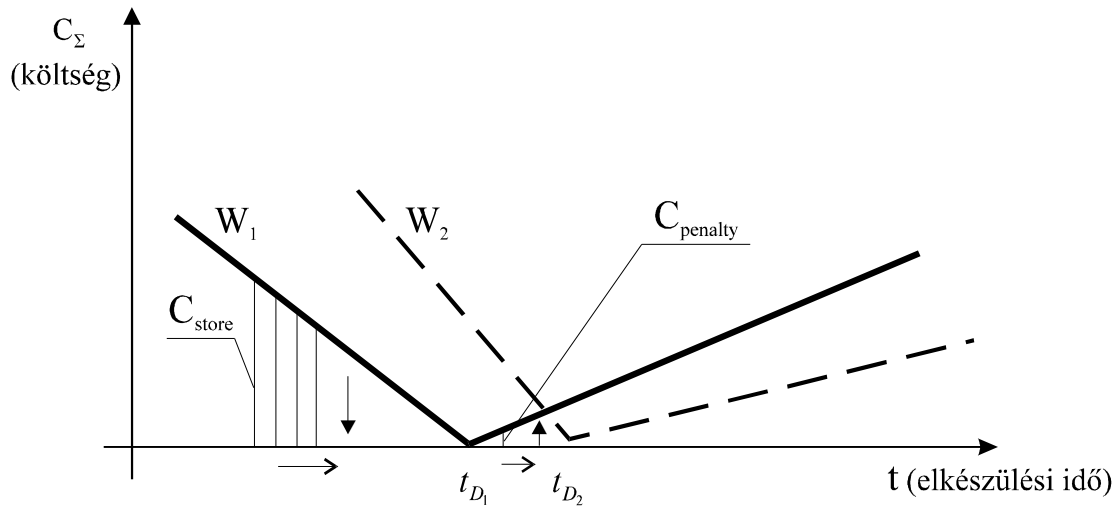
Az ütemezési feladatoknál a *gép* jelenthet egy *konkrét gépet* vagy egy (ekvivalens gépekből álló) *homogén gépcsoportot*. Általában, amikor *m* számú *gépről* van szó, akkor ez *m* különböző homogén gépcsoportot jelent.

A munkák nem feltétlenül érintik a rendelkezésre álló összes gépet a gyártási műveletek során. A munkák *technológiai útvonala* jelenti azt az utat, amelyet az egyes tételek (sorozatok) a műveletek előírt technológiai sorrendje szerint a gépeken be kell, hogy járjanak a teljes legyártásig.

a) 1-gépes termelés

Adott egy gép, vagy 1 géppel helyettesíthető homogén gépcsoport. Ezen a gépen *n* munkát kell elvégezni, amelyeknek gépidőigényét ismerjük. Feladatunk a munkák elvégzésének optimális sorrendjét meghatározni a következő esetekben:

a) Adottak a munkák elvégzési határidői és a gépidőigények. Ha egy munka határidőn túl készül el, az okozott kárt a késéssel arányos *kötbérrel* modellezhetjük, amely az egyes munkákra - fontosságuktól függően - más és más. Ha határidő előtt készül el valami, ez pedig *raktározási költséget* okoz, amely arányos azzal, hogy mennyivel előbb készül el a munka.



4.1. ábra: Költségösszetevők alakulása az elkészülési idő függvényében

Az optimalitási kritérium a minimális költség, amely ideális esetben, amikor mindegyik munka pontosan határidőre készül el, *zérus*. A feladat visszavezethető a *hozzárendelési feladatra*.

b) Az egyes munkák elvégzésének határidői nem szorosak, viszont a munkák közötti átállás ideje nem elhanyagolható. Ez az átállási idő attól függ, hogy melyik munka után melyik másik munka következik és ezek az idők különbözőek. Nem mindegy tehát, hogy milyen sorrendben követik egymást a munkák, ha a célunk az, hogy az *átállási idők összege* minimális legyen.
Ha most az egyes munkákat *városoknak*, az átállási időket *utazási időknek* tekintjük, máris eljutottunk az ún. ***utazó ügynök feladathoz***. Az ügynök körútja megfelel a munkák elvégzésének. Az utazó ügynök feladatot később részletezzük.

b) Sorozatgyártás a terv teljesítésére

Raktárra termelnek, szoros határidők nincsenek. A tervidőszakra kijelölt munkát *egyenletesen kell elosztani időben*, valamint *a gépek és a munkások* között. Az egyenletes terhelés elérése céljából az optimálás változói közé a forgácsolási paramétereket is be kell vonni. *Az irányítás célja: a tervfeladatok teljesítése minimális költséggel*.

A költségekben számításba vesszük:

- a megmunkálás közben felhasznált energiát; továbbá
- a szerszám-elhasználódást, amelyek függenek a forgácsolási paraméterektől;
- a gép amortizációs költségét, amely csak közvetve, a megmunkálási idő révén függ a forgácsolási paraméterektől;
- a kapacitáskorlátokra nézve
  - a gépek és
  - a munkaerő

kapacitását.

Az önköltségszámításban a $C_{E,i}$ energiaköltséget, a $C_{s,i}$ szerszámköltséget és a $C_{G,i}$ géplekötési költséget vesszük számításba (az i index valamely aktuális munka sorszámát jelzi):

$$min \sum_{i=1}^{n} l_i \left[C_{E,i}(d_i, f_i, v_i) + C_{S,i}(d_i, f_i, v_i) + C_{G,i}(d_i, f_i, v_i) \right]$$

$$\sum_{i=1}^{n} l_i t_{i,j}^{G} \le T_j^{G} \qquad j = 1,2,...,m \qquad \text{(gépkapacitás korlát)}$$

$$\sum_{i=1}^{n} l_i t_{i,k}^{M} \le T_k^{M} \qquad k = 1,2,...,M \qquad \text{(munkaerő-kapacitás korlát)}$$

ahol: n - a munkák száma; m - a gépek száma; M - a munkások száma; l_i - az i-edik munkából elkészítendő darabszám; $d_i, f_i v_i$ - az i-edik munka forgácsolási paraméterei (rendre: fogásmélység, előtolás, forgácsolási sebesség); $C_{E,i}(d_i, f_i, v_i)$ - az i-edik munka felhasznált energiaköltsége; $C_{S,i}(d_i, f_i, v_i)$ - az i-edik munka felhasznált szerszámköltsége; $C_{G,i}(d_i, f_i, v_i)$ - a megmunkáló gép lekötésének költsége; $t_{i,j}^{G}(d_i, f_i, v_i)$ - az i-edik munka gépidőigénye a j-edik gépre; $t_{i,k}^{M}(d_i, f_i, v_i)$ - az i-edik munka időigénye a k-adik munkásra; T_j^{G} - a j-edik szerszámgép rendelkezésre álló munkaideje (időalapja); T_k^{M} - a k-adik munkás rendelkezésre álló munkaideje (időalapja).

