

Budapesti Műszaki és Gazdaságtudományi Egyetem
Gépészmérnöki Kar
Gyártástudomány és -technológia Tanszék

Gyulai Dávid

ÚJRAKONFIGURÁLHATÓ GYÁRTÓRENDSZER SZIMULÁCIÓJA

TDK dolgozat

KONZULENSEK:

Dr. Váncza József

Vén Zoltán

Budapest, 2011

Tartalomjegyzék

Absztrakt	4
Rövidítések	5
1. Irodalmi áttekintés	6
1.1. A gyártórendszerek fejlődése	7
1.1.1. Rugalmas gyártórendszerek	7
1.1.2. Újrakonfigurálható gyártórendszerek	9
1.2. Gyártórendszerek szimulációja	12
1.2.1. Modellezés és szimuláció	12
1.2.2. Siemens Plant Simulation	16
2. A feladat ismertetése	18
2.1. A gyártórendszer bemutatása	18
2.2. Modellépítés	19
2.3. Szimuláció	20
2.4. Az eredmények kiértékelése és javaslattétel	20
3. A rendszermodell felépítése	21
3.1. A szimuláció bemeneti értékei	21
3.2. A rendszer DES modellje	23
4. A rendszer szimulációs vizsgálata	30
4.1. A legfontosabb szimulációs paraméterek	30
4.2. A kiértékeléshez szükséges vizsgálatok	32
4.2.1. Előzetes teszt	32

4.2.2.	Érzékenység vizsgálat 1.	33
4.2.3.	Érzékenység vizsgálat 2.	35
4.2.4.	Az ütemezés hatása a teljes rendszerre	38
4.2.5.	Az ütemezés hatása a rendszerre az L5 sor nélkül	39
4.2.6.	Az ütemezés hatása a rendszerre: sorrendiség	40
5.	Az eredmények értékelése és a dolgozat összefoglalása	42
5.1.	Javaslat a rendszer magvalósításához	42
5.2.	Az elért eredmények értékelése	43
5.3.	Nyitott kérdések a témával kapcsolatban	44
5.4.	A feladat folytatása	44

Absztrakt

A digitális vállalati technológiák elterjedésével egyre több lehetősége nyílik a tervező és fejlesztőmérnököknek arra, hogy az új elvárásoknak (rövidebb átfutási idő, alacsonyabb tétel-nagyság, növekvő termékféleség stb.) megfelelően működő gyártórendszereket tervezzenek, illetve tökéletesítsék, optimalizálják a már meglévő rendszereket. A fejlődés a gyártórendszerek esetén is töretlen, a korábban újdonságnak számító rugalmas gyártórendszerek (FMS) mellett egyre gyakrabban alkalmaznak újrakonfigurálható rendszereket (RMS), amelyek még hatékonyabban tudnak alkalmazkodni a változó megrendelői igényekhez, illetve a termelési láncban bekövetkező változásokhoz, zavarokhoz.

TDK dolgozatomban az előbbi két tématerület együttes alkalmazását szeretném bemutatni egy konkrét, az iparban is alkalmazott megoldás alapján. A fő cél egy olyan diszkrétidejű, eseményorientált szimulációs modell elkészítése volt, amely egy újrakonfigurálható gyártórendszer működését a legfontosabb szempontok alapján reprezentálja. A bemeneti paraméterek változtatása a modell mélyrehatóbb ismerete nélkül is lehetséges, valamint nem igényel semmilyen előzetes programozási, illetve modellezési ismeretet. A kapacitás-szimuláció elvégzése alapján meghatározható a munkaállomások, gépek, illetve az operátorok optimális száma, valamint becsülhető az átállások és a sorösszeépítések ideje.

A szimuláció egyik bemeneti értéke a termelés ütemezése, így megvizsgálható az is, hogy az egyes átütemezések milyen hatással vannak az erőforrások kihasználtságára, illetve hogyan változik a gyártási időtartam adott termékmennyiségek esetén, megváltozott ütemezéssel.

A szimulációs modell segítségével megtervezhető egy az autóiparban is jól alkalmazható újrakonfigurálható szerelőrendszer, amely a jelentős hely- és költségmegtakarítás miatt sokkal eredményesebben működtethető, mint a korábbi dedikált szerelősorok.

Rövidítések és jelölések

Rövidítés	Jelentés
-----------	----------

CNC	computer numerical control
DES	discrete event simulation
DML	dedicated manufacturing line
FMS	flexible manufacturing system
JIT	just-in-time
MF	machining flexibility
PLC	programmable logic controller
RAS	reconfigurable assembly system
RF	routing flexibility
RMS	reconfigurable manufacturing system
RMT	reconfigurable machine tool
TPS	Toyota production system

Pszeudokódok jelölései

A dolgozatban található fontosabb „*methodok*” pszeudokódjaiban a „.” karakter az objektumok hierarchiáját jelzi, a pont előtt álló objektum mindig egy magasabb hierarchiai szintet jelent. A táblázatok esetén a szögletes zárójelben lévő értékek közül első szám az oszlop-, a második pedig a sorindex.

1. fejezet

Irodalmi áttekintés

A gyártórendszerek felépítése az elmúlt évtizedekben folyamatosan változott, egyre fejlettebb, az ipar igényeit jobban kielégítő struktúrák alakultak ki. A rendszerek fejlődésének két fő oka a következő:

- A technikai és a technológiai újítások lehetővé teszik a legújabb gépek és berendezések alkalmazását.
- Gyakran a változások nem a technikai újítások alkalmazásában rejlenek, hanem a gyártórendszerek struktúrájának megváltoztatásában. Ennek az egyik oka az lehet, hogy az elmúlt években nem voltak olyan korszakos újítások, mint korábban az NC/CNC technológia, vagy a PLC-k gyártórendszerbe integrálása.

A rendszerek struktúrájának megváltoztatása a következő evolúciós folyamatot eredményezte a gyártórendszerek esetén:

- tömeggyártás
- *just-in-time* gyártás
- *lean*¹
- rugalmas gyártórendszerek
- újrakonfigurálható gyártórendszerek

¹A magyar szakirodalomban a „*lean*” elnevezés helyett gyakran a „karcsúsított gyártás” kifejezést használják.

Mivel az egyes struktúrák a különböző iparágak esetén eltérő mértékben terjedtek el, ezért nehéz lenne időrendi sorrendet felállítani az előzőek esetén. A *lean* koncepció azonban mér-földkőnek tekinthető a fenti evolúciós folyamatban mivel több később kialakult módszer –mint például a rugalmas gyártórendszer– alapjául szolgált.

1.1. A gyártórendszerek fejlődése

A XX. század közepén még az úgynevezett „dedikált” gyártósorok (DML) jelentették a technika csúcsát, amelyek egy adott termék gyors, automatizált gyártását tették lehetővé alacsony költségek mellett. Azonban a gyártott termék fejlesztése, megváltoztatása esetén az egész gyártósort át kellett alakítani, ami egyrészt nagy kiesést jelentett a termelésben, másrészt az átalakítás is jelentős költségvonzattal járt. A termékek fejlesztése az életciklus le-rövidülésével egyre gyakoribb lett, így szükség volt arra, hogy a gyártórendszerek megfelelően tudjanak reagálni a változásokra a rugalmasságuk révén. A nem egy konkrét gyártástechnológiai eljárás vagy gyártórendszer fajta, hanem egy menedzsment „filozófia”. A *lean* a Toyota Production System-ből alakult ki, és terjedt el az egész világon, mivel a Toyota menedzsmentje sikeresen bebizonyította, hogy az új koncepció bevezetésével jelentős piaci előnyre tudnak szert tenni a versenytársaikkal szemben.

A *lean* célja nem más, mint kielégíteni az összes vevői igényt mégpedig úgy, hogy a folyamatokat igyekeznek tökéletesíteni azáltal, hogy kiküszöbölik az összes lehetséges hibát az ellátási láncban, és biztosítják a folyamatos anyagáramlást. Ez röviden úgy foglалható össze, hogy igyekeznek a legjobb terméket gyártani a lehető legrövidebb idő alatt. Egy jól működő, *lean* alapján kialakított gyártórendszer üzemeltetése kevesebb helyet, munkaerőt, nyersanyagot és időt igényel mint a korábbi gyártórendszerek. James P. Womack és Daniel T. Jones rámutattak arra, hogy a Toyota *lean* alapú gyártórendszerének termelékenysége kétszer, minősége pedig százszor jobb a versenytársak rendszereinél [11]. A *lean* tehát jó kiindulási alapként szolgál az előző fejezetben már említett rugalmas gyártórendszerhez.

1.1.1. Rugalmas gyártórendszerek

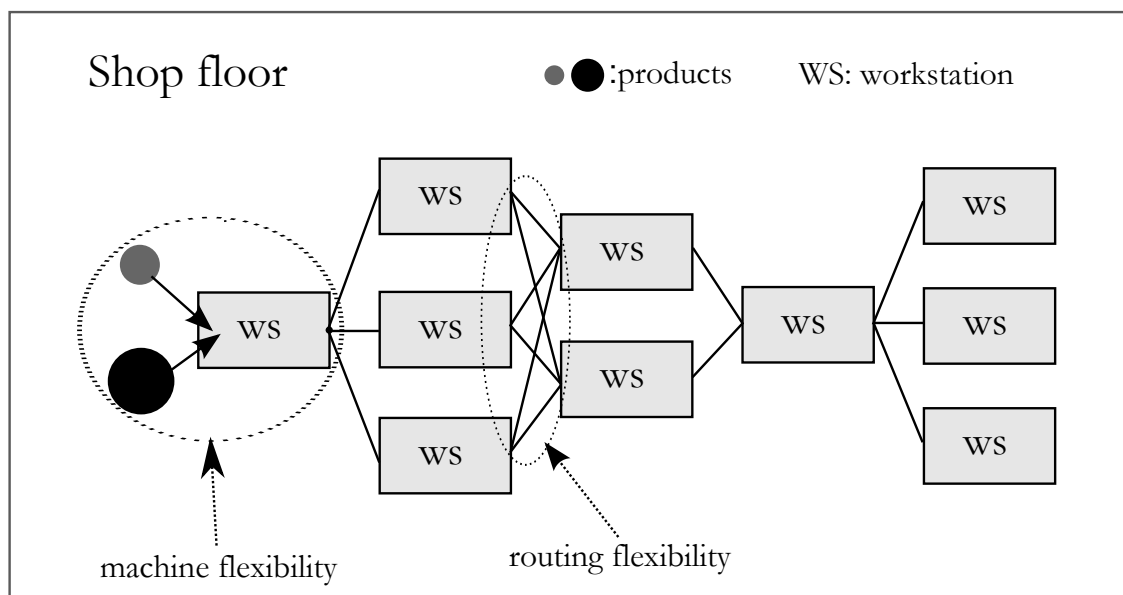
A rugalmasság a rendszer azon tulajdonsága, hogy képes gyorsan alkalmazkodni a változásokhoz, mint például a termékváltás, terheltség (kihasználtság) illetve a meghibásodások. Alapvetően két fajtája létezik a rugalmasságnak, amelyek alapján további típusok definiálhatók. A két fő típus az úgynevezett *machine flexibility* vagyis géprugalmasság valamint a *routing flexibility*, vagyis az anyagáramlás rugalmassága.

A géprugalmasság a gyártórendszer elemeinek azon tulajdonsága, hogy szerszám-, illetve palettacserékkel képes többféle termék előállítására is. A nagyobb géprugalmasság kisebb sorozatnagyságokat, nagyobb gépkihasználtságot illetve rövidebb átfutási időket eredményez. Az anyagáramlás rugalmassága révén a rendszer képes ugyanazt a terméket eltérő gépsorrenddel is legyártani. Kihasználva a rendszernek ezt a képességét, a meghibásodások illetve karbantartások esetén is lehetséges az előírt kihozatal elérése [9]. A rugalmasság két fő fajtáját az 1.1 ábra szemlélteti.

A rugalmas gyártórendszereket (FMS) általánosságban a következő három egység alkotja:

- *gépek*, amelyek a kívánt rugalmasság elérése érdekében CNC vezérléssel rendelkeznek, leggyakrabban esztergák, marógépek illetve megmunkálóközpontok;
- *anyagmozgató rendszer* a gépek között, ez jellemzően konvektor pályákból, manipulátorokból, illetve anyagmozgató-robotokból áll;
- *irányítórendszer*, amely jellemzően számítógép illetve PLC hálózathoz épül fel és biztosítja a gépek közti anyagáramlást, illetve a vezérli a gépeken végbemenő folyamatokat.

A rugalmas gyártórendszerek legnagyobb előnyei a következők: nagyobb termelékenység, alacsonyabb egységár és nagyobb hatékonyság változó termelési körülmények és vevői igények esetén. Nagyon fontos megemlíteni, hogy megfelelő tervezés és telepítés esetén a gépek kihasználtsága, valamint az ütemezés optimalizálható.



1.1. ábra. A géprugalmasság, illetve az anyagáramlás rugalmasságának szemléltetése

A számos előny mellett persze több hátránya is van a flexibilis gyártórendszereknek, ezek közül is talán a telepítés költsége emelhető ki, ez ugyanis ideális esetben is többszöröse lehet egy dedikált gyártórendszernek. A másik jelentős hátrány, hogy a rugalmas gyártórendszerek esetén alkalmazott CNC technológia (jellegéből adódóan) nem termékspecifikus, így a munkadarabok megmunkálása időigényesebb és körülményesebb mint a célgépek, illetve a dedikált rendszerek esetén [2].