A feladat megoldását nehezíti, hogy a C_E, C_S, C_G függvények nemlineárisak. Megoldási módszerként a *geometriai programozás* alkalmazható. Ez a módszer a matematikai programozás eszköztárának egy eleme és lényegében a számtani és a mértani közép között fennálló egyenlőtlenség általánosításán alapszik. *A geometriai programozás általánosított polinom alakú célfüggvények minimalizálásával (maximalizálásával) foglalkozik, amikor a korlátozó feltételek is általánosított polinom alakú egyenlőségek, vagy egyenlőtlenségek.*

4.13. Integrált gyártórendszerek termelésirányítása (IGYR)

Az IGYR-ekben a termék általában palettára szerelve járja végig a technológia által előírt sorrendben a munkahelyeket.

Ebből következik, hogy a megmunkálások közti tárolást - tekintve, hogy a tárhelyek száma korlátozott - figyelembe kell venni az irányításban.

Ugyanilyen korlátozás érvényes az egyes gépek szerszámtáraira is. Az ilyen magas fokon automatizált gyártórendszerek beruházási és fenntartási költsége igen nagy, ezért a rendszer maximális kihasználtságát, lehetőleg szakadatlan működését kell megvalósítani. Az irányítás három szintjén ezeket a szempontokat az alábbi módon kell figyelembe venni:

a) ***Tervezéskor*** a *legjobb termékválaszték* meghatározása és a *hosszútávú terv lebontása havi tervekre* a két legfontosabb feladat.

b) ***Ütemezéskor***:
- a *haviterv lebontása heti-* vagy *dekádtervekre*;
- a *munkák géphezrendelése*;
- a *gyártás-előkészítés* és *anyagellátás megszervezése*.

c) ***Programozáskor***: a már előkészített, raktáron lévő munkadarabok közül kell kiválasztani a soronkövetkező műszakban megmunkálásra kerülőket és azok megmunkálási sorrendjét elő kell írni.

4.14. Matematikai modellek a termelésirányításban

A termelésirányítással kapcsolatban számos matematikai feladat adódik az *irányítás szintjétől* és a *technológiai adottságoktól* függően. Nyilván minél szűkebb az a terület, amelynek tevékenységét vizsgáljuk, annál pontosabb modellre van szükség. Megfordítva, ha nagyobb egységet vizsgálunk, nem vehetünk számításba minden részletet.

Néhány fontosabb modell ill. módszer:

- a lineáris programozás;
- diszkrét programozási feladatok, ezen belül pl. a hátizsák-feladat;
- az utazó ügynök feladata;
- a hozzárendelési feladat;
- heurisztikus termelésprogramozási módszerek.

4.14.1. Lineáris programozás

Egy gyár bizonyos időszakra szóló termelési feladatának meghatározása a lineáris programozás egyik tipikus alkalmazási területe. Meg kell határozni azokat a mennyiségeket, amelyeket a gyár termékeiből elő fogunk állítani egy adott tervidőszakban. Korlátozó tényezőt jelentenek a *rendelkezésre álló erőforrások*. A különböző termékek egységnyi mennyiségének előállításához szükséges erőforrások ismertek. További megkötés a piaci elvárások rendszere (az adott időszakban teljesítendő szerződések és más követelmények), valamint a gyártmánystruktúrából származó feltételek. Ilyen módon bonyolult feltételrendszer jöhet létre. A termelési feladat egészét tekintve nem haladhat meg bizonyos korlátokat, de egyes termékek mennyiségére külön alsó és felső korlátok is adódhatnak. Ismert továbbá a termékek valamilyen értékelése. A célunk az *adott feltételek mellett minél nagyobb értékű termelési feladatot meghatározni*. Az említett értékelés lehet a termék ára vagy a terméken jelentkező haszon. A megfelelő értékelést az adott körülményeknek megfelelően kell kiválasztani. Fontos megjegyezni, hogy a *különböző értékelések más-más optimális megoldásokhoz vezetnek.*

Egy másik példa a lineáris programozás alkalmazására a következő. Az egyes termékekre *alternatív technológiák* ismertek. Egy-egy termék különböző technológiai folyamatai a berendezéseken különböző kapacitás-lekötéseket jelentenek és megvalósítási költségük is eltérő. A feladat: a technológiai folyamat-alternatívák olyan kiválasztása, hogy a kapacitáskorlátokat ne lépjük túl és az összköltség minimális legyen.

Egy jellemző termelésirányítási feladat:

Készítsünk havi tervet egy gyáregység részére, mely n féle különböző terméket állít elő. A termeléshez rendelkezésre álló erőforrások (gépek, áram, víz, ember, stb.) száma legyen m. A j-edik termék egységének előállításához az i-edik erőforrásból a_{ij} mennyiség szükséges. Az egyes erőforrásokból rendelkezésre álló mennyiség legyen b_i. A j-edik termék egységnyi mennyiségén jelentkező haszon legyen c_j. Az adott hónapban x_j mennyiséget állítsunk elő a j-edik gyártmányból.

Ekkor a legnagyobb hasznot hozó havi tervet a

$$max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i (i = 1,2,...,m)$$

$$x_j \ge 0 (j = 1,2,...,n)$$

lineáris programozási modell adja, ahol c_j, a_{ij}, b_i konstansok, x_j változó.

(Természetesen más feltételek is figyelembe vehetők, a feladat lényege nem változik.)

4.14.2. Diszkrét programozási problémák

Ha egy optimumszámítási problémában az ismeretlenek egy része, vagy az összes ismeretlen csak diszkrét értékeket vehet fel, *nemfolytonos* modellről beszélünk. A nemfolytonos modellek - értelemszerűen - "tiszta diszkrét" és "vegyes" típusúak lehetnek.

Csupán matematikai szempontból nézve, szinte minden folytonos feladatnak felírható az integer megfelelője. Az integritás előírására gyakran szükség is van, amikor kis darabszámú, nem osztható termék esetén a folytonos optimum kerekítésével nyerhető megoldás távol eshet az integer optimumtól.

Az irányításban szükséges döntések két csoportra, *minőségi* és *mennyiségi* döntésekre oszthatók. A folytonos változók a mennyiségi döntések jellemzésére alkalmasak. Ha a modellben minőségi döntések hatásait is figyelembe kell venni, akkor szükségessé válik diszkrét változók bevezetése is, amelyek csak előre meghatározott értékeket vehetnek fel.

Ha mennyiségi és minőségi döntések együtt fordulnak elő a feladatban, *vegyes diszkrét* modellekhez jutunk. Ilyen jellegű modellt kapunk, ha egyes tevékenységeket meghatározott szintek valamelyikén végezhetünk, de a szintek mennyiségi jellemzői bizonyos határok között mozoghatnak. Ekkor a diszkrét változók az optimális szintek kiválasztását szolgálják, míg a folytonos változók - a szinteken belül - a mennyiségi paramétereket állítják be.

A) A hátizsák feladat

A diszkrét programozási problémák közül a legegyszerűbb az ún. *hátizsák feladat*. Az elnevezés onnan ered, hogy a problémát meg lehet fogalmazni a következő módon: egy turista különböző tárgyakat kíván magával vinni egy útra. Ezek súlya együttesen azonban már meghaladja teherbíró képességét. Így kénytelen szelektálni. Minden tárgynak ad egy értéket.

A válogatást úgy akarja elvégezni, hogy a súlykorlátozás betartása mellett a magával vitt tárgyak összértéke maximális legyen.
A "hátizsák" feladat matematikai megfogalmazása a következő:

$$\sum_{j=1}^{n} a_j x_j \le b$$
$$x \in B_n$$

$$H = \sum_{j=1}^{n} c_j x_j = \max$$

ahol B_n - a bináris számok halmaza $\forall a_j, \forall c_j$ és b - természetes szám; továbbá n tárgyból áll a hátizsák-rakomány, amely összességében a b súlynál nem nehezebb; a_j - a j-edik tárgy súlya, c_j az aktuális használati értéke. Az x_j aszerint 0 vagy 1, hogy az illető tárgy bekerül-e a hátizsákba, vagy sem.

A feladat gyakorlati alkalmazására egy példa: n különböző méretű csődarabra van szükségünk. Az egyes darabok mérete $a_j (j = 1,...,n)$. Egy b hosszúságú csövet vágunk szét. Kérdés, mely darabokat leggazdaságosabb a csőből előállítani. Legyen az egyes darabok valamilyen szempont szerinti értéke $c_j (j = 1,...,n)$. Ha az anyagveszteség minimalizálására törekszünk, akkor $c_j = a_j$. Vezessük be az $x_j (j = 1,...,n)$ döntési változókat. Az x_j aszerint 1 vagy 0, hogy a megfelelő méretű darabot levágjuk-e vagy sem.
Így a feladat:

$$max \sum_{j=1}^{n} c_j x_j$$
$$\sum_{j=1}^{n} a_j x \le b$$
$$x_j = 0 \,|\, x_j = 1 (j = 1,...,n)$$

$$\forall a_j \ge 0 \ \& \ \forall c_j \ge 0 .$$

A hátizsák-feladat alkalmas előre ismert *szűk keresztmetszet* vizsgálatára. Bizonyos munkák közül választhatunk, amikor a termelési feladatot össze akarjuk állítani. A j-edik munkán a haszon legyen c_j és a munka igénye a szűk keresztmetszetet jelentő erőforrásból

a_j. Az erőforrásból rendelkezésre álló mennyiség legyen *b*. Ekkor a fenti feladat optimális célfüggvényértéke tájékoztatást ad az elérhető legnagyobb haszonról.

A hátizsák-feladatok általánosításaként, több feltétel esetén a **c** ill. **b** *n* ill. *m* dimenziós vektorokkal és az **A** *m*x*n* méretű mátrixszal, mint együtthatóval a

$$max\mathbf{cx}$$
$$\mathbf{Ax} \le \mathbf{cx}$$
$$x_j = 0 \,\big|\, x_j = 1 (j = 1,...,n)$$

alakú problémát nyerjük.

A vegyes diszkrét feladatok esetében a változók egy csoportjától csak *nemnegatívitást* követelünk meg:

$$max(\mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y})$$
$$\mathbf{Ax} + \mathbf{By} \le \mathbf{b}$$
$$\mathbf{x} \ge \mathbf{0}$$
$$y_j = 0$$
$$y_j = 0 \,\big|\, y_j = 1, j = 1,...,n$$

A megoldási módszereknek két csoportja alakult ki: a *Benders*-dekompozíció és a *Branch* & *Bound* (szétválasztás & korlátozás) típusú eljárások.

B) Az utazó ügynök feladata

Adva van *n* számú város. Jelöljön ezek közül kettőt *i* és *j*. Ekkor ismert ezek c_{ij} távolsága. Feladat egy adott városból kiindulva a legrövidebb úton bejárni valamennyit úgy, hogy mindegyiket csak egyszer érintsük és utunkat a kezdőpontban fejezzük be. Ugyanehhez a problémához jutunk, ha *egy gépen egy adott időszakra* (pl. egy műszakra) *úgy akarjuk az elvégzendő munkák sorrendjét meghatározni, hogy az átállítási idők összege minimális legyen*. Ebben az esetben c_{ij} az *i*-edik és *j*-edik munkadarab közötti átszerszámozási idő.

Több kísérlet történt arra, hogy az alapmodellt további feltételekkel egészítsék ki. Az egyik ilyen változat az, amikor a körüljárást úgy kívánjuk megvalósítani, hogy egyetlen városból kiindulva és oda ismételten visszatérve több kisebb körút során jutunk el valamennyi városba. Továbbá nemcsak egyszerű utazásról van szó, hanem az ügynök csomagokat is szállít a városokba. Az egy részkörúton szállított árú összsúlya korlátozva van.

Ütemezési szempontból ez annak az esetnek felel meg, amikor az átszerszámozási időkön kívül a megmunkálási időket is figyelembe vesszük.

Ugyanez a feladat eredeti értelmezésben is felmerülhet a következőképpen. Egy nagy kiterjedésű gyárban sok helyen felhasználják az egyik üzemrész által előállított alkatrészeket. Ekkor meghatározott útvonalakon menetrendszerűen célszerű közlekedtetni a szállítóeszközöket. Az optimális útvonalakat - és így a szállítóeszközök optimális kihasználásának módját - a fenti probléma megoldásával nyerhetjük.

Egy másik alternatíva, amikor határidőket veszünk figyelembe, azaz bizonyos városokat egy megadott időpontig el kell érni. Ez lehetőséget ad arra, hogy a berendezés más gépek ütemezéséhez alkalmazkodni tudjon, azaz több berendezésre együttes programot tudjunk készíteni.

A probléma egyik lehetséges matematikai megfogalmazása a következő:

Tegyük fel, hogy az ügynök az i_1 városból indul, innen i_2-be megy, stb. Így

$$P = (i_1, i_2, ..., i_n, i_{n+1} = i_1)$$

egy körbejárás. Feladatunk az összes lehetséges körbejárások közül a minimális költségű kiválasztása, azaz

$$\min_P \sum_{j=1}^{n} c_{i_j i_{j+1}}$$

mivel az i_1-ből i_2-be való utazás költsége $c_{i_1 i_2}$, az i_2-ből i_3-ba való utazás költsége $c_{i_2 i_3}$, stb., és ezek nyilván összegeződnek.

Amikor több kisebb körutat valósítunk meg, akkor ezek száma legyen *m*, az egyes körbejárások pedig:

$$P_1, P_2, ..., P_m$$

Ezekben az utakban valamennyi városnak szerepelnie kell, azaz

$$\bigcup_{k=1}^{m} P_k = \{1, ..., n\}. \qquad (1)$$

Legyen a körutak közös kiinduló- és végpontja i_0. Nyilván nincs értelme, hogy egy városba többször is elmenjünk, mert ezzel csak a költségeket növelnénk feleslegesen. (Az

ütemezési esetben ez azt jelentené, hogy ugyanazt az alkatrészt többször akarnánk legyártani.) Tehát a különböző körutaknak i_0-n kívül közös csúcsuk nem lehet, azaz

$$P_l \bigcap P_k = \{i_0\} \text{ és } l \neq k . \tag{2}$$

Minden *j*-edik városba q_j lehet a megmunkálási idő vagy a szállított alkatrészek mennyisége a szállítóeszköz szempontjából mérve.

Legyen *G* az egy körútban szállítható mennyiség (a műszak hossza, ill. a gépkocsi kapacitása). Így a

$$\sum_{j \in P_k} q_j \leq G \qquad \text{(k=1,...,m)} \tag{3}$$

feltételhez jutunk.