1.1.2. Újrakonfigurálható gyártórendszerek

A rugalmas gyártórendszerek jellemzője, hogy képesek alkalmazkodni a változó termelési körülményekhez úgy, hogy csak a rendszer működése változik meg, a struktúrája pedig változatlan marad. Ezzel szemben az újrakonfigurálható rendszerek rendelkeznek azzal a képességgel, hogy a szerkezetük megváltoztatásával alkalmazkodnak a termeléshez. Ez a két típus alkotja a változóképes termelési rendszerek („*changeable systems*”) csoportját. Általánosságban elmondható, hogy a változóképes rendszerek célja a termelésben bekövetkező előre tervezett és a váratlan változásokhoz való alkalmazkodás a gyártási hierarchia minden szintjén a gazdaságos működés érdekében.

Az újrakonfigurálható gyártórendszerek (RMS) költséghatékony megoldást jelentenek a piaci változásokra, egyesítik a dedikált és a rugalmas rendszerek két előnyös tulajdonságát, a termelékenységet illetve a rugalmasságot. A korábban már bemutatott evolúciós folyamat esetén látható, az újrakonfigurálható rendszerek kialakulását megelőző legfontosabb lépés a rugalmas gyártórendszerek kifejlesztése volt a '80-as években. A '90-es években egyre fontosabbá vált, hogy megtalálják a rugalmas rendszerek hátrányait, csökkentsék a költségeket, illetve optimalizálják őket. H. ElMaraghy a következőképpen foglalta össze a az RMS koncepcióját [1]:

„Az újrakonfigurálható gyártórendszer koncepciója az elmúlt néhány évben alakult ki azért, hogy megpróbálja elérni a változóképes rendszerek funkcionalitását, illetve a nagy gyártási kapacitást. Ez úgy érhető el, hogy a rendszerben levő gépek, gépelemek, cellák illetve anyagmozgató rendszerek igény szerint hozzáadhatók, elvehetők illetve átcsoportosíthatók. Teljes mértékben újrakonfigurálható rendszer még nem létezik, de világszerte a kutatások középpontjában áll, főleg a gépesítés, illetve az irányítási rendszer megvalósítása szempontjából. Az elképzelés támogatói szerint az új rendszer hosszú távon képes lesz a rugalmas rendszereknél olcsóbb megoldást kínálni, hosszabb élettartam, illetve a jobb kihasználtság által. Összefoglalva, az újrakonfigurálható gyártórendszer egy olyan rendszer, amelynek változtatható a

rugalmassága, míg a rugalmas rendszerek esetén ez az érték állandó.”

Az újrakonfigurálható gyártórendszer a következőképpen definiálható: olyan termelési rendszer, amely képes gyorsan változtatni a struktúráját (a hardveres és szoftveres összetevőket is ideértve), ezáltal alkalmazkodva a termékcsaládon belül bekövetkező kapacitás-, illetve funkcionalitásbeli változásokhoz [3].

Egy tipikus újrakonfigurálható rendszert az alábbi hat tulajdonság jellemez (nem feltétlenül rendelkezik mindegyikkel):

- modularitás,
- integrálhatóság,
- változtatható rugalmasság,
- skálázhatóság,
- konvertálhatóság,
- diagnosztizálhatóság.

Az alábbi táblázat a három korábban bemutatott gyártórendszer típust hasonlítja össze a legfőbb tulajdonságaik alapján:

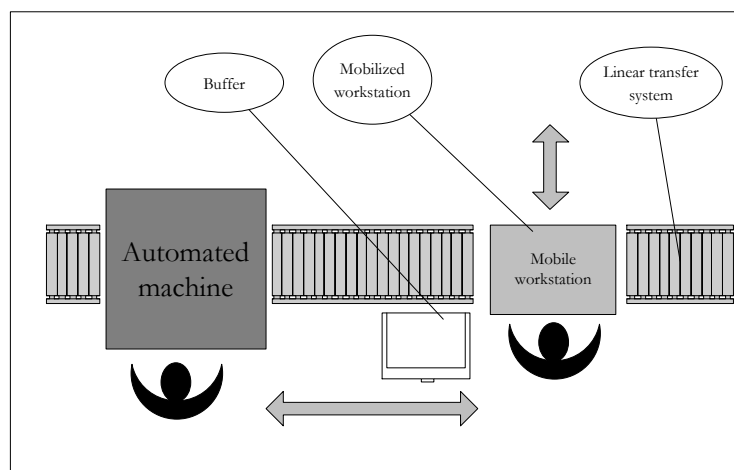
	DML	RMS	FMS
Rendszer struktúra	Fix	Változtatható	Változtatható
Gép-felépítés	Fix	Változtatható	Fix
Rendszer-fókusz	Termék	Termékcsalád	Gép
Skálázhatóság	Nem	Igen	Igen
Rugalmasság	Nem	Változtatható	Általános
Egyidőben üzemelő gépek	Igen	Lehetséges	Nem
Termelékenység	Nagyon magas	Magas	Alacsony
Ár/Darab	Alacsony	Közepes	Eltérő

1.2. ábra. A három fő gyártórendszer típus összehasonlítása

A rugalmas gyártórendszerekhez hasonlóan az újrakonfigurálható rendszerek fő összetevői is CNC gépek illetve rugalmas anyagmozgató-rendszer (pl. konvektor), azzal a különbséggel hogy rendelkezik néhány speciális rendszer-összetevővel is. Ilyenek például az újrakonfigurálható szerszámgépek (*reconfigurable machine tool, RMT*), illetve az újrakonfigurálható ellenőrző-berendezések (*reconfigurable inspection machine*). Az általános célú CNC gépekkel szemben az RMT-k speciálisan az újrakonfigurálható rendszerekhez készülnek, azzal a céllal, hogy a követelmények, illetve paraméterek megváltozásához költséghatékonyan tudjon alkalmazkodni [4].

Újrakonfigurálható szerelősorok

A tipikus újrakonfigurálható gyártórendszert magas fokú automatizáltság és számítógépes irányítás jellemzi, amint azt az előző fejezetben már ismertettem, azonban létezik egy speciális fajtája is az újrakonfigurálható rendszereknek, amelynek ismerete nélkülözhetetlen a dolgozat témájának szempontjából.



1.3. ábra. Hibrid szerelőrendszer sémája

A gépipari termékek, például az autóalkatrészek esetén viszonylag könnyen megoldható az, hogy az anyagmegmunkálás újrakonfigurálható rendszer segítségével történjen. A tipikus RMS-t ugyanis megmunkálógépek, például esztergák illetve megmunkálóközpontok alkotják, léteznek azonban olyan újrakonfigurálható rendszerek, amelyek elsődleges célja az, hogy a termékek összeszerelését lehessen rajtuk elvégezni. Ezeket újrakonfigurálható szerelősoroknak/-rendszereknek nevezzük. Rugalmas szerelősorok már évek óta léteznek, több gyártó is foglalkozik ilyen rendszerek összetevőinek a gyártásával. Megállapítható, hogy napjainkban egyre nagyobb szerepet kap a gyártási követelmények közül a nagyobb gyártási kihozatal, illetve az alacsony ár. Érdekes megoldást kínálnak erre a problémára az úgynevezett hibrid-rendszerek, amelyekben az automatizált eszközöket a kézi szereléssel kombinálják [6]. Az ilyen hibrid rendszerek gazdaságosság szempontjából ideálisak a közepes sorozatnagyságú termékekhez, alkalmazásukat azonban nem csak a gyártási mennyiség indokolja, hanem olyan fontos paraméterek is mint például a műveleti idő, valamint a szerelési művelet bonyolultsága. Az 1.3 ábra bemutatja egy ilyen hibrid rendszer általános felépítését. Ebben az esetben a munkafolyamlatok lineáris anyagmozgató rendszerrel vannak összekötve, az állomások percek alatt kicserélhetők, ezáltal megoldható a termékváltás, illetve a manuális műveletek automatizáltra váltása. Természetesen rengeteg elrendezési változat megvalósítható, például osztóasztalok,

illetve ívelt vonalú konvektorok alkalmazásával, az alapkoncepció azonban minden esetben változatlan. Megfelelő elrendezés kiválasztását költségelemzés segíti, az egyes változatok alkalmazásának költségvonzata között nagyságrendbeli különbség is lehetséges [10]. Az előbb ismertetett koncepció alkalmazása főként gyorsan változó termék, illetve mennyiségek esetén ideális, a következő okok miatt:

- A beruházási költsége egy hibrid rendszer kialakításának jelentősen alacsonyabb, mint egy moduláris automatizált rendszernek.
- Alacsonyabb anyagi kockázattal jár a telepítése.
- Egyszerűbb bővíthetőség.
- Ha a körülmények megkövetelik, akkor párhuzamosítani lehet a folyamatokat újabb cellák beüzemelésével.

1.2. Gyártórendszerek szimulációja

A dolgozat témájából adódóan elengedhetetlen a szimuláció, és ehhez kapcsolódóan a modellezés alapvető fogalmainak ismerete. A szimulációs vizsgálatok tárgya nem más mint a *rendszer*, amely a fizikai valóság meghatározott szempontok alapján elhatárolt része. A rendszert a *modell* segítségével lehet különböző formális eszközökkel leképezni. Ilyen formális eszköz lehet a rendszer működését leíró differenciálegyenlet-rendszer, amely segítségével meghatározhatjuk például a rendszer állapotter-modeljét. A modellben a rendszernek a vizsgálat szempontjából fontos paramétereit reprezentáljuk, így módon könnyebb megérteni a működését is. A modellek többsége azonban túl összetett ahhoz, hogy csupán analitikus módszerek segítségével elvégezhessünk rajtuk minden szükséges vizsgálatot. A *szimuláció* során számítógép segítségével végezzük el a modellen a numerikus vizsgálatokat, így a nagyobb számítási kapacitás révén komplex rendszermodelleket is viszonylag gyorsan lehet vizsgálni [7]. Fontos megemlíteni az állapotváltozók fogalmát, amely nem más mint azon paraméterek halmaza, amelyek egy adott pillanatban egyértelműen leírják a rendszer állapotát. Az állapotváltozók megfelelő megválasztása gyakran a modellezés kulcsa, a helyes paraméterhalmaz megválasztásával ugyanis rengeteg idő és energia takarítható meg.

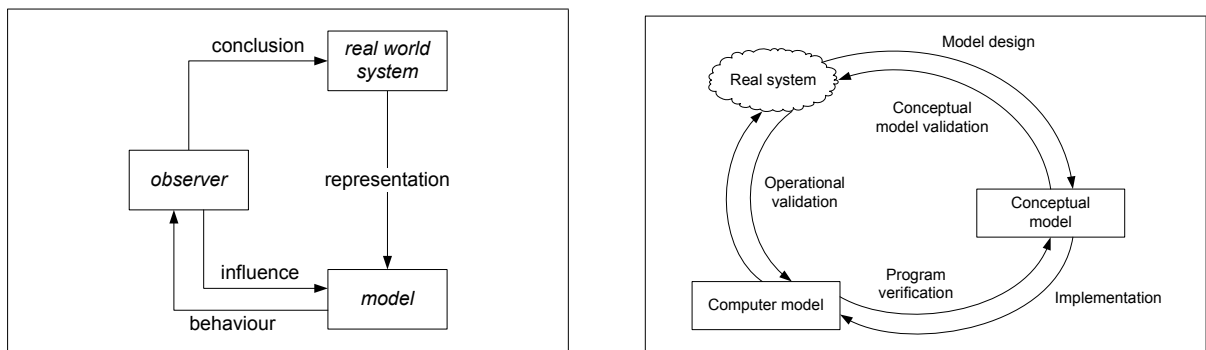
1.2.1. Modellezés és szimuláció

A dolgozatomban a gyártórendszer vizsgálatát diszkrétidejű esemény-orientált szimuláció (DES) segítségével végzem el, így elengedhetetlen ennek a módszernek a rövid bemutatása.

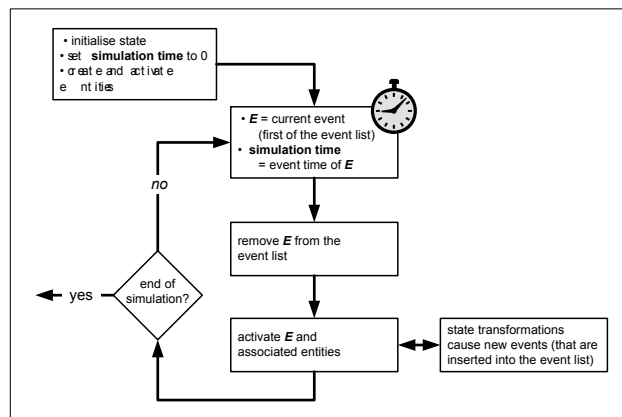
A DES során az állapotváltozók értékeit számítjuk ki meghatározott időpontokban. Matematikailag ez úgy fogalmazható meg, hogy a rendszer állapota csak adott időpontokban változhat meg, ezen időpontok halmaza pedig véges. Ezekben az időpontokban történnek az *események*, amelyek megváltoztatják a rendszer aktuális állapotát. A rendszer állapotváltozóinak halmaza minden egyes esemény során megváltozik, ami azt jelenti hogy egyes erőforrások felszabadulhatnak mások állapota pedig foglalttá válhat [5].