Jelölje D_k a P_k halmazon az utazó ügynök feladat optimális megoldását. Ekkor feladatunk a

$$\min_{(P_1,\ldots,P_m)} \sum_{k=1}^{m} D_k$$

szélsőérték meghatározása az (1), (2) és (3) feltételek mellett.

C) A hozzárendelési probléma

Adott *n* férfi és ugyanennyi nő, akik valamennyien házasságot szeretnének kötni. Mivel a bigámiát a törvény tiltja, így minden férfihoz pontosan egy nő tartozhat és megfordítva is ez áll. Legyen c_{i_j} az a hozomány, amit az *i*-edik nő családja ad, ha a nőt a *j*-edik férfi veszi el feleségül. A hozzárendelési feladat célja lehet ez esetben: megkeresni azt a házasítást, amely mellett a hozományok összege minimális.

Hosszabb időszak gyártási feladatainak szétosztása több műhely (gép) között hasonló hozzárendelési problémára vezet. Egy adott tervidőszakban *n* tétel (sorozat) gyártását kell szétosztani ugyanennyi műhely (gép) között. Ezek közül mindegyik csak egy tétel gyártását köteles elvégezni és egy tétel gyártását csak egy műhelynek (gépnek) lehet kiadni. Legyen c_{i_j} az a költség, amellyel az *i*-edik tétel gyártása jár a *j*-edik műhelyben (gépen). Ismertnek feltételezve ezeket a költségeket, keresni lehet azt a szétosztást, amely mellett a gyártási összköltség minimális.

Valamennyi lehetséges variációt nem célszerű megvizsgálni, hiszen csupán 10 gyártandó tétel esetén is ez 3628000 lehetőséget jelent.

A feladat matematikai megfogalmazásához vezessük be az x_{i_j} segédváltozót. Ennek értéke legyen 1, ha az *i*-edik tétel gyártását a *j*-edik műhelyre (gépre) osztjuk ki, egyébként pedig *0*. Így a feladat az alábbi, összköltségre vonatkozó célfüggvénnyel és korlátozó feltételekkel adható meg:

$$min\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij}$$

$$\sum_{j=1}^{n}x_{ij}=1 \qquad (i=1,2,...,n)$$

minden *j* sorszámú gyártandó tételhez egy és csakis egy műhely (gép) tartozik

$$\sum_{i=1}^{n}x_{ij}=1 \qquad (j=1,2,...,n)$$

minden *i* sorszámú műhely (gép) egy és csakis egy tételt köteles gyártani.

A feladat megoldására - még nagy méretek esetén is - hatékony eljárás az ún. "Magyar módszer".

4.14.3. Ütemezési problémák és megoldási lehetőségeik

A gyakorlati tapasztalatok, amelyeket a felhasznált termelésprogramozási módszerekre vonatkozó nemzetközi szakirodalom is alátámaszt, azt mutatják, hogy a gyártási műveletek ütemezésére eddig ismert egzakt megoldási módszerek csak kis rendszerekre érvényesek és nagyon számításigényesek. Így ipari méretű problémák kezelését ezekkel meg sem szabad kísérelni. Ez a magyarázata, hogy a gyakorlatban *heurisztikus eljárásokat* alkalmaznak nagyméretű ipari feladatok megoldására. (A heurisztikus módszer itt ismert egzakt eljárások felhasználását jelenti új problémák megoldására, figyelembe véve a termelő üzembeli empíriát, más szavakkal *"trial and error ≡ kísérlet és hiba"* módszere.)

A termelésprogramozási feladat általánosan a következőképpen fogalmazható meg: *Adott a gyártó műhelyben **n** számú elvégzendő munka és a gyártáshoz **m** gép áll rendelkezésre. Meg kell határozni a munkáknak azt a sorrendjét a gépeken, amely bizonyos előre megadott szempontok szerint optimális*. Például aszerint, hogy minimális legyen a munkák átfutási ideje vagy előírt határidőre készüljenek el a munkák, vagy minimális legyen a gyártási összköltség, stb.

Az általános termelésprogramozási feladatkörön belül egy konkrét gyártásütemezési problémát matematikai tárgyalás szempontjából a következő főbb jellemzőkkel lehet behatárolni:

a) ***A munkák beérkezési mintájával.*** Ha egyidejűleg *n* munka érkezik a géphez, amely szabad és azonnal tud fogadni munkát, akkor az ütemezési probléma *statikus*. Ha a munkák megszakításokkal érkeznek a géphez, sztochasztikus sorrendben és a gép foglalt is lehet, akkor az ütemezési probléma *dinamikus*.

b) ***Azoknak a gépeknek a számával***, amelyekre a munkákat ütemezni kell. A *két* és *háromgépes* ütemezési feladatok ún. *kisfeladatok*, amelyekre még ismert egzakt megoldás. A többgépes ütemezési feladatok az ún. *nagyfeladatok*, amelyekre egzakt megoldás nem ismert.

c) A munkáknak a ***gépeken előírt technológiai útvonalával***. Ahol adott a technológiai útvonal és ez minden munkadarabra nézve azonos, ott *egyutas* gyártásütemezési feladatokról van szó. Ha a munkadaraboknak a műhelyben eltérő technológiai útjaik vannak, a feladat *többutas*.

d) ***Az ütemezés optimumkritériumával***, amely az ütemezési hatékonyság kiértékelésének alapja.

Megjegyezzük, hogy az üzemi gyakorlatban tipikusan a dinamikus ütemezési problémát felvető nagyfeladatok a jellemzőek. Az előadásvázlatban - terjedelmi okokból - csak az egyutas gyártásütemezési feladatok megoldására szolgáló heurisztikus módszerekkel foglalkozunk.

Feladat: *n* számú munkát *m* számú különböző gépen kell elvégezni. Minden egyes munkának egy és ugyanaz a technológiai útvonala. Ez az útvonal, valamint a műveleti idők előre adottak (manuális számítás alapján, vagy valamely CAPP rendszer kimenetén). A gyártásütemezés célja lehet - egyebek között - a munkák olyan sorrendjének meghatározása, amely mellett az *átfutási idő*, azaz az első munka gépre kerülésétől az utolsó munkadarab megmunkálásának befejezéséig terjedő időintervallum, *minimális*.

Az angol nyelvű szakirodalomban az *egyutas* problémákat a "*flowshop*" elnevezés alatt tárgyalják. (A gyakorlatban ilyen pl. a fogaskerekek gyártásának menete, amikor az egyes munkadarab-sorozatok azonos típusú darabokból állnak. A *flowshop* típusú ütemezési feladatokat további két osztályba sorolják, ezek az ún. "*passing*" és "*no-passing"* feladatok. (Passing: a munkák az egyes gépeken előzhetik egymást; no-passing: előzés nem lehetséges)).

Célszerű bevezetni ezekre a megfelelő magyar terminológiát: az *egyutas előzéses* és az *egyutas előzés nélküli* ütemezés fogalmát. Az előzéses esetre példa: különböző méretű fogaskerekek gyártása, amikor *megmunkálásközi raktárak* vannak, és így lehetővé válik, hogy egy munkadarab-sorozat, amely az *i*-edik gépen első volt, a *j*-edik gépen pl. ötödikként kerüljön megmunkálásra. Az előzés nélküli eset tipikus példája a futószalag.