A DES modellek objektumokból épülnek fel, melyeket *entitásoknak* nevezünk. Az entitások valós rendszerkomponenseket modelleznek, ezáltal egymással kapcsolatban állhatnak. Az entitás fogalma kapcsolatban áll az *objektummal*, amit az objektum-orientált programozás során használunk. Az entitások olyan objektumoknak tekinthetők, amelyek viselkedése a szimuláció közben van definiálva. Az entitásokat az állapotuk, valamint az időbeli viselkedésüket leíró szabályok (*methods of rules*) alapján lehet definiálni. A előbbi tulajdonságok alapján a legtöbb DES modell az alábbi részekből épül fel [5]:

- *Állapottér*: Az állapotváltozók halmaza ahhoz szükséges, hogy a rendszert egy adott időpontban jellemezhessük.
- *Szimulációs óra*: Egy olyan változó, amelynek mindenkor értéke a szimulációs idő.
- *Eseménylista*: Egy olyan lista, amely tartalmazza az eseményeket, és a hozzájuk tartozó időpontokat
- *Statisztikai változók*: A rendszer statisztikai jellemzőinek értékeit tárolják.
- *Inicializáló eljárás*: Alprogram, amely a szimulációs modellt a „0” időpontbeli állapotba állítja.



1.4. ábra. A modellezés és szimuláció fontosabb foglmainak és folyamatainak szemléltetése [7]

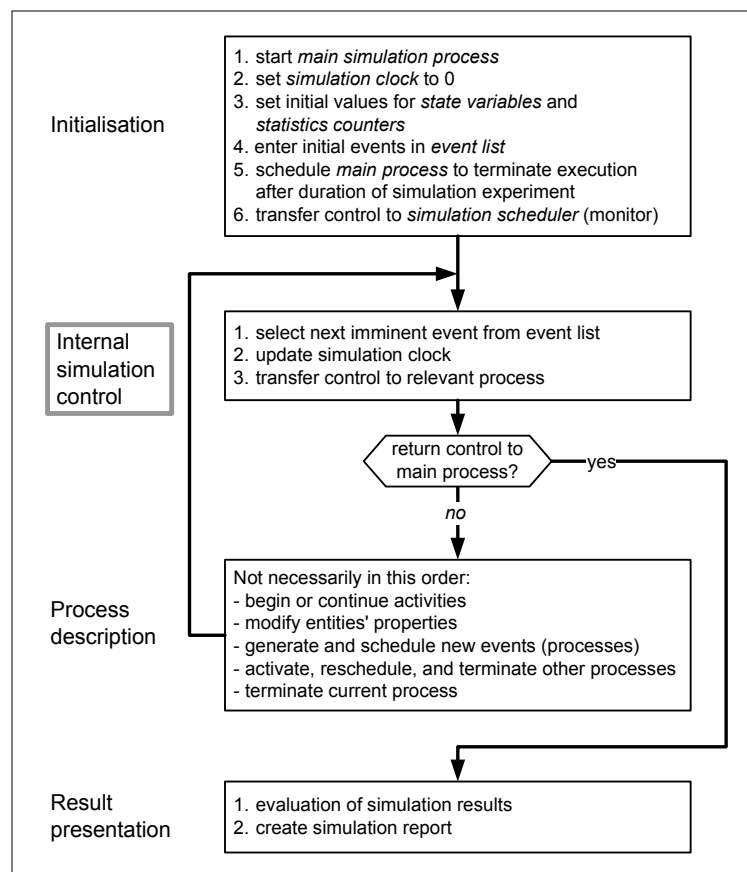


1.5. ábra. A DES egyszerűsített folyamatábrája [7]

- *Időzítő eljárás*: Alprogram, amely az eseménylistából kiválasztja a következő eseményt, és a szimulációs órát az esemény időpontjára állítja át.
- *Eseménykezelő eljárás*: Alprogram, amely átállítja a rendszer állapotát az adott esemény hatására (minden eseményhez tartozik egy eseménykezelő eljárás).
- *Könyvtár eljárás*: Alprogramok összessége, amelyek véletlenszerű megfigyeléseket végeznek el a valószínűségi eloszlások alapján.
- *Jelentéskészítő eljárás*: Alprogram, amely becsléseket készít a szimuláció várható értékeiről és a futás idejéről.
- *Főprogram*: Eljárás, amely az időzítő eljárástól megkapja a következő eseményt, majd az eseménykezelő eljárás segítségével átállítja a rendszer állapotát. A főprogram kezeli a különböző megszakítási eljárásokat, illetve a szimuláció lefutásakor meghívja a jelentéskészítő eljárást.

A szimuláció egy interdiszciplináris módszer, amely segítségével a tudomány különféle területeinek problémái oldhatóak meg, például természettudományos, társadalomtudományi, gazdasági problémák. A dolgozat esetében a szimuláció segítségével egy gyártórendszer működését lehet vizsgálni, és a kapacitását megbecsülni.

A gyártórendszerek szimulációs vizsgálatának legnagyobb előnye, hogy a mérnök illetve menedzser a lokális változások globális hatását tudja vizsgálni, így átfogó képet kaphat a rendszer működéséről. Ez legtöbb esetben azt jelenti, hogy a különböző munkaállomásokon végzett változtatások teljes rendszerre gyakorolt hatását lehet becsülni, mérni. A szimulációs vizsgálatok legnagyobb előnye gyártási, illetve ütemezési problémák esetén az alábbiak[8]:



1.6. ábra. Folyamatábra DES indításához és futtatásához [7]

- A szimuláció segítségével feltárhatók a gyártásközi készletek és csökkenthetők az átfutási idők, ezáltal növelhető az erőforrások kihasználtsága továbbá csökken az állótőke és a működtetés költsége.
- A folyamatok jobban átláthatóak és könnyebben megérthetőek a modellépítési fázisban.
- Virtuális statisztikai adatokhoz lehet jutni a szimulációs eredmények elemzése alapján.

Modellezési szempontból a gyártórendszerek tipikusan diszkrétidejű esemény-orientált rendszerek. Ahhoz, hogy validálható modellt tudjunk építeni, figyelembe kell venni a gyártórendszer diszkrétidejű eseményeinek egymásutánosságát, valamint a folytonos idejű folyamatok működését is. A szimuláció alkalmazása gyártási folyamatok esetén több célt szolgálhat, a vizsgálatokat alapvetően három fő kategóriába lehet besorolni:

- erőforrásigények számítása
 - a gépek száma, típusa és elrendezése valamilyen adott esetben,

- az anyagmozgató rendszer, valamint az ellátási lánc elemeinek kiválasztása,
- a pufferek helye valamint mérete,
- stb.
- teljesítmény kiértékelés
 - kapacitás szimuláció,
 - átfutási idők elemzése,
 - szűk keresztmetszetek elemzése,
- műveleti sorrendek elemzése
 - ütemezések elemzése,
 - irányítási stratégiák kidolgozása,
 - megbízhatósági számítások (pl. gépleállások hatása),
 - stb.

1.2.2. Siemens Plant Simulation

A gyártórendszerek esetén használt szoftverekkel szemben támasztott követelmények nagyon hasonlóak más szimulációs alkalmazásokéhoz, egy különbséggel: a legtöbb modern termelési rendszer rendelkezik valamilyen anyagmozgató rendszerrel, amelyeket gyakran nagyon nehéz megfelelően modellezni, működésük ugyanis folytonos időben írható le megfelelően. A alkalmazás-orientált szimulációs szoftverek felhasználóbarát megoldásokat kínálnak erre a problémára, így a modellépítés jóval egyszerűbb mint az általános célú programok esetén. Napjainkban a kereskedelmi forgalomban elérhető alkalmazás-orientált programok kínálata meglehetősen nagy, az iparban leginkább elterjedt programok a következők: *Siemens Plant Simulation*, *Arena*, *TaylorED*, és a *Simul8*. Az újrakonfigurálható rendszer szimulációjához és modellezéséhez a Siemens Plant Simulation programját használtam, világszerte ez a leginkább ismert és elterjedt program a gyártórendszerek modellezéséhez. A szoftver sok speciális eszközt kínál fel, amelyek megkönnyítik a mérnökök munkáját, illetve sokoldalúvá teszik a használhatóságot.

A Plant Simulation rendelkezik grafikus felhasználói felülettel, amely segíti a felhasználót a modellépítés, a szimuláció és az adatelemzés során is. A program rendelkezik egy könyvtárral, amely előre definiált entitásokat tartalmaz, ezeket „drag and drop” módszerrel lehet mozgatni illetve összekötni, így gyorsan elő lehet állítani egyszerűbb modelleket. A

könyvtárak felépítése hierarchikus szerkezetű, így a modellépítés is történhet hierarchikusan (pl. cella- illetve gépszint). A Plant Simulation egyesíti a komponens-alapú megközelítést az objektum-orientált programozással, ezáltal a létrehozott modell-blokkok örökölhetnek tulajdonságokat más blokkoktól, de a tulajdonságaik és működésük egyedileg is beállítható. A blokkok hat kategóriába sorolhatóak:

- anyagáram objektumok,
- erőforrások,
- általános objektumok,
- mobil objektumok,
- listák és táblázatok,
- megjelenítő objektumok.

A Plant Simulation beépített saját programozási nyelve a *SimTalk*², amellyel az objektumok viselkedése szinte korlátlanul befolyásolható. A SimTalk segítségével létrehozott szkripteket *methodoknak* nevezzük, használatukkal a felhasználó széleskörű beállítási lehetőségekhez jut.

²A SimTalk az Eiffel programozási nyelven alapul, amely nagyon hasonló a többi általános célú programozási nyelvhez.

2. fejezet

A feladat ismertetése

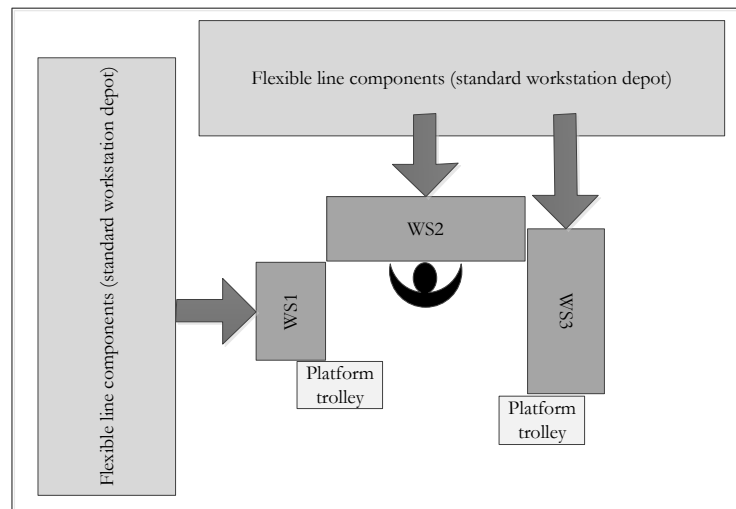
Mielőtt a szimulációs modell felépítését részletesen ismertetném, fontos, hogy pontosan bemutassam a dolgozat témáját, illetve a megoldandó feladatot.

2.1. A gyártórendszer bemutatása

A szimulációs modell egymultinacionális autóipari vállalat hazai gyárának egy, még tervezési fázisban levő gyártórendszeréről készül. A modell elkészítése ebben az esetben a tervezés első lépése, a célja pedig az, hogy az eredmények alapján javaslatot lehessen tenni a rendszer fizikai megvalósítására. Nagyon fontos tehát, hogy a modell validálható legyen, mivel egy nagyon komplex és nagy anyagi beruházást igénylő rendszerről van szó. A gyártott termékek a gépjárművekben fontos biztonsági funkciókat látnak el, vagyis a tervezés során kiemelt figyelmet kellett fordítani a minőségre. A modellben a termékek szerelési folyamatát lehet vizsgálni, a munkadarabok mechanikus megmunkálása korábban történik meg. A rendszer komplexitását az adja, hogy több mint negyven, korábban dedikált szerelősorként működő egységet kell összeszervezni egyetlen újrakonfigurálható rendszerré. Erre azért van szükség, mert a korábbi sorok fenntartása nagyon költséges feladat, és a területkihasználást is nagymértékben rontja, mivel egy egész épületet kell fenntartani az említett sorok miatt.

Az említett gyártósorokon jelenleg kis szériájú termékeket szerelnek, főleg olyanokat amelyek 10-30 éve még a gyártási kapacitás legnagyobb részét adták. A cég garanciavállalási kötelezettségei miatt még most is szükség van a sorok fenntartására annak ellenére, hogy azok gyakran hetekig nincsenek használatban. A szerelés nagyrészt kézzel történik, a leggyakrabban alkalmazott gépesített eszközök a prések, illetve csavarozók.

A termékek szerelése változatlan műveleti sorrenddel fog történni az új rendszerben is, a legnagyobb különbséget az jelenti majd a korábbi állapothoz képest, hogy a sorokat mobil



2.1. ábra. Az újrakonfigurálható szerelősor sémája

munkaállomásokból (*standard workstation*, *SWS*) lehet összeépíteni, így a szerelés megkezdésekor a megfelelő munkaállomásokat összetolják, majd a szerelés végén visszatolják őket a helyükre és elkezdődhet egy új sor konfigurálása valamint egy új termék szerelése. Az elképzelés szerint nem kell minden korábbi eszközt lecserélni, mivel a korábbi készülékek mindegyikét igyekszik megtartani a gyár, azonban már nem fixen lesznek az asztalokra szerelve, hanem szabványos felfogó felületekkel, így minden munkaállomáson használni lehet majd őket, a rendszer sokkal rugalmasabb lesz és sok hely felszabadul.

2.2. Modellépítés

A feladat megoldásának első, és talán legfontosabb lépéseként építeni kell egy szimulációs modellt, amely megfelelően reprezentálja az ismertett újrakonfigurálható rendszer működését. Ez a feladat jelenti a legnagyobb innovációt a dolgozatomban, mivel irodalomkutatási munkám során nem találkoztam hasonló, nyilvánosan is elérhető megoldással. A modellépítésnek automatikusan kell megtörténni a bemenő adatok alapján, mivel a rendszerben működő sorok folyamatosan cserélődnek. Az automatikus modellépítésnek tárolt bemeneti adatok alapján kell dolgozni, amelyek alapján a sor egyértelműen felépülhet, illetve az egyes munkaállomások paraméterei is beolvashatóak. A sorok működési paraméterei mellett szükség van még egy ütemezésre is, amely megadja, hogy éppen mely termékeket kell szerelni a rendszerben.

2.3. Szimuláció

A szimulációs vizsgálat ebben az esetben egy kapacitás-szimuláció, amelynek célja, hogy adott ütemezés esetén megadja az erőforrásigényt, illetve termékek legyártásához szükséges időt. Az ütemezés a cég által a rendelkezésemre bocsájtott historikus adatok alapján készült, megpróbáltam az általuk öt hónapra ütemezett feladatokat a szimulációs modellen elvégezni. A historikus adatok előnye, hogy ismert a jelenlegi gyártási idő, ezt összevetve a szimulációs idővel a modell validálhatóvá válik. További cél, hogy a bemenő paraméterek változtatásával együtt változó erőforrásigényt is meg tudja jeleníteni eredményként a szimulációs vizsgálat.

2.4. Az eredmények kiértékelése és javaslattétel

Az egész projekt kimenete a megbízó cég felé egy javaslattétel a megvalósításra, amely a szimuláció eredményei alapján készíthető el. A javaslatnak tartalmaznia kell, hogy milyen eszközállománnyal érdemes üzemeltetni a rendszert, illetve hogy egy műszakban hány dolgozóra van szükség a termeléshez. Optimalizálásra a tervezés korai fázisa miatt egyelőre nincs szükség, azonban a szimuláció eredményei alapján igyekszem a lehető legjobb megoldást javasolni a gyár számára.

3. fejezet

A rendszermodell felépítése

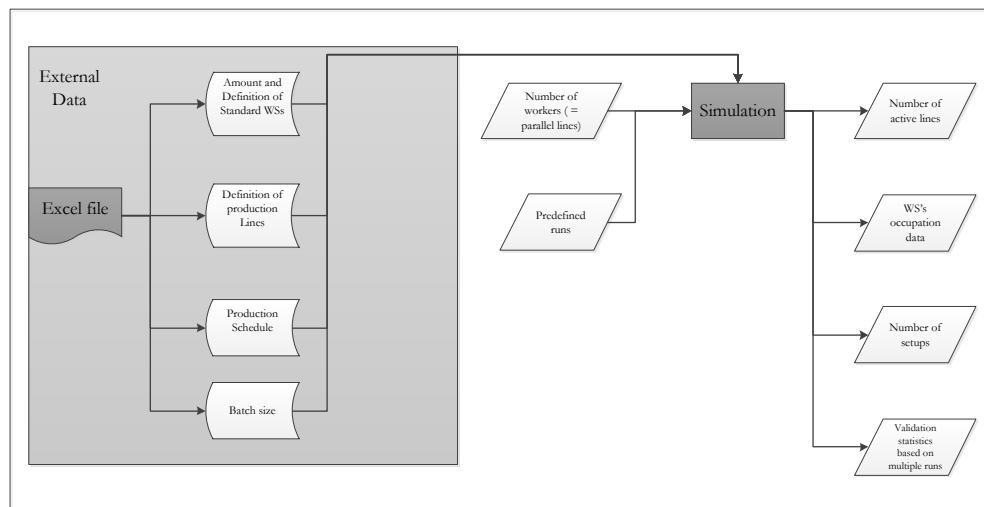
Modellezési szempontból a rendszer két fő részre osztható, az egyik az adatmodell, amelyben a szimuláció bemeneti értékei vannak tárolva, a másik a gyártórendszer DES modellje, amely tulajdonképpen a dolgozatom témája.

3.1. A szimuláció bemeneti értékei

A feladat megvalósítása során igyekeztem végig szem előtt tartani, hogy a szimulációt a gyakorlatban mérnökök, nem programozók fogják használni, ezért lényeges, hogy a paramétereket a modell, valamint a Plant Simulation mélyreható ismerete nélkül is kezelni lehessen. Ebben az esetben azonban nem egy grafikus kezelőfelületet készítettem, hanem megpróbáltam a modellt MS Excelben tárolt adatok alapján felépíteni. Ennek több gyakorlati oka is van:

- Az Excel egy mindenki által gyakran használt, könnyen kezelhető program, így a bemeneti értékek változtatása nem igényel speciális ismeretet.
- A bemenő adatok egyike a már említett ütemezési lista, amely méretéből adódóan nem kézzel bevitt értékek sorozata, hanem a vállalatirányítási rendszerből lekérdezett adathalmaz. Mivel az adatbázis jól strukturált, és folyamatosan frissül, így lehetőség van historikus értékek állandó frissítésére és tesztelésére.
- A Plant Simulation-ben az adatokat célszerű táblázatos formában tárolni az egyszerű kezelhetőség miatt, ezek a táblázatok pedig Excel fájlokból közvetlenül importálhatóak.

A bemeneti értékek tehát egyetlen Excel fájlban vannak tárolva, amelyet a szimuláció minden egyes futtatás során beolvas így mindig az aktuális értékek alapján végezhető el a vizsgálat. A szimuláció teljes adatmodellje a 3.1 ábrán látható, ahol jól megkülönböztethetőek a



3.1. ábra. A kimeneti és bemeneti adatok struktúrája

bemeneti- illetve kimeneti értékek. Néhány paraméter szabadon állítható a felhasználó igénye szerint, azonban a sorok összetételére vonatkozó adatok állandónak tekinthetők (amennyiben nincs jelentős változás a termék szerelésében).

A szimuláció bemeneti értékeit tartalmazó lines_summary.xls az alábbi munkalapokat tartalmazza:

- *sws munkalap*: A termelés során felhasználható munkaállomások legfontosabb adatait tartalmazza. Minden sora egy munkaállomás-típust tárol, amennyiben új típust szeretnének felvenni a listára (pl. újfajta csavarozógép, présgép stb.) elég csak egy új sort beszúrni, és feltölteni adatokkal. A munkaállomások fontos paraméterei (a szimuláció számára) az állomás száma, felépítése, megnevezése illetve a típusonként rendelkezésre álló mennyiség. A táblázat utolsó oszlopa a vizuális megjelenítést könnyíti meg, itt lehet megadni, hogy milyen színnel jelenjen meg az állomás a szimulációs modellben.
- *matrix_sws munkalap*: A sorok felépítése adható meg, mégpedig az előző lapon definiált standard munkaállomások segítségével. Ez azt jelenti, hogy a sorokat jelenlegi felépítésük és a rajtuk elvégzett feladatok alapján lebontottam munkaállomásokra, majd az így kapott szekvenciákat standard munkaállomások sorozatává alakítottam. Ezt elvégezve a rendszer minden elemére, a sorok előállnak munkaállomások sorozataként (a „#” karakter előtti szám a munkaállomás típusát jelzi). Annak ellenére, hogy a TDK dolgozatnak nem szerves része a folyamatfelmérés, fontos megemlíteni hogy a munkaállomások standardizálása, illetve a folyamatok elemzése rendkívül időigényes folyamat volt.

Ahhoz, hogy a sorok működését modellezni lehessen, nem elég a munkaállomásokat megadni, szükség van a hozzájuk tartozó műveleti időkre is. A munkaállomásokon végzett műveletek idejét a „#” karakter utáni számérték adja meg századpercben (a vállalatirányítási rendszerben is percben kifejezve szerepelnek a műveleti idők). Minden sor adatai előtt szerepel két további cella, az elsőben található érték a beállási idő percben kifejezve, a második pedig az úgynevezett *batch size*, ami megadja, hogy egy munkaállomáson egyszerre hány munkadarabon dolgoznak. A beállási idő tartalmazza a sorok összeépítéséhez szükséges időt, illetve az előszerelési műveletek idejét.

- *schedule2 munkalap*: Az ütemezési lista, amelynek első oszlopában a sor neve található, a másodikban pedig a gyártandó mennyiség. A lista egy makró segítségével készül a historikus adatok alapján, a sorösszeépítések sorrendjét az ütemezés sorrendje adja meg.

3.2. A rendszer DES modellje

Mivel a bemeneti értékek külső forrásból kerülnek beolvasásra, ezért a modellépítést célszerű az adatok importálásával kezdeni, a modell többi része ugyanis ezekkel az elemekkel fog dolgozni a továbbiakban. Az adatok importálását (illetve a modell legnagyobb részének működtetését) a már említett SimTalk *method*okkal oldottam meg.

Az első importáló eljárás a munkaállomásokat tartalmazó munkalap értékeit olvassa be egy úgynevezett tablefile-ba, amely a PlantSimulation-ben használt táblázat-objektum. A *method* tulajdonképpen a teljes „SWS” adattáblát átmásolja a workStations nevű táblába.

Ehhez hasonlóan oldható meg a sorok adatait tartalmazó táblázat feltöltése is, azonban az itt használt eljárás kicsit összetettebb az előzőnél, lévén, hogy a már bemutatott „matrix_sws” táblázat is jóval több adatot tartalmaz. A tablefile-ban való tároláshoz, illetve a későbbi beolvasáshoz érdemes átalakítani az adatszerkezetet úgy, hogy a létrehozott táblázat minden sora egy gyártósor adatait tárolja. Az egyetlen nehézség ebben az, hogy minden sorhoz tartoznak skalár illetve mátrix jellegű értékek is. Ez azt jelenti, hogy a beállási idő például tárolható egyetlen cellában a soron belül, míg a sorok felépítése (a munkaállomások típusa illetve a hozzájuk rendelt műveleti idő) egy táblázat jellegű adathalmaz. Ezt a problémát úgy oldottam meg, hogy a feltöltendő „Lines” táblázat egyik oszlopa táblázat típusú cellákat tartalmaz, vagyis minden egyes cella egy úgynevezett belső táblázat, amelyben már megfelelően tárolható a sor felépítése. A táblázat feltöltését a „fill_Lines_Tab” *method* végzi el, ennek pszeudokódja a következő:

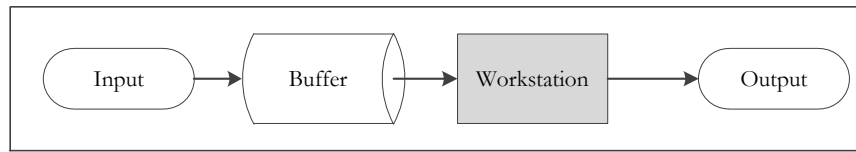
```
1 initialize tab, tab2, type, procTime, i, j
2 create Tab as a main table
3 import matrix_sws from lines_summary.xls
4   for i=1 to Tab.Ydimension do
5     if (i mod 2) not 0 then
6       insert lineName(i) from matrix_sws into Tab[Line, i]
7       insert sumOfSetupTimes(i) from matrix_sws into
          Tab[Sum_of_setup_times (min), i]
8       insert batchSize(i) from matrix_sws into Tab[Batch size, i]
9       create Tab2 as inner table
10      set Tab2.column header(1) as Workstation Type
11      set Tab2.column header(2) as Process Time
12      for j=6 to Tab.Xdimension do
13        if Tab[j,i] not void then
14          set Workstation Type as the string afore # in Tab[j,i]
15          set Process Time as the string after # in Tab[j,i]
16        next j
17      end if
18    next i
19 end method
```

3.1. táblázat. A *fill_lines* method pszeudokódja

Így tehát rendelkezésre áll az összes adat amelyek segítségével a sorok modellje automatikusan felépíthető. A harmadik importálást végző *method* az ütemezés beolvasására szolgál, felépítése teljesen hasonló az elsőhöz azzal a kivétellel hogy az importálandó munkalap itt a „schedule2”.

Miután minden importált adat rendelkezésre áll, létre kell hozni az összes munkaállomást; később ezeket fogom a szimulációs modellben paraméterezni, illetve összekötni egymással.

A korábbi fejezetekben nem esett szó róla, hogy a jelenlegi gyakorlat szerint egy-egy művelet befejezését követően (egy adott munkaállomáson) a termékeket valamilyen rekeszbe vagy kézikocsira teszik, mielőtt a következő műveletet elkezdenék. Ilyenkor a rekesz pufferként működik két munkaállomás között, amelyet a modellezés során is érdemes figyelembe venni. A munkaállomások modellezéséhez tehát létrehoztam egy önálló „WorkStation” nevű objektumot (frame-t) amelynek felépítése a 3.2 ábrán látható.



3.2. ábra. A *WorkStation* frame felépítése a szimulációs modellben

Az új frame-hez néhány új változót is definiáltam, amelyeknek a szimuláció során kell értéket adni, valamint figyelni is lehet a változókat, így a sorok összekötése egyszerűsödik. Ezek a tulajdonságok a következők:

- *description* (string): a munkaállomás leírása,
- *free* (boolean): ha az értéke hamis, akkor a munkaállomás éppen használatban van,
- *type* (string): a munkaállomás típusa,
- *used_in* (integer): annak a sornak a sorszáma (a *Lines* táblában), amelyben az állomás éppen üzemel,
- *y_pos* (integer): az állomás függőleges koordinátája a képernyőn.