A feladat megoldására bemutatandó módszerek *közelítő eljárások*, mert elhanyagolják a gépek közötti anyagmozgatási vagy tranzitidőt, a gépbeállítási (előkészületi) időket a műveleti időben veszik figyelembe és nem engednek meg átlapolást a műveletek között.

(a) Az egyutas, előzés nélküli ütemezési feladat

Legyen a kitűzött cél: *a munkák átfutási idejének minimalizálása*. Ez más szavakkal az egyes gépek állásidejének minimalizálását is jelenti.

A munkadarabok meghatározott sorrendben kerülnek a gépekre. Az első munkadarab azonosítója legyen j_1, a másodiké $j_2,...,$ az utolsóé j_n. Jelöljön ebből $S_{j_s} = (j_1,...,j_s)$ egy kezdeti részsorozatot, $T(S_{j_s},i)$ pedig legyen az S_{j_s} sorozat átfutási ideje a technológiai útvonal szerinti első *i* számú gépen.

Tegyük fel, hogy az S_{j_s} részsorozat után a j_{s+1} azonosítójú munkát végezzük el. Az átfutási időkre az alábbi rekurzív formulát kapjuk:

$$T(S_{j_s+1},i) = max\left[T(S_{j_s},i),T(S_{j_s+1},i-1)\right]+t_{j_s+1,i},$$

ahol: $T(0,i)=T(S,0)=0$ minden *S*-re és *i*-re, továbbá $t_{j_s,i}$ a j_s-edik alkatrész megmunkálási ideje az *i*-edik gépen. Így a flowshop-probléma egy tetszőleges *P* munkadarab-sorrendre:

$$\min_P T(P,m)$$

alakban írható fel. Az elmondottakra vegyünk egy példát (4.2. ábra).

Megjegyzés: $j_1, j_2,..., j_n$ helyett *1,2,...,n* is írható, ekkor

$$t_{j_s,i} \equiv t_{k,i}$$

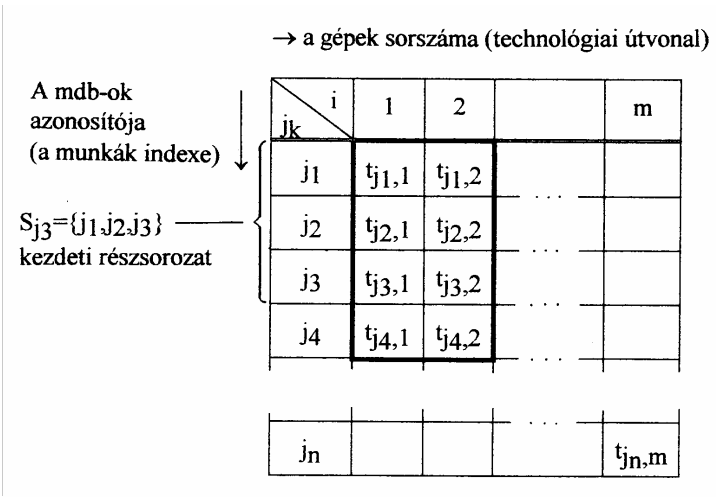
amelyre

$$1 \le k \le n$$

és

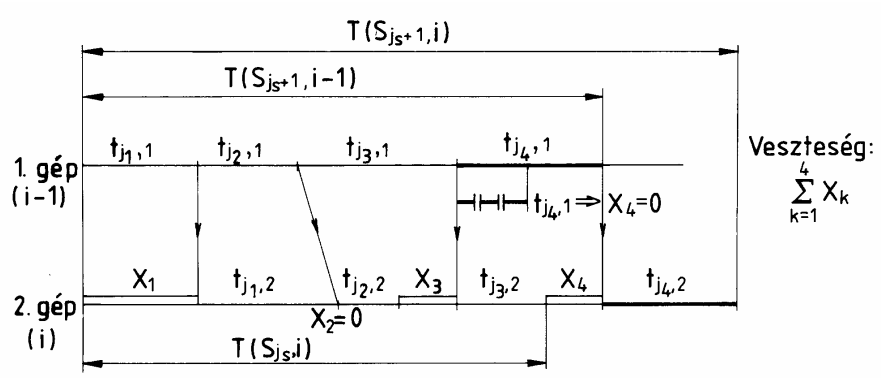
$$1 \le i \le m$$

(*n* számú munka *m* számú gépen) $t_{j_s,i}$ - a j_s-edik alkatrész megmunkálási ideje az *i*-edik gépen.



4.2. ábra: Indexelési technika az egyutas, előzés nélküli ütemezési feladathoz

Legyen most $i=2$; $j_s = j_3$; $j_{s+1} = j_4$; $S_{j_s} = S_{j_3} = \{j_1, j_2, j_3\}$.
Rajzoljuk fel az 1. és 2. gép terhelésére vonatkozó ún. *Gantt*-diagramot (4.3. ábra).



4.3. ábra: Gantt-diagram az 1. és 2. gép terhelésére vonatkozóan

Ha $i=2$, $j_s = j_3 = 3$, $S_{j_s} = S_{j_3}$; $S_{j_s+1} = S_4$, akkor a 4.3. ábra alapján látható, hogy:

$$T(S_4,2) = max[T(S_3,2), T(S_4,1)] + t_{4,2}.$$

Általánosan is igaz, hogy

$$T(S_{j_s+1},i) = max[T(S_{j_s},i), T(S_{j_s+1},i-1)] + t_{j_s+1,i},$$

amint azt korábban feltettük. A bizonyítás *teljes indukcióval* könnyen elvégezhető.

A feladat kezelhetősége erősen függ a berendezések számától. $m=2$ esetén egy igen egyszerű módszer, a *Johnson-algoritmus* használható fel a feladat egzakt megoldására, amely $m=3$ esetén is bizonyos esetekben alkalmazható. Az egyutas előzés nélküli feladat megoldására $m>2$ esetére számos heurisztikus módszert dolgoztak ki. A legfontosabbak *Palmer, Campbell-Dudek-Smith, Page* és *D.G.Dannenbring* nevéhez fűződnek.

(b) Johnson algoritmusa

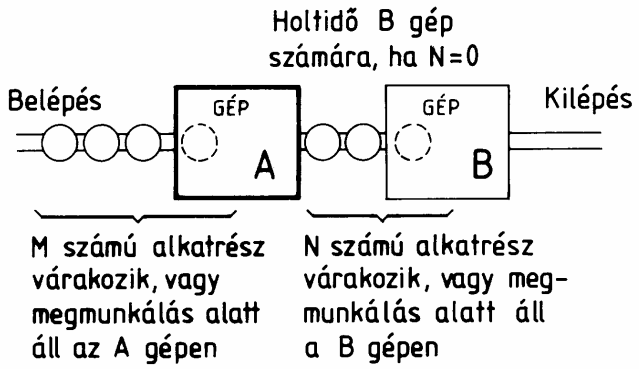
Johnson algoritmusa lehetővé teszi az olyan sorbarendezési (itt egyutas, előzés nélküli ütemezési) feladatok megoldását, amelyekben n számú munkát kell két, egymást követő munkahelyhez (géphez, berendezéshez) hozzárendelni. Egyes speciális esetekben a feladat $m=3$-ra is kiterjeszthető. Bár $m>3$ esetén nem használható, rendkívül fontos eljárás, mert

alapját képezi a nagyobb feladatokhoz kifejlesztett heurisztikus módszereknek. Tekintsük a 4.4. ábrát.