A *create_SWS method* feladata, hogy létrehozza ezeket a munkaállomásokat a már korábban feltöltött *workStations* tábla alapján. A *method* szerkezete a következő:

```
1 initialize obj, name, i, j
2   for j=1 workstations.Y-dimension do
3     for i=1 to workstations.quantity do
4       set obj as a workstation
5       set obj.coordinates as x=20 y=20
6       set obj.size as 0,5
7       set name as "ws" + workStations["Name",j]
8       set obj.name as name "_" & i
9       set obj.description as workStations["description",j]
10      set obj.type as workStations[1,j]
11      set obj.curricon as workStations[6,j]
12      insert obj into createdObjects[j,i]
13    next i
14  next j
15 end method
```

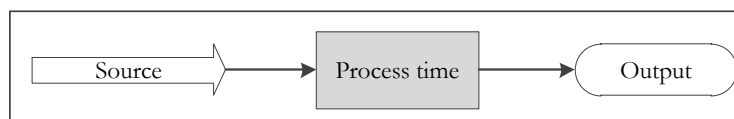
3.2. táblázat. A *create_WS method* pszeudokódja

A következő *method* a *creator* tulajdonképpen a szimuláció motorja, figyeli az ütemezésben, hogy melyik sort kell legközelebb összerakni, és ennek megfelelően meghívja a sor-összeépítő *methodot*. Mivel nagyon fontos, hogy ennek az eljárásnak a működése tökéletes legyen, ezért úgy döntöttem, hogy minden egyes feladathoz külön függvényt¹ írok amit a *methodnak* csak meg kell hívni a paraméterek segítségével. Ezt azért tartom előnyösnek, mert így sokkal kisebb a hibázás lehetősége mint egyetlen hosszú programkód esetén, illetve sokkal könnyebb később módosítani a működést. A *creator method* által hívott függvények funkciója a következő:

- *whats_next*: feladata hogy az ütemezésben következő gyártósor Lines táblázatbeli sor-számát adja értékül. A függvény a Schedule táblában végez keresést, az *sch_next* globális változó alapján.
- *free_space*: Ez a függvény, amely megvizsgálja hogy van-e szabad hely az új sor összekötéséhez (figyelembe véve a maximálisan üzemeltethető sorok számát). Ahhoz, hogy ezt meg lehessen tenni, létrehoztam korábban a Spaces táblázatot, amelybe azoknak a soroknak az azonosítója kerül amelyek éppen üzemelnek. Amennyiben tehát ezek száma kevesebb mint a maximálisan használható sorok száma, a függvény visszatérési értéke a szabad hely Y koordinátája. Abban az esetben ha minden hely foglalt, a függvény visszatérési értéke 0.
- *connect*: A szimulációs modell legbonyolultabb része, ez végzi el a rendszer legfontosabb feladatát: a sorok összekötését. Négy bemeneti paraméterrel rendelkezik, ezek mindegyikét a *creator method*tól kapja a már korábban említett többi függvény segítségével. A bemeneti paraméterek a következők: *lineID*, *amount*, *y_pos* és *batch* (lásd: 3.3 táblázat).

A függvény összeköti a *lineID* által megadott sort és elhelyezi az *y_pos* által adott szabad helyen. Ezt követően beállítja az Excelben megadott *batch* értéket valamint a gyártandó mennyiséget. A *connect* függvény a meghívás után először is megvizsgálja, hogy az adott sor már össze van-e rakva (a *used_objects* tábla tartalmát figyeli). Amennyiben a sor már üzemel, a függvény a valóságnak megfelelően jár el, vagyis nem rakja össze újra a sort, csak növeli azon a gyártandó mennyiséget. Ha a sor még nem üzemel, akkor a *method* a *pre_check* függvény segítségével eldönti, hogy lehetséges-e az összeépítés a rendelkezésre álló szabad munkaállomásokból. A *pre_check*-nek egyetlen

¹A SimTalk-ban a többi programozási nyelvhez hasonlóan nem csak eljárásokat, hanem függvényeket is létre lehet hozni, amelyek rendelkeznek ki- és bemenettel egyaránt. Ennek megfelelően a függvényhívás is a függvény névvel, illetve a bemeneti értékkel történik.



3.3. ábra. A *source frame* felépítése a szimulációs modellben

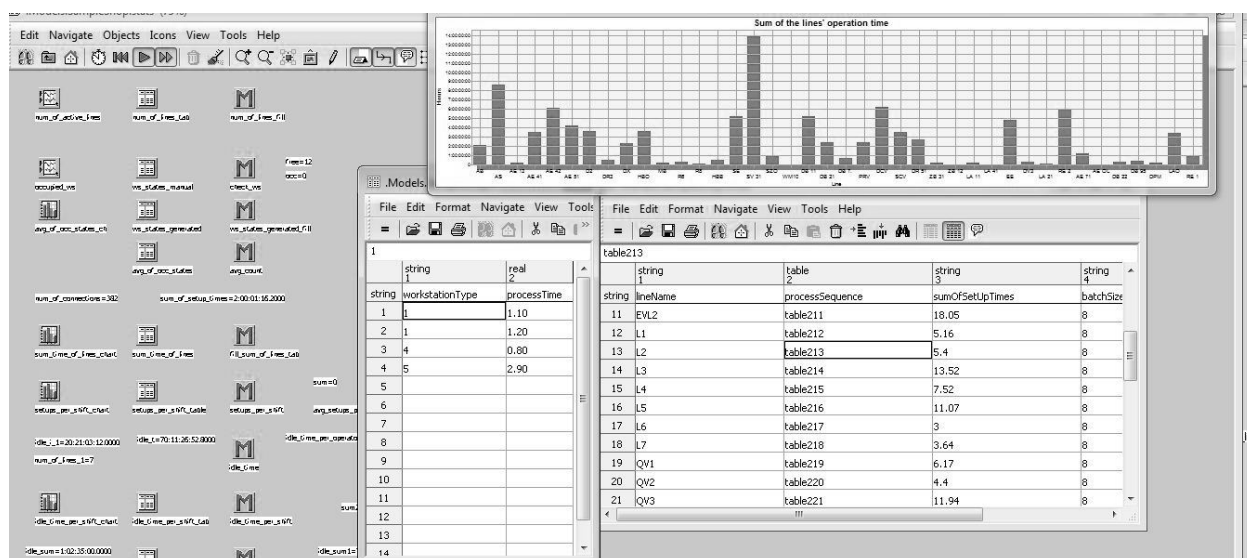
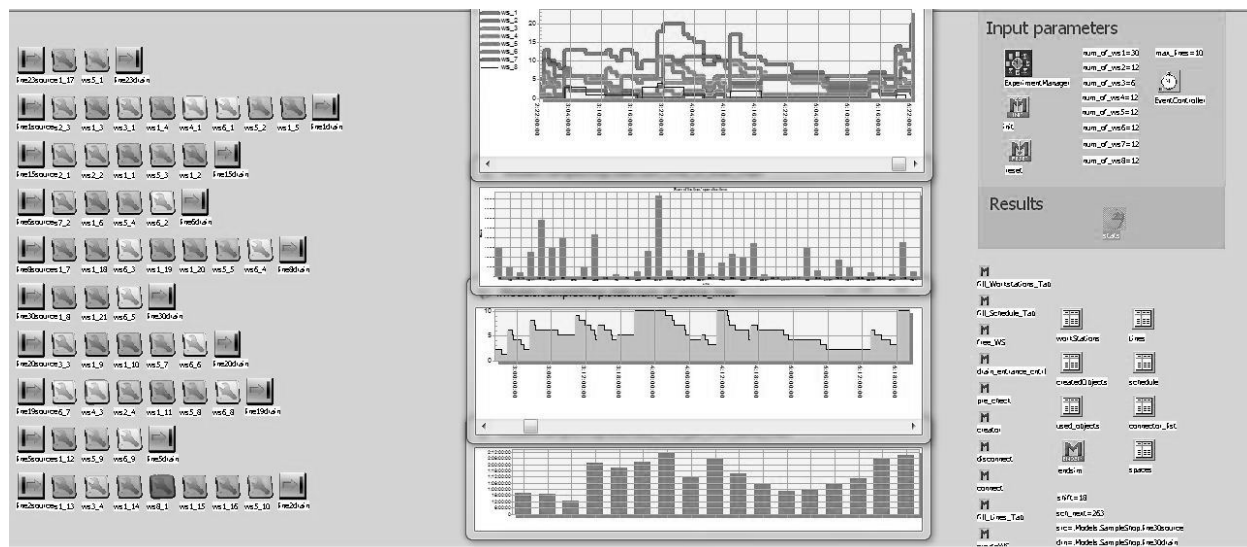
boolean típusú visszatérési értéke van, amely megadja, hogy a sort össze lehet-e építeni. Abban az esetben ha ez a feltétel teljesül, a *connect* megnöveli a sorösszeépítések számát tároló globális változó értékét, majd a szabad helyen elkezdni összeépíteni a sort.

Első lépésként a sor előtt kiírja a legfontosabb adatokat, mint például a sor nevét, illetve a *batch* értékét. Ez a lépés azt a célt szolgálja, hogy a szimuláció futása során a folyamatokat könnyen lehessen követni, ellenőrizni. Ezt követően a *method* meghívja a *free_WS* függvényt, azoknak az elérhető munkaállomásoknak az azonosítóját adja értékül amelyekre az aktuális sornak szüksége van. Ezeket rögtön be is írja a felhasznált objektumok táblába, beállítja az aktuális műveleti időket, az Y-koordináták értékét, valamint a *used_in* attribútumot. Így a sor összekötése megtörtént a szabad helyen, a gyártás elindításához már csak egy *source* illetve *drain* típusú objektumra van szükség.

A *source* objektum a munkaállomásokhoz hasonlóan egy általam definiált objektum a modellben, erre azért volt szükség hogy a beállási időket könnyebben kezelni lehessen. Az így létrehozott *source* objektum felépítése a 3.3 ábrán látható. A beállási idő egy *single proc* objektum segítségével van megadva, amelyhez a szükséges időt mindig csak az első munkadarab szerelése előtt állítja be a modell. A függvény emellett beállítja a gyártott termék típusát (megegyezik a sor nevével), valamint a gyártandó mennyiséget az importált adatok segítségével. A *drain* objektum a *source*-tól eltérően már nem speciális, a függvény csak a gyártandó mennyiséget, illetve a termék nevét állítja be, valamint a sor végére helyezi. Így tehát teljesen felépült a sor, amelyben elindulhat a termékek gyártása a megfelelő műveleti- illetve beállási időkkal. A függvény utolsó lépésként a létrehozott *source* illetve *drain* objektumokat beírja a *used_objects* táblába, valamint tárolja a kiértékeléshez szükséges statisztikai paramétereket. A *connect method* három további eljárást hív meg, amelyek a sorok üzemelését ellenőrzik. A *drain_entrance_control* feladata, hogy a leggyártott termékek mennyiségét figyelje, és a gyártandó mennyiséggel összehasonlítsa. Amint az utolsó termék is elkészült, meghívja a *disconnect* eljárást, ami a sor szétszedését végzi el. A *reset_buffer* eljárás az „x-piece flow” elvű gyártást segíti, mégpedig a munkaállomások között levő pufferek kimenetének szabályozásával. Ehhez nagyon hasonlít az *entrance* eljárás, amely a pufferek bemenetét zárja le, amint az „x-piece flow” által megadott mennyiség beérkezett.

3.2. A RENDSZER DES MODELLJE

A DES modell felépítése tehát az elvárásoknak megfelelően automatikusan történik, a Plant Simulation előre definiált elemei közül csak nagyon keveset használ fel mivel a működése szinte teljes mértékben a fent ismertetett eljárásokra épül. A *methodok* felépítése biztosítja a szimuláció automatikus lefutását is, a felhasználónak csak el kell azt indítani a „RUN” gombbal, futás közben azonban semmilyen külső beavatkozásra nincs szükség. A modell grafikus felhasználói felülete a 3.4 ábrán látható.



3.4. ábra. A modell felhasználói felülete

3.2. A RENDSZER DES MODELLJE

```
1 function arguments: lineID; amount; y_pos; batch
2 initialize obj1...obj6, i , temp_tab
3     set temp_tab as the process sequence of the line
4     if used_objects[lineID,1] not void then
5         increase source.number with amount
6         set source.exitlocked as false
7     else
8         create obj5 as new variable
9         insert their coords into spaces table
10    for i=1 to temp_tab.Ydim do
11        set obj1 as the return value of free_WS
12        set obj1.proctime as temp_tab[2,i]
13        insert obj1 into used_objects[lineID,i]
14        set obj1.used_in as lineID
15        set obj1.zoom as 1
16        set obj1.position as (100+(i*45),y_pos*65)
17        if i=1 then
18            create src as source
19            set src.used_in as lineID
20            set src.number as amount
21            set src.setuptime as lines[3,lineID]
22            set src.position as (100,y_pos*65)
23            create obj3 as connector between obj1 and obj 3
24        elseif (i>1) then
25            create obj3 as connector between obj1 and obj 2
26        elseif i=(temp_tab.Ydim) then
27            create drn as drain
28            set drn.amount as amount
29            set drn.used_in as lineID
30            set drn.position as (100+(i+1)*45,y_pos*65)
31            create obj4 as connector between obj1 and drn
32        end if
33        set obj2 as obj1
34        set obj2.used_in as lineID
35        insert obj3 into Connector_list[lineID,i]
36    next i
37    insert src into used_objects[lineID,i+1]
38    insert drn into used_objects[lineID,i+2]
39    increase num_of_lines with 1
40    increase sch_next with 1
41 end method
```

3.3. táblázat. A connect method pszeudokódja

4. fejezet

A rendszer szimulációs vizsgálata

A szimulációs modell önmagában nem nyújt segítséget a rendszer megvalósításához, hiszen a megfelelő szimulációs vizsgálatok elvégzése legalább akkora feladat, mint a modellépítés. A gyár munkatársaival egyeztetve megállapítottam, hogy többféle teszt szükséges a tervezett rendszer alapos ismeretéhez.