Az ábra két gépből álló gyártósejtet mutat, amelyen a megmunkálási folyamat a két gépnek megfelelően legyen a következő:

$A \rightarrow$ furatesztergálás (nagyolás),
$B \rightarrow$ furatesztergálás (simítás).

Tegyük fel, hogy a furatmegmunkálás két (különböző pontosságú) gépre való bontása a tűrések miatt indokolt.



4.4. ábra: Két gépből álló gyártócella modellje a *Johnson*-algoritmus levezetéséhez

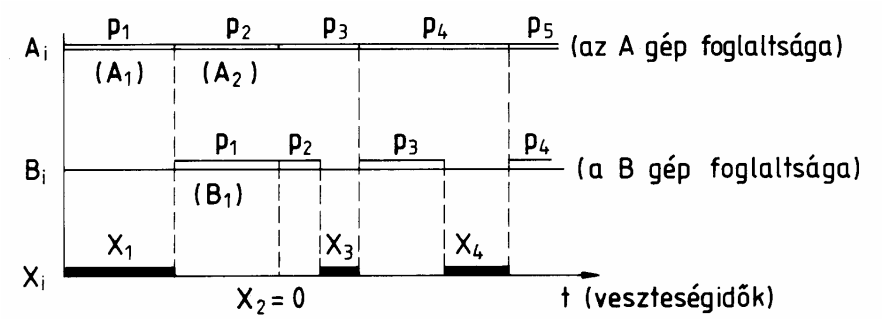
A kétféle megmunkálási műveletet n számú különböző alkatrészen végzik el, amelyek jellege hasonló, de méreteik különbözőek, továbbá tetszőleges sorrendben jelentkezhetnek. Következésképpen az átfutási idő egy tetszőleges, de választás után rögzített sorozatra igen eltérhet ugyanannak a munkadarab sorozatnak egy más sorrendben gyártott variánsáétól.

Jelöljük most A_i-vel és B_i-vel azt az időt, amelyet az i-edik alkatrész megmunkálására fordítanak az A és B gépen. *A feladat a B gép állásidejének minimalizálása.* Vagyis olyan $p_1, p_2, ..., p_i, ..., p_n$ sorrendet akarunk megállapítani, amelynél a p_j és a p_{j+1} alkatrészek simító furatesztergálása közötti állásidők összege a legkisebb, az egymást követő j értékekre számítva az összeget. (p - a "*part*" = alkatrész szó kezdőbetűje).

Nevezzük T-nek az összes időt, amely eltelik az első alkatrész furat-nagyolásának megkezdése és az utolsó alkatrész furat-simításának befejezése között. Legyen X_i a holtidő a p_{i-1}-edik feladat befejezése és a p_i-edik megkezdése között. A 4.5. ábra alapján felírhatjuk:

$$T = \sum_{i=1}^{n} B_i + \sum_{i=1}^{n} X_i ,$$

és mivel $\sum_i B_i$ adott és ismert, csak $\sum_i X_i$ minimalizálandó.



4.5. ábra: A Gantt-diagram alapértelmezése a kétgépes gyártócella-modellhez

A probléma egzakt megoldását a következő könyvben találhatjuk meg:
Tóth Tibor: *Tervezési elvek, modellek és módszerek számítógéppel integrált gyártásban.* Miskolci Egyetemi Kiadó, 1998. (3. sz. Függelék, 235-238.old.).

A részletes levezetésből azonnal látható, hogy a sorrendet lépésről-lépésre meg lehet határozni a *Johnson* algoritmus segítségével. Az operációkutatásban azt a matematikai módszert, amely egymástól függő döntések sorozatát optimalizálja, *dinamikus programozásnak* nevezzük. *Johnson* algoritmusa tulajdonképpen dinamikus programozási problémát old meg.

A dinamikus programozás alaptétele a *Bellman*-féle optimalitási tétel, amelyet sokféleképpen meg lehet fogalmazni és be lehet mutatni.

A *Bellman*-féle optimalitási tétel (diszkrét és determinisztikus rendszerekre):

Legyen egy rendszerünk, amelynek állapota az egyes k fázisokban megváltozhat döntés következtében. Az egyes k fázisokban ($k=0,1,2,...,N$) a rendszer véges vagy végtelen, de megszámlálható állapotban lehet. "Politikának" nevezzük a döntések egy bizonyos sorozatát $k=0$-től $k=N$-ig. 'Alpolitikának" nevezzük a kapcsolódó döntések olyan sorozatát, amely valamely politikának része. Ha ebben az esetben az állapotváltozásoktól függő értékfüggvényt veszünk fel és ezt a függvényt optimálni kívánjuk, akkor igaz az alábbi tétel: ***Egy optimális politika csak optimális alpolitikákból állhat.***

Bellman megfogalmazásában: *"Valamely politika akkor optimális, ha egy adott periódusban - akármilyenek voltak az előző döntések - a hátralévő döntések optimális politikát alkotnak, figyelembe véve az előző döntések eredményét."*

(c) A Johnson-algoritmus kiterjesztése 3 gépre

A *Johnson*-algoritmust fel lehet használni az alábbi két különleges esetben:

$$min\ A_i \geq max\ B_i \quad \text{vagy} \quad min\ C_i \geq max\ B_i,$$

amikor az n elvégzendő műveletből álló munkáknál 3 gép A, B és C adott. Ilyenkor az idők vizsgálatát az $A_i + B_i$ és $B_i + C_i$ összegekkel végezzük.

Tekintsük a következő feladatot:

Definiálják az esztergálási, marási és köszörülési műveleteket az A_i, B_i és C_i időtartamok a $p_1,..., p_5$ alkatrészeken. Induljunk ki a következő táblázatból:

I. táblázat

	Esztergál (A_i)	Mar (B_i)	Köszörül (C_i)
p_1	8	6	7
p_2	12	3	10
p_3	9	5	4
p_4	15	4	18
p_5	11	5	10

Érvényes, hogy $min\ A_i \geq max\ B_i$, mert $8 > 6$. Összeállíthatjuk a második táblázatot is:

II. táblázat

	$A_i + B_i$	$B_i + C_i$
p_1	14	13
p_2	15	13
p_3	14	9
p_4	19	22
p_5	16	15

Alkalmazva a Johnson algoritmust, kapjuk:

1.

p_1	14	13
p_2	15	13
p_4	19	22
p_5	16	15
p_3	14	9

2.

p_1	14	13
p_4	19	22
p_5	16	15
p_2	15	13
p_3	14	9

3.

p_4	19	22
p_5	16	15
p_1	14	13
p_2	15	13
p_3	14	9

Az algoritmus jól követhető az 1)-2)-3) táblázatokon.