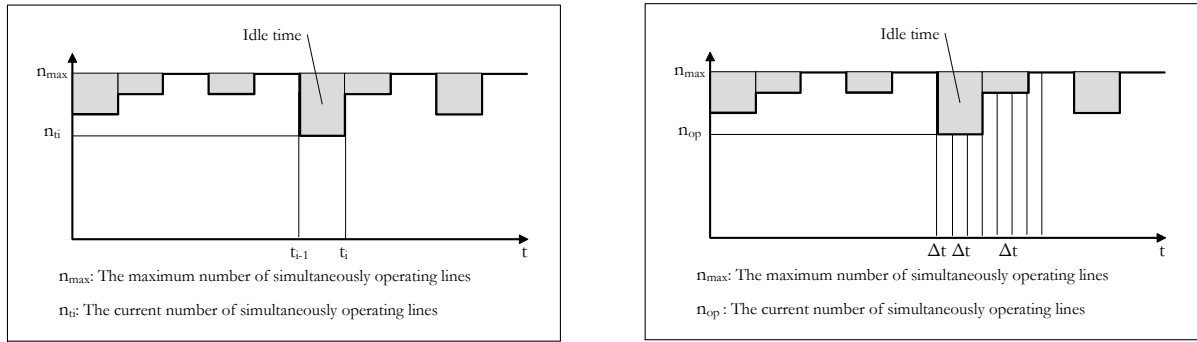
4.1. A legfontosabb szimulációs paraméterek

A rendszer megvalósításához szükséges szimulációs értékek kiválasztása főleg tapasztalaton alapuló feladat, hiszen szimulációs modellből kinyerhető korlátlan mennyiségű adat közül csupán néhányra van szükség a tervezés során. A Plant Simulation-ben *methodok* segítségével az adatok gyűjtése és szűrése is megoldható, ez azt jelenti, hogy futtatás közben a releváns értékeket táblázatba, illetve globális változóba lehet kiíratni. A beépített grafikonok segítségével a táblázatokban tárolt értékek automatikusan megjeleníthetők, a grafikonok a szimuláció során folyamatosan frissülnek.

Az első és legfontosabb lépés a kimeneteket kezelő eljárások programozásakor az, hogy eldöntsük, melyek a szimulációval meghatározható legfontosabb és leghasznosabb adatok. Természetesen a kimenetek megválasztása során is aktívan együttműködtem a gyár mérnökségével és együtt határoztuk meg a tervezéshez szükséges paramétereket. A létrehozott modell esetén a kimeneti értékek számára készítettem egy új, *stats* nevű framet, amely tartalmazza a szükséges *methodokat*, táblázatokat, globális változókat illetve grafikonokat.

A bemeneti értékek tárgyalásánál már ismertettem, hogy az egyik legfontosabb paraméter a gyártórendszer esetében az egy időben működő sorok maximális száma, amely jelen modell esetén megegyezik az operátorok számával. Ez azonban egy maximális érték, ami azt jelenti, hogy a termelés függvényében sok lehet az állásidő, amikor nem minden sor üzemel,

mert például nincs elég rendelkezésre álló munkaállomás. Éppen ezért az egyik legfontosabb statisztikai jellemző a működő sorok száma, amit grafikonon az idő függvényében jelenít meg a modell. Ahhoz, hogy ezt még könnyebb legyen számszerűen jellemezni, a gyárnál kíváncsiak voltak az ütemezésenkénti állásidő (*idle_time*) értékére is. Ezt a 4.1 ábrán a szürke téglalapok jelölik. A téglalapok területe megegyezik az állásidővel, mivel ha több mint egy sor nem üzemel egy adott időszakban akkor az állásidő is a többszörösére növekszik. Így kedvezőtlen esetben nyolc sor esetén például a gyártási időt megközelítő állásidőt is kaphatunk eredményként. Ezért szükségesnek találtam még a műszakonkénti állásidő, illetve az egy operátorra eső állásidő kiszámítását.



4.1. ábra. A veszteségidő kétféle számítási módja

A teljes állásidő számításánál egy viszonylag egyszerű módszerrel a téglalapok területét számoltam ki és ezeket adtam össze. Kihasználva az eseményvezérelt működést, a téglalapok területét mindig akkor számította ki a program amikor a t és a $t - 1$ időponthoz tartozó értékek is rendelkezésre álltak, így az állásidő (T gyártási idő esetén):

$$t_{idle} = \sum_{i=1}^T (n_{\max} - n_{t_i}) \cdot (t_i - t_{i-1})$$

Ez a módszer a műszakonkénti számításnál hibát okozna, hiszen ha egy sor több mint nyolc óráig nem üzemel, akkor eredményként azt kapnánk, hogy az egyik műszak esetén nem volt állás, míg a másik műszak eredménye a teljes kumulált érték lenne. Ennek kiküszöbölésére egy másik módszert találtam ki: egy területszámító algoritmust kellően rövid, fix időközönként (jelenleg $\Delta t = 5$ perc) meghív a szimuláció, és ott a maximális sorok számának és a működő sorok számának különbségével kiszámítja az állásidőt (4.1 ábra jobb oldala).

$$t_{idle} = \sum \Delta t \cdot (n_{\max} - n_{op})$$

Ezeket az értékeket nyolc óránként, vagyis műszakonként összegzi és táblázatban tárolja. Ennek nagy előnye, hogy nem kell tárolni a $t - 1$ időpontbeli értékeket viszont az algoritmus bizonyos hibával dolgozik. Elvégeztem néhány kísérletet ezzel kapcsolatban, és arra

az eredményre jutottam, hogy ez a hiba általában 1-2% körüli érték ami ebben az esetben elfogadható. Így tehát a szimulációs idő mellett (amit a rendszer automatikusan számol kimenetként) már rendelkezésre áll a veszteségidők értéke is.

Grafikon segítségével jelenítettem meg meg a felhasznált munkaállomásokat az idő függvényében, illetve az egyes sorok összes működési idejét az ütemezés alatt. Erre azért volt szükség, hogy meg lehessen találni a kiugróan sokat üzemelő sorokat. Fontos érték egy újra-konfigurálható rendszer esetén a sorösszeépítések, vagyis jelen esetben a beállítások száma. Ezt az értéket célszerű a vizsgálatok során a különféle paraméterek állításával minimálisra csökkenteni. A sorösszekötések számát (*num_of_connections*) a beállási időkkel soronként megszorozva megkapható az egy ütemezésre jutó beállási idők összege (*sum_of_setup_times*).

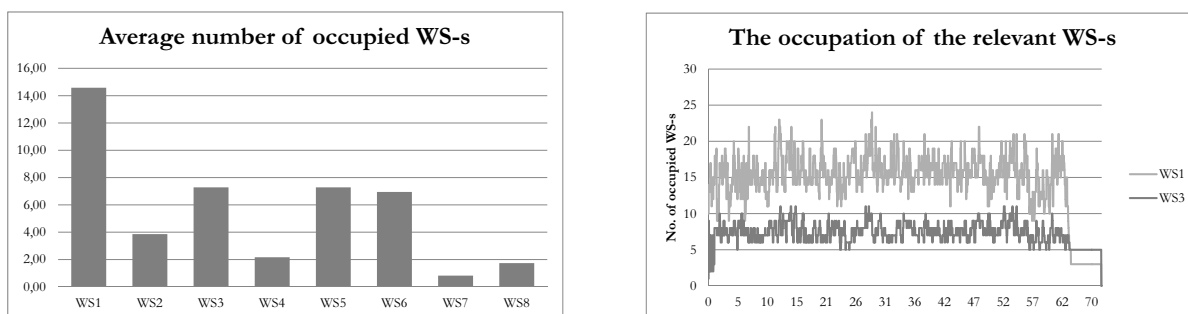
A legtöbb mért paraméter és adat esetén rendelkezésre áll a grafikonos és táblázatos megjelenítés is, így az adatok további feldolgozás céljából akár táblázatkezelőbe is exportálhatóak.

4.2. A kiértékeléshez szükséges vizsgálatok

A historikus adatok természetesen nagy segítséget nyújtanak abban, hogy a modellt a jelenlegi állapottal össze lehessen hasonlítani. A rendelkezésre álló adatok közül az egyik legfontosabb a jelenlegi gyártási idő, ennek ismeretében ugyanis meg lehet becsülni hogy az újrakonfigurálható rendszernek mekkora erőforrásigénye lesz, hogy tartani tudja-e a jelenlegi gyártási kapacitást.

4.2.1. Előzetes teszt

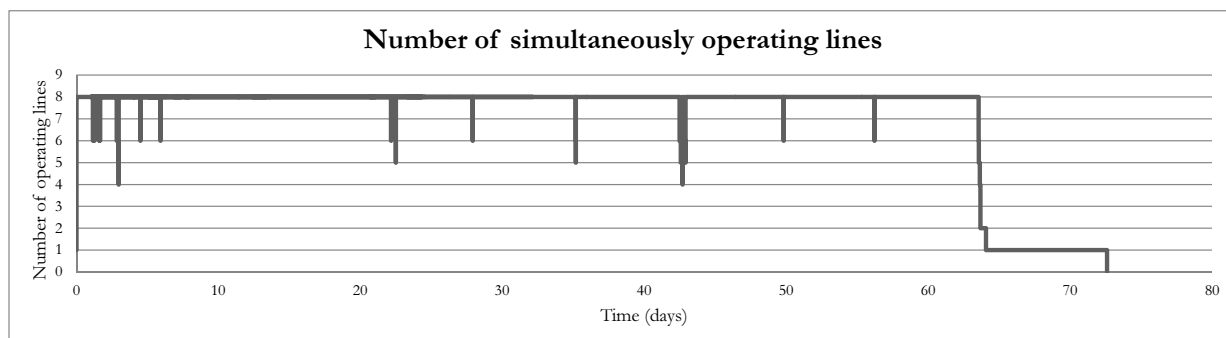
A szimuláció egyik célja tehát az erőforrásigény meghatározása, vagyis a munkaállomások számának meghatározása az egyes típusok esetén. Mivel nyolc különböző típus létezik, ezért rendkívül hosszú idő lenne az az összes típus mennyiségét kombinálva futtatni a szimulációt. Ennek elkerülése érdekében elvégeztem egy előzetes futtatást, amely során a munkaállomások számát tulajdonképpen végtelen mennyiségűnek tekintettem (mindegyikhez 40-es értéket állítottam be), és kizárólag az egy időben üzemelő sorok számát korlátoztam. Itt a cég mérnökei által javasolt 8-as értéket választottam, amely azt feltételezi, hogy a gyártórendszerben egyszerre nyolc operátor dolgozik („x-piece flow” esetén). Az első futtatás során arra voltam kíváncsi, hogy melyek azok a munkaállomások amelyekből kiugróan többre volt szükség, illetve végig jobb volt a kihasználtságuk. A célgépek és az univerzális tesztgépek értékeit figyelmen kívül hagytam, hiszen a célgépek értéke egyrészt adott (három darab), a tesztgépek száma pedig mindig megegyezik a maximálisan üzemeltethető sorok számával.



4.2. ábra. A foglalt munkaállomások átlagos száma, valamint a releváns típusok kihasználtsága a gyártás során

A grafikonokat elemezve arra az eredményre jutottam, hogy a WS1, illetve WS3 típusból a többinél többre van szükség (ezek a kézi csavarozók illetve a gépi prések) és a kihasználtságuk is jobb volt a többi típusénál (4.2 ábra).

A munkaállomások kihasználtsága mellett felfigyeltem még arra, hogy a gyártás 72 napot vesz igénybe, ami ugyan kevesebb a jelenlegi értéknél (két műszakos termelés esetén is csak 108 nap a jelenlegi 140-el szemben), azonban nagyjából 64 nap után az L5 sor kivételével az összes többi befejezi a gyártást. Ez a késleltetés a későbbi vizsgálatok során még nagy jelentőséggel fog bírni.



4.3. ábra. Előzetes teszt: az L5 sor hatása a gyártási időre

4.2.2. Érzékenység vizsgálat 1.

A következő, úgynevezett érzékenység-vizsgálat (*sensitivity analysis*) során arra voltam kíváncsi, hogy a rendszer hogyan reagál az általam fontosnak ítélt paraméterek megváltozására. Ezek a paraméterek az előző vizsgálat alapján a WS1 illetve WS3 munkaállomások száma, illetve az egyszerre üzemelhető sorok maximális száma. A Plant Simulation beépített *experiment wizard* nevű objektumával a vizsgálat egyszerűen automatizálható mivel rendelkezik

4.2. A KIÉRTÉKELEÉSHEZ SZÜKSÉGES VIZSGÁLATOK

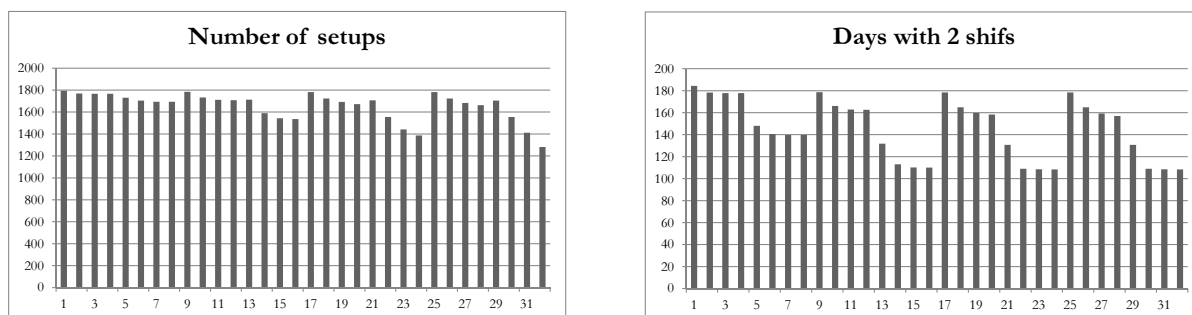
	WS1 (csavarozó)	WS3 (gépi prés)	Sorok max. száma
Minimum	12	6	6
Maximum	24	10	12
Lépésköz	4	4	2

4.1. táblázat. Az első érzékenység vizsgálat bemenő paraméterei

multi-level experiment opcióval, ami azt jelenti hogy meg kell adni a bemeneti változók minimális és maximális értékét valamint a lépésszámot, és a program ebből generál egy bemeneti táblázatot amely tartalmazza az előző paraméterek összes kombinációját (4.1 táblázat).

A táblázat alapján a lehetséges kombinációk száma 32, vagyis a szimuláció ennyiszer fog lefutni miközben a bemeneti értékek folyamatosan változnak. A szimuláció kimenete ebben az esetben a sorösszeépítések száma, valamint a gyártási idő (4.4 ábra).

Megjegyzés: A 4.4-4.11 ábrák vízszintes tengelyén a szimulációs futás sorszáma található.



4.4. ábra. A sorösszeépítések száma, illetve a gyártási idő az első érzékenység vizsgálat esetén

Az eredmények alapján megállapítható, hogy a bemeneti paraméterek kiválasztása megfelelő volt, mivel az eredmények szórása meglehetősen nagy érték (4.2 táblázat). Éppen ezért nagyon fontos olyan bemenet kombinációt választani, amellyel rövid gyártási időt lehet megvalósítani alacsony erőforrásigény mellett. A két grafikon jellegében nagyon hasonló, a paraméterek hatása mindkettőn jól követhető. Érdekes azonban itt is kitérni arra a korábban

	Összeépítések száma	Gyártási idő (2 műszak)
Átlag	1656,4	145,4
Szórás	129,2	27,1
Terjedelem	513	75,9

4.2. táblázat. Az első érzékenység vizsgálat statisztikai eredményei

már tárgyalt tényre, hogy az L5 sor napokkal megnöveli a gyártási időt míg más sorok már befejezték a termelést. Mivel a rendszer teljesítményét látványosan rontja, és van lehetőség a jövőben is dedikált sorként üzemeltetni (a nagy éves mennyiség miatt továbbra is szükség van rá), ezért a következő vizsgálatban nem szerepel az L5 sor. Ezzel a lépéssel a gyártási idő drasztikusan rövidülhet, illetve a sorok kihasználtsága is valószínűleg nőni fog.

4.2.3. Érzékenység vizsgálat 2.

Az előzőekhez képest nem csak az ütemezés változott meg, hanem a bemeneti paraméterek eloszlását és a kombinációk számát is megváltoztattam. A minimum és maximum értékeket illetve a lépésközt is megváltoztattam ez pedig együtt járt azzal, hogy a kombinációk száma is 60-ra nőtt (4.3 táblázat). A szimulációs eredmények az előzetes várakozásoknak megfelelően alakultak (4.5 ábra).

	WS1 (csavarozó)	WS3 (gépi prés)	Sorok max. száma
Minimum	10	6	6
Maximum	21	10	10
Lépésköz	3	2	1

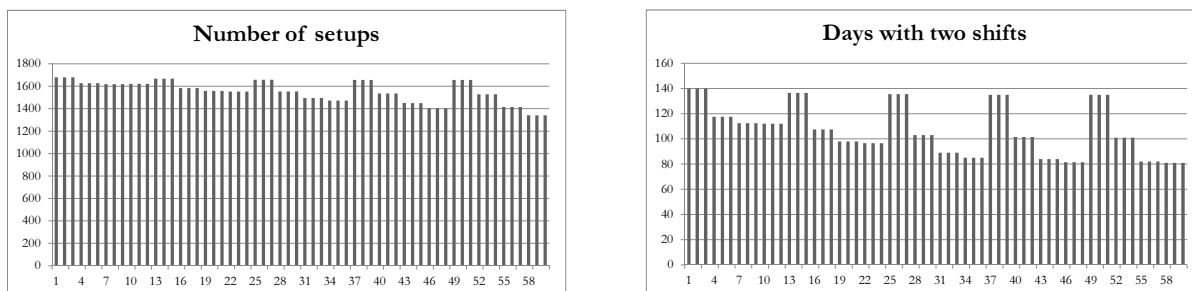
4.3. táblázat. A második érzékenység vizsgálat bemenő paramétere

Mivel a második vizsgálatot az előzetes információk alapján kellően pontosra tudtam hangolni, érdemesnek tartottam minél több hasznos információhoz jutni a futtatás révén. Ennek megfelelően az előzőtől eltérően itt már több kimeneti paramétert állítottam be.

	Összeépítések száma	Gyártási idő (2 műszak)
Átlag	1553,8	106,7
Szórás	96,7	20,25
Terjedelem	338	59

4.4. táblázat. A második érzékenység vizsgálat statisztikai eredményei

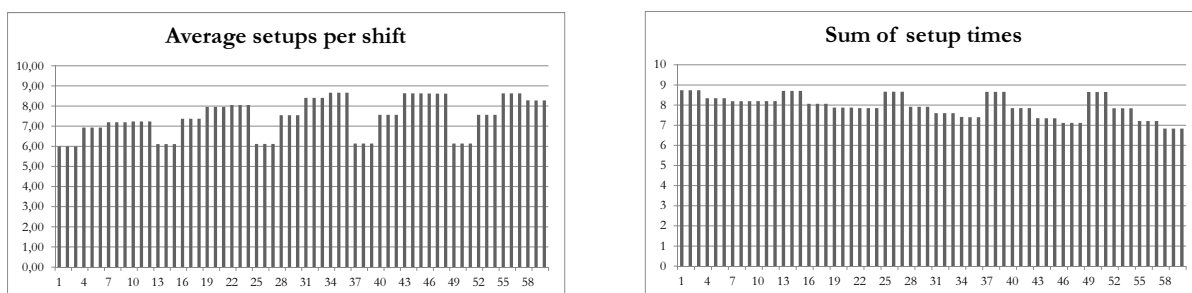
Az eredmények alapján elmondható (4.4 táblázat), hogy a rendszer a várakozásoknak megfelelően viselkedett, a sorösszeépítések száma csökkent, amellyel együtt jár az is hogy a gyártási idő is sokkal rövidebb lett. A két érték alapján számított statisztikai mutatók mindegyike javult, vagyis a rendszer stabilabban reagál a paraméterváltozásokra, kevésbé befolyásolja az erőforrások száma. A grafikonokon megfigyelhető még egy nagyon fontos dolog: a oszlopok hármas csoportokba rendeződtek amelyek egyenlő magasak. Ez azt jelenti



4.5. ábra. A sorösszeépítések száma, illetve a gyártási idő a második érzékenység vizsgálat esetén

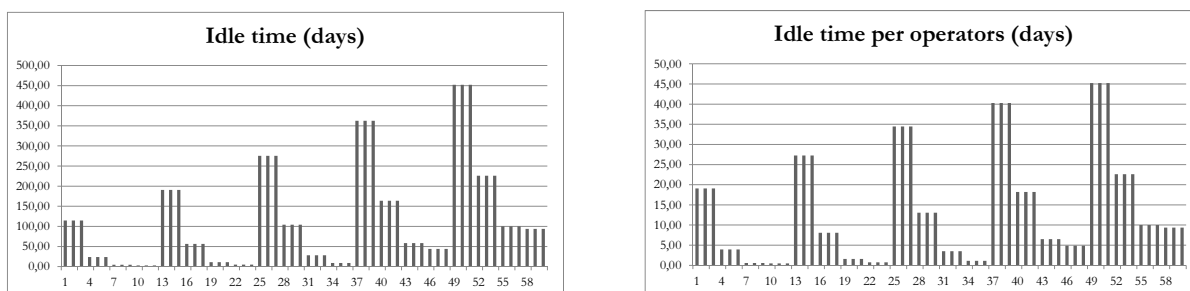
(a bemeneti paramétertábla alapján), hogy a WS3 számának változására a rendszer már egyáltalán nem reagál, így a továbbiakban már elég csak a másik két paraméter hatását figyelni. A jelenség magyarázata a figyelmen kívül hagyott L5 sor összetételében keresendő: mivel az L5 sorhoz öt darab WS3 típus szükséges és a sor szinte végig üzemelt, ezért a többi sornak a fennmaradó WS3 mennyiséggel kellett gazdálkodnia. Ez akár a gyártási időt is meghosszabbíthatta (a legrövidebb és a leghosszabb gyártási idő közötti eltérés 40%-os) abban az esetben, ha egy sort éppen amiatt nem lehetett összeépíteni mert nem volt elég rendelkezésre álló gépi prés. Ez a hatás az összes mért paraméter grafikonján jól megfigyelhető.

Fontos mért jellemző még ebben az esetben az átlagos műszakonkénti sorösszeépítések száma, valamint az összeépítés és beállási idők összege is, ez ugyanis a gyártási időt drasztikusan megnövelheti. (4.5 ábra).



4.6. ábra. A műszakonkénti sorösszeépítések átlagos értéke, illetve az összeépítések idejének összege

Megfigyelhető, hogy a műszakonkénti összeépítések száma az erőforrások számával együtt növekszik, érdemes azonban itt is figyelembe venni az előző grafikonokat, hiszen a műszakonkénti beállítások átlagos értéke nem az összeépítések számának növekedése miatt lesz magasabb, hanem a műszakok számának csökkenése miatt (4.6 ábra). Ez az érték azonban még így is ideálisnak mondható, hiszen nem éri el a 9-et, ami azt jelenti hogy nyolc operátorral

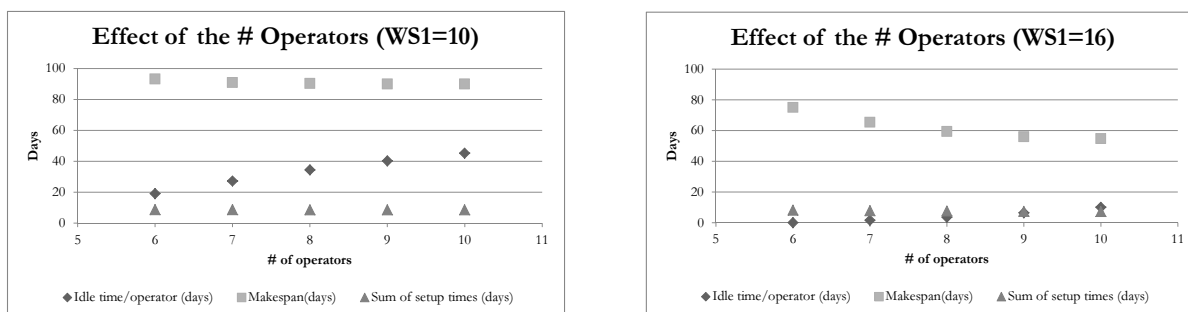


4.7. ábra. A veszteségidők értéke a bemenő paraméterek változása esetén

számolva műszak közben mindegyik sort egyszer kell átépíteni.

Az összeépítések idejeinek összege együtt csökken az összeépítések számával, amit a sorok maximális mennyiségének növekedése okoz. Ez egyértelmű, hiszen ha több sor üzemelhet, akkor lehet hogy egy sor még nem fejezte be a termelést de az ütemezésben már újra rá kerül a sor, így nincs szükség a szétszedésére.

A második érzékenység vizsgálat esetén még két további érték ingadozását mértem: a veszteségidőt, illetve az egy operátorra jutó veszteségidőt. Ebben az esetben az vártam, hogy az erőforrások és sorok számának növekedésével a veszteségidő csökkenni fog, hiszen ha több csavarozó áll rendelkezésre, akkor ritkábban fordulhat elő az az eset, hogy egy sort amiatt nem lehet összeépíteni mert nem áll rendelkezésre elegendő WS1 állomás. Ez a hatás jól megfigyelhető az értékek ingadozásán, azonban észre lehet venni egyfajta növekvő tendenciát is (4.7 táblázat). Az előbb említett hatás csak kevés operátor esetén működik, mert a munkaállomások számának növekedése nem arányos az operátorok számának növekedésével. Erre a magyarázat az, hogy sorok számának növelése változatlan erőforrás-mennyiség mellett növeli a veszteségidőt. Hozzájárul még a teljes veszteségidő értékéhez az a tény is, hogy az L5 sor kihagyása után még mindig maradt olyan sor amely a többihez képest aránytalanul sokat üzemel (lásd: BR2) Mivel ezek a soroknak a gyártási ideje a minimális összeépítési szám esetén is változatlan, a többi sor gyártási idejének lerövidülése esetén a veszteségidő értéke növekedni fog. Talán a 4.8 ábrán lehet legjobban megfigyelni az előbb ismertetett hatást: a bal oldali grafikonokon jól megfigyelhető, hogy nem jelent jó megoldást ha kizárólag az operátorok számát növeljük kis erőforráskészlet mellet. Ez ugyanis azt eredményezi, hogy az új operátoroknak nem lesz elegendő eszközük a gyártáshoz, ami nagyon megnöveli az állásidőt, míg a gyártási idő alig változik. A jobb oldali grafikon esetén a csavarozók száma már elegendően nagy ahhoz, hogy az operátorok számát is növelni lehessen. Amennyiben tehát rendelkezésre áll elegendő munkaállomás, a működő sorok száma már megnövelhető és ez pozitív hatással van minden termelési mutatóra, hiszen mint látható, a gyártási idő drasztikusan csökkenni kezdett, az állásidő pedig alacsony maradt.