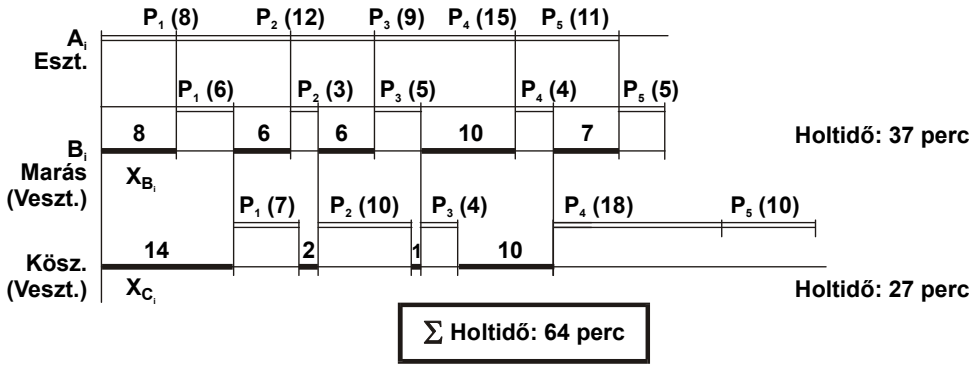
1. Megvizsgáljuk, hogy a II. táblázatban melyik a legkisebb idő; jelen esetben ez 9 perc.
2. Ha ez az érték a táblázat *első* oszlopába tartozó érték, akkor a neki megfelelő munkadarabbal kezdünk; ha második oszlopba tartozó, akkor a neki megfelelő munkadarabbal fejezzük be a megmunkálást.
3. Elkülönítjük a táblázat megfelelő sorát, majd a maradék táblázatrésszel folytatjuk az 1.-2.-3. lépést. Az optimális ütemezési sorrendre azt kapjuk, hogy

$$S = (p_4, p_5, p_1, p_2, p_3).$$

Vegyük az eredeti *ad hoc* sorrendet:

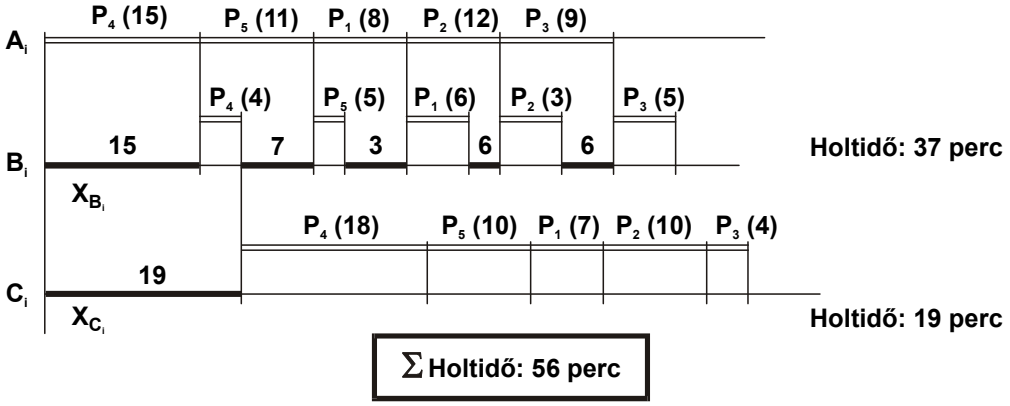
$$S_1 = (p_1, p_2, p_3, p_4, p_5).$$

Ábrázolva (4.6. ábra):



4.6. ábra: A Johnson-algoritmus 3 gépre való kiterjesztésének szemléltetése konkrét példán (Gantt-diagram, *ad hoc* sorrend)

A *Johnson*-algoritmus szerint (4.7. ábra):



4.7. ábra: A *Johnson*-algoritmus 3 gépre való kiterjesztésének szemléltetése konkrét példán (Gantt-diagram, *optimális* sorrend)

A két *Gantt*-diagramból látható, hogy a *Johnson* algoritmus 8 perccel csökkenti az állásidőket.

(d) Egyutas, előzés nélküli nagyfeladatok megoldása

Az egyutas, előzés nélküli feladat megoldására $m>2$ esetén számos heurisztikus módszert dolgoztak ki. Az alábbiakban néhány ilyen módszert mutatunk be:

(1) A leggyorsabb és legegyszerűbb a *Palmer* által kifejlesztett módszer, amelynek az a lényege, hogy minden egyes munkához prioritási indexet rendel és ezután a munkákat a prioritási indexek csökkenő sorrendjében ütemezi. A prioritási index kiszámítására szolgáló formula azt az elvet fejezi ki, hogy ez a módszer azokat a munkákat veszi előre, amelyeknek a megmunkálási idői az első gépeken rövidebbek a többihez képest. A prioritási index kiszámításának módja:

$$I_i = -\sum_{j=1}^{m} \{[m-(2j-1)]/2\} t_{ij},$$

ahol:

I_i - az i-edik munka prioritási indexe;

j - azon gép sorszáma, amelyen a j-edik műveletet kell végrehajtani;

m - az összes végrehajtandó műveletek száma;

t_{ij} - az i-edik munka j-edik műveletének ideje (az ugyancsak j-edik sorszámú gépen).

(2) A *Campbell-Dudek-Smith* algoritmus a feladatot k-számú kétgépes ütemezési problémára vezeti vissza, ahol $1 \le k \le m-1$. Ez az algoritmus m-1 mesterséges kétgépes problémát generál az eredeti m gépes feladatból, amelyek egyenként megoldhatók a *Johnson*-módszerrel. Kiválasztva közülük az átfutási idő szempontjából legjobbat, ez lesz az m-gépes probléma megoldása.

(3) *Page* gondolata a következő volt: mivel az ütemezésben a munkák sorbaállítása hasonló az adatfeldolgozás sorbarendezési problémájához, ezért az utóbbinál általánosan használt módszerek sikeresen alkalmazhatók a munkák sorbaállítási feladataira is.
Négy heurisztikus módszert fejlesztett ki, amelyek a következő rendezési eljárásokon alapulnak: *egyesítés* (M: merging), *párosítás* (P: pairing), *egyedi csere* (IE: individual exchange) és *csoportos csere* (GE: group exchange).

(4) Az ún. gyors hozzáférés eljárását *Dannenbring* dolgozta ki. Ez a heurisztikus módszer is egy súlyozási szkémát használ, mint a *Campbell-Dudek-Smith* algoritmus. Szintén a *Johnson* által megoldott kétgépes problémára vezeti vissza az m gépes feladat megoldását.
A végrehajtási időket az alábbi súlyozási séma határozza meg:

$$P_{i1} = \sum_{j=1}^{m} (m-j+1) t_{ij},$$

$$P_{i2} = \sum_{j=1}^{m} j t_{ij}.$$

Itt P_{i1} és P_{i2} az *i*-edik munka végrehajtási idői az *1*-es és *2*-es sorszámmal azonosított két fiktív gépen (idő dimenziójú mennyiségek);

m az *i*-edik munka összes műveleteinek száma és egyúttal a rendelkezésre álló gépek száma is (flowshop);

j az egy munkához szükséges műveletek és egyúttal a gépek futó indexe (sorszáma);

t_{ij} az *i*-edik munkavégrehajtási ideje a *j*-edik gépen (az eredeti *m* - gépes feladatból).

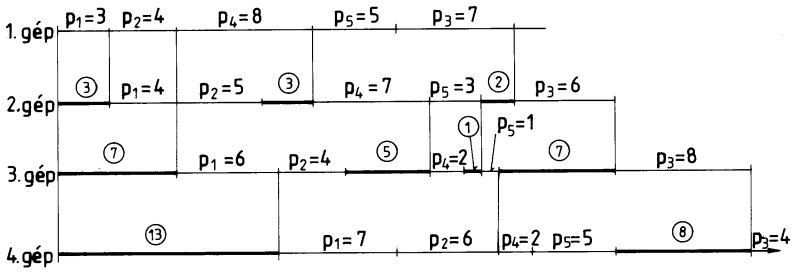
Az eredeti feladat megoldása ezután a *Johnson* eljárásnak a fent definiált két fiktív gépre való alkalmazásával áll elő.

Példa:

Induljunk ki a következő feladatból. Legyen *m=4* és *n=5*, az ütemezésnél a kiinduló táblázatot tekintsük egyben egy *ad hoc* megmunkálási sorrendnek. Határozzuk meg a veszteségidők összegét az *ad hoc ütemezés*, a *Palmer-féle prioritási indexek* szerinti sorrend és a *Dannenbring-féle sorrend* szerint.

$p_i \setminus j$	1	2	3	4
p_1	3	4	6	7
p_2	4	5	4	6
p_4	8	7	2	2
p_5	5	3	1	5
p_3	7	6	8	4

ad(1) A fenti táblázat *Gantt*-diagramja (4.8. ábra):



4.8. ábra: ad hoc megmunkálási sorrend Gantt-diagramja 4 gép esetén

A veszteségidők összege:

$$\sum_i t_{v,i} = 0 + 8 + 20 + 21 = 49 \text{ perc.}$$

ad(2) A *Palmer*-féle prioritási indexek alapján:

$$I_i = -\sum_{j=1}^{m} \left[(m-(2j-1))/2 \right] t_{ij}$$

$$I_1 = -\Big\{ \underbrace{\left[(4-(2-1))/2\right]}_{1.5} 3 + \underbrace{\left[(4-(4-1))/2\right]}_{0.5} 4 +$$
$$\underbrace{\left[(4-(6-1))/2\right]}_{-0.5} 6 + \underbrace{\left[(4-(8-1))/2\right]}_{-1.5} 7 \Big\} = 7.$$

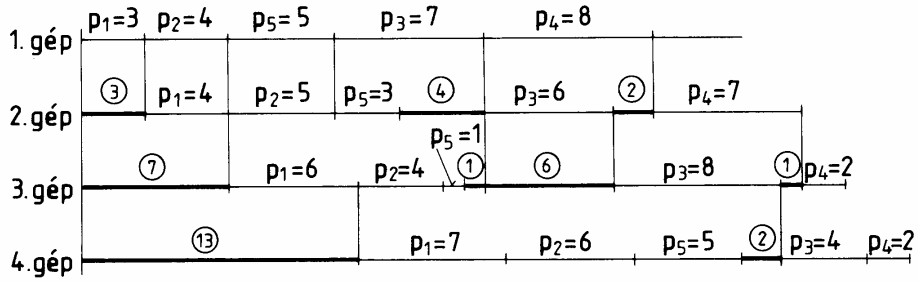
$$I_2 = -\{1.5.4 + 0.5.5 - 0.5.4 - 1.5.6\} = 2.5.$$
$$I_4 = -\{1.5.8 + 0.5.7 - 0.5.2 - 0.5.2\} = -11.5.$$
$$I_5 = -\{1.5.5 + 0.5.3 - 0.5.1 - 1.5.5\} = -1.$$
$$I_3 = -\{1.5.7 + 0.5.6 - 0.5.8 - 1.5.4\} = -3.5.$$

A *Palmer*-indexek szerint a sorrend:

$$(S_P) = (p_1, p_2, p_5, p_3, p_4),$$

mert $7>2.5>-1>-3.5>-11.5$.

A megfelelő *Gantt*-diagram (4.9. ábra):



4.9. ábra: Optimális megmunkálási sorrend Gantt-diagramja 4 gép esetén (*Palmer*-módszer)

A veszteségidők összege:

$$\sum_i t_{v,i} = 0 + 9 + 15 + 15 = 39 \text{ perc.}$$

ad(3) *Dannenbring* szerint:

A két fiktív gépre generált P_{i1} és P_{i2} idők:

$$P_{i1} = \sum_{j=1}^{m} (m-j+1) t_{ij} \text{ és } P_{i2} = \sum_{j=1}^{m} j t_{ij}.$$

Behelyettesítve:

$$\begin{cases} P_{11} = 4.t_{11} + 3.t_{12} + 2.t_{13} + 1.t_{14} = 4.3 + 3.4 + 2.6 + 1.7 = 43 \\ P_{12} = 1.t_{11} + 2.t_{12} + 3.t_{13} + 4.t_{14} = 1.3 + 2.4 + 3.6 + 4.7 = 57 \end{cases}$$

$$\begin{cases} P_{21} = 4.4 + 3.5 + 2.4 + 1.6 = 45, \\ P_{22} = 1.4 + 2.5 + 3.4 + 4.6 = 50. \end{cases}$$

$$\begin{cases} P_{31} = 4.7 + 3.6 + 2.8 + 1.4 = 66, \\ P_{32} = 1.7 + 2.6 + 3.8 + 4.4 = 59 \end{cases}$$

$$\begin{cases} P_{41} = 4.8 + 3.7 + 2.2 + 1.2 = 59, \\ P_{42} = 1.8 + 2.7 + 3.2 + 4.2 = 36. \end{cases}$$

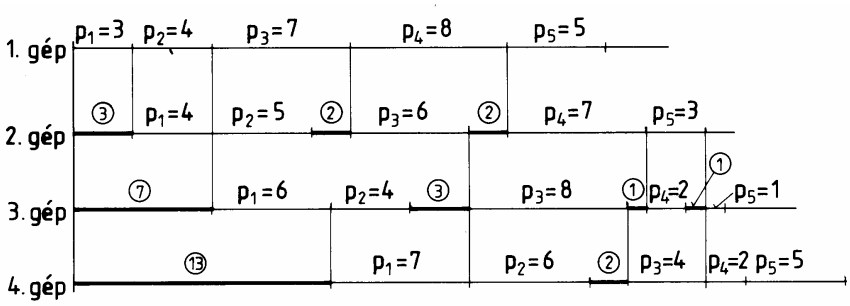
$$\begin{cases} P_{51} = 4.5 + 3.3 + 2.1 + 1.5 = 36, \\ P_{52} = 1.5 + 2.3 + 3.1 + 4.5 = 34. \end{cases}$$

A két fiktív gépnek megfelelő *Johnson*-tábla:

$p_i \setminus j$	1.fiktív gép	2.fiktív gép
p_1	43	57
p_2	45	50
p_3	66	59
p_4	59	36
p_5	36	34

Az optimális sorrend: $(S_D) = (p_1, p_2, p_3, p_4, p_5)$.

A *Gantt*-diagram (4.10. ábra):



4.10. ábra: Optimális megmunkálási sorrend Gantt-diagramja 4 gép esetén (*Dannenbring*-módszer)

A veszteségidők összege:

$$\sum_i t_{v,i} = 0 + 7 + 12 + 15 = 34 \text{ perc.}$$

Az összehasonlításból jól látszik, hogy az *ad hoc* sorrendhez képest a *Palmer*-módszer is hoz lényeges javulást, de a legjobb eredményt a *Dannenbring* módszer adja.