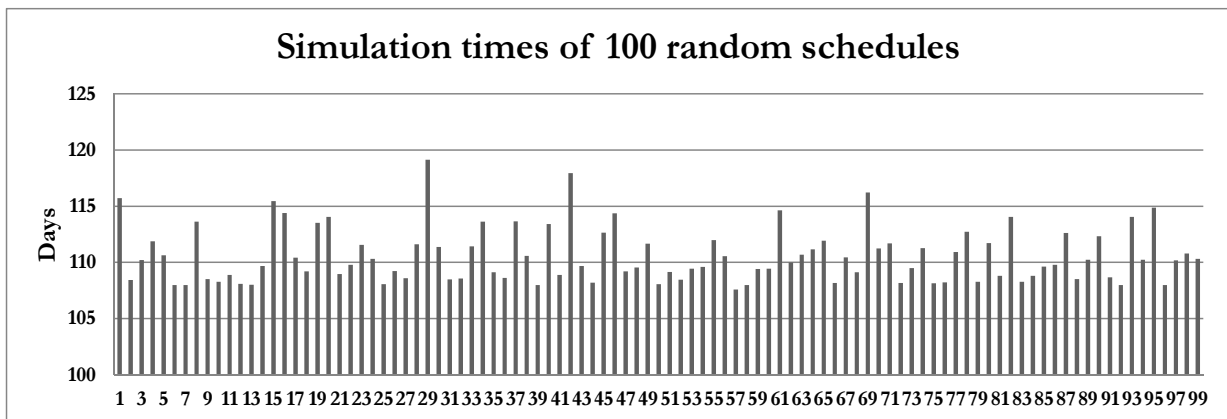


4.8. ábra. Az operátorok számának hatása a gyártási és beállási időkre

4.2.4. Az ütemezés hatása a teljes rendszerre

Az általam preferált szimulációk másik típusa az ütemezés megváltozásának a modellre gyakorolt hatását vizsgálja. A rendelkezésre álló historikus adatok alapján generált ütemezésben a gyártott mennyiségeket változatlanul hagyva, egy új *method* segítségével véletlenszerű ütemezéseket generáltam. Minden ütemezésnek ugyanannyi sora van tehát mint az eredetinek, egyedül a sorrend változott meg. Az új ütemezéseket egy táblázat celláiban tárolja el a *method*, ezekre egy *pointerrel* lehet rámutatni a szimuláció során. A vizsgálatot ebben az esetben is Experiment Manager-rel készítettem, ahol egyetlen bemeneti változóként a *pointer* értéket adtam meg. Ez egytől százig lépked így a szimuláció is százszor fut le. Az első tesztet próbaként szerettem volna kipróbálni, ezért csak egy paramétert állítottam be kimenetként, a gyártási időt. Ezzel az volt a szándékom, hogy felmérjem a teszt szükségességét és kipróbáljam hogy van-e számottevő hatása az ütemezés megváltozásának az újrakonfigurálható rendszer esetén. Az ütemezések tartalmazták az összes sort, a korábban már kihagyott L5-t is. A futtatás előtt azt jósoltam, hogy a gyártási idő számottevően változni fog, annak függvényében hogy az ütemezésben ugyanazok a sorok egymás mellé kerülnek, ezáltal a gyártási idő a összeépítés kihagyásával drasztikusan csökkenhet (a gyártási idők átlaga 110 nap).

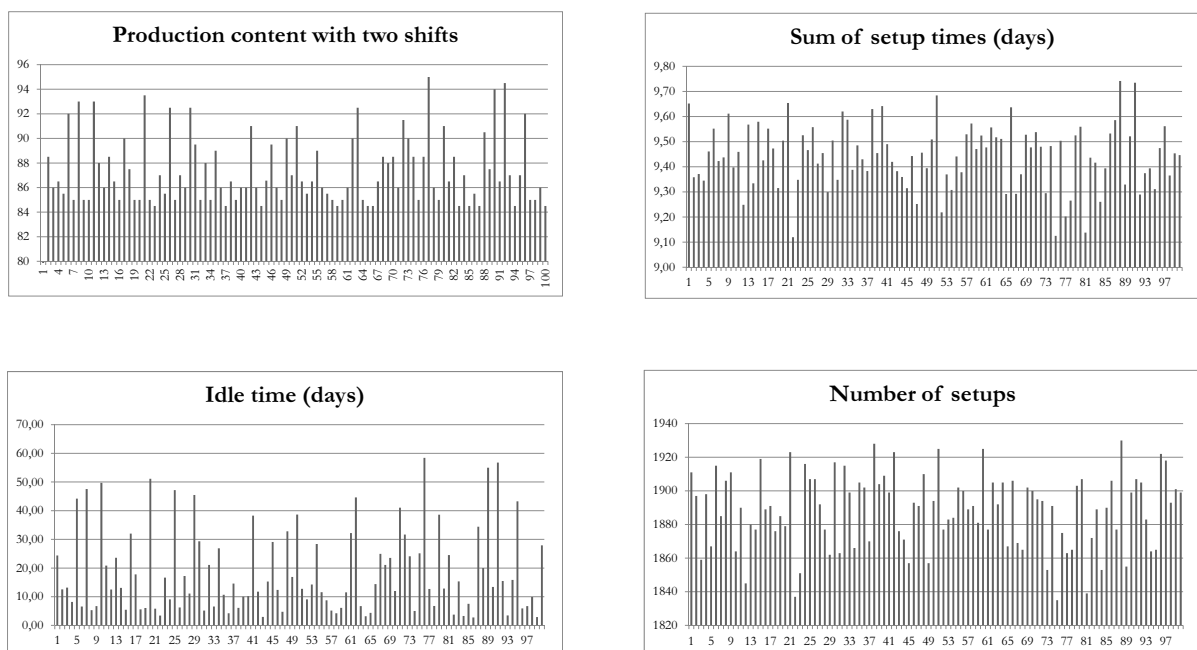
A várakozásoknak megfelelően két műszakos termelés esetén a gyártási idő ingadozása az ütemezés függvényében meglehetősen nagy (4.9 ábra). Érdekes ismét megemlíteni, hogy a gyártási idő a historikus adatok alapján 140 nap volt a jelenlegi rendszerben, vagyis az újrakonfigurálható rendszer esetén még ezek az értékek is megfelelőek lennének. Azonban ha figyelembe vesszük a statisztikai mutatószámokat akkor kiderül, hogy megfelelő ütemezés esetén rengeteg időt, ezzel együtt pedig pénzt is meg lehet takarítani. A minta terjedelme 11,5 nap, ami 110 napos gyártási idő esetén 10%-ot jelent, amit megfelelő ütemezéssel meg lehet spórolni.



4.9. ábra. A ütemezés hatása a gyártási időre (első teszt)

4.2.5. Az ütemezés hatása a rendszerre az L5 sor nélkül

Az előző teszt tapasztalatai alapján érdemesnek tartottam az ütemezés és a rendszer kapcsolatának további vizsgálatát. A második vizsgálat során az ütemezésben már nem szerepelt az L5 sor, valamint beállítottam több kimeneti paramétert is, hogy az ütemezés összes hatása megfigyelhető legyen. Ezek az összehasonlíthatóság érdekében megegyeznek a második érzékenységi vizsgálat paramétereivel. Mivel itt is véletlenszerű értékekről van szó, ezért nem vártam hogy a statisztikai mutatók megváltoznak, azonban a gyártási idő drasztikus rövidülése egyértelműen elvárás volt.



4.10. ábra. Az ütemezés hatása a termelési mutatókra (második teszt)

A 4.10 ábra a gyártási idő rövidülését igazolja, valamint jól látszik, hogy az ütemezés még mindig elég nagy hatással van az összes releváns termelési mutatóra. Érdekes még kiemelni a veszteségidő alakulását az ütemezés függvényében, hiszen itt tapasztalható a legnagyobb ingadozás. Megfigyelhető, hogy ideális esetben minimálisra lehet csökkenteni az értékét, egy rossz ütemezés hatására azonban az állásidő akár a hatszorosára is növekedhet a többi paraméter változatlansága mellett.

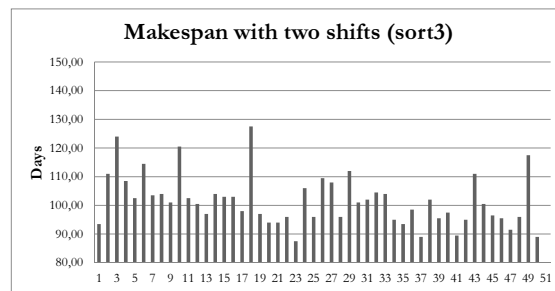
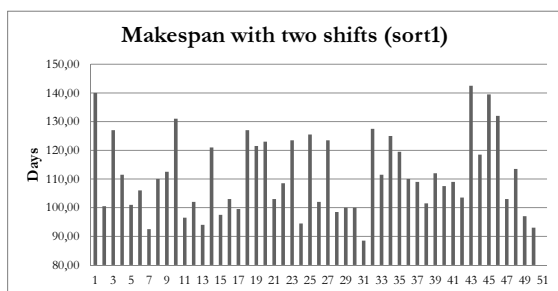
4.2.6. Az ütemezés hatása a rendszerre: sorrendiség

Az előző két vizsgálat eredményei alapján elhatároztam, hogy teszek néhány célzott kísérletet az ütemezés javítására, illetve a gyártási idő lerövidítésére. Ezt az ütemezés rendezésével lehet megpróbálni, hiszen ha egy termék célzottan csak egyszer, kétszer vagy háromszor szerepel a listában, akkor a sorösszeépítések száma ezzel együtt pedig a gyártási idő is lecsökkenhet.

Mivel nem optimalizálásról vagy ütemezési feladatról van szó, ezért itt is van bizonyos véletlenszerűség a listában. Erre azért van szükség, hogy meg lehessen vizsgálni a sorrendiség szerepét összevont tételek esetén is. A feladat megoldását azzal kezdtem, hogy készítettem három ütemezési listát:

- *sort1*: Minden sor csak egyszer szerepel az ütemezésben, ekkor a teljes mennyiséget legyártják.
- *sort2*: Minden sor kétszer szerepel a listában, a teljes gyártandó mennyiség véletlenszerűen van két részre osztva
- *sort3*: Minden sor háromszor szerepel a listában, a teljes gyártási mennyiség véletlenszerűen van három részre osztva.

Ebből a három listából generáltam mindhárom esetben ötven, véletlenszerű ütemezést amelyeket Experiment Manager segítségével adtam meg a szimuláció bemeneteként.



4.11. ábra. Az ütemezési tételek összevonásának hatása a gyártási időre

A legfontosabb mutató ebben az esetben is a gyártási idő, a 4.11 ábrán az egy-, illetve háromtételű összevonás látható. A bal oldali grafikon jól mutatja, hogy a tétel teljes összevonása nem jelent megoldást a gyártási idő lerövidítésére, hiszen rossz sorrend esetén nagyon elhúzódhat a termelés. A három tétellel történő összevonás esetén a gyártási idő szórása kisebb, illetve az átlag is alacsonyabb. Megfigyelhető egy kiugróan alacsony érték is, amely azt bizonyítja, hogy a rendszer megvalósítása esetén érdemes optimalizálni az ütemezést, mert rengeteg tartalék rejlik még benne.

A vizsgálat eredményeit összegezve: Érdemes nagyobb gyártási mennyiségeket beütemezni, az új rendszer esetén, ami jelentheti azt is, hogy minden terméket csak 3-4 havonta szerelnék nagyobb mennyiségben. Mivel a gyártási időt sem csökkentik, valamint a raktárkészletet is nagyon megnöveli az éves mennyiség egyszeri legyártása, ezért érdemes elgondolkodni az évente 3-4 alkalommal történő gyártáson. Az ütemezés átalakítása során pedig az összevonás mellett sokkal fontosabb a tétel sorrendjének optimalizálása.

5. fejezet

Az eredmények értékelése és a dolgozat összefoglalása

A következő néhány bekezdésben szeretném összefoglalni a szimulációs eredményeket, és ezek alapján levonni a legfontosabb következtetéseket. Az összefoglalásban ismertetem a dolgozat során elért eredményeket, illetve a jövőre vonatkozó feladatokat a témával kapcsolatban.

5.1. Javaslat a rendszer magvalósításához

Az előzői vizsgálatok rengeteg adatot szolgáltatnak az új rendszer vélt működéséről, amelyek segítségével könnyebb javaslatot tenni a rendszer megépítésére vonatkozóan. A szimulációs vizsgálat eredményeit valamint az ezek alapján tett javaslatokat igyekeztem tételszerűen összefoglalni a könnyebb áttekinthetőség érdekében:

- I. Az előzetes szimuláció alapján látható, hogy ha egyes sorok a többinél kimagaslóan többet üzemelnek, akkor a gyártási idő és ezzel együtt a veszteségidő is nagyon megnőhet. Belátható, hogy ez egy újrakonfigurálható rendszer esetén nem megengedhető (lásd: 4.2.1 fejezet). Ez kiküszöbölhető úgy, ha a problémát okozó sorokat nem vonjuk be az új rendszerbe hiszen feleslegesen csökkentenék annak eredményességét. A gyár mérnökeivel egyetértettünk abban, hogy amennyiben az említett sorok esetén is cél a helymegtakarítás akkor érdemes azok layoutját lean-módszerek segítségével áttervezni.
- II. Az érzékenység-vizsgálatok közül a második kiemelkedő jelentőséggel bír, a cég számára ez szolgáltatta a legtöbb információt. Az vizsgálat kimutatta, hogy elég két bemenő paraméter (WS1 és az operátorok száma) számát változtatni a rendszer optimalizálásához. Ezek hatásai azonban erős korrelációban van egymással, mint azt már koráb-

ban bemutatam (4.8 ábra). Amennyiben az anyagi keretek lehetővé teszik, az állítható nyomatéktartományú csavarozókból mindenképpen érdemes legalább 16-18 darabot beüzemelni, hiszen ebben az esetben a dolgozóknak nem kell várni a szabad eszközökre és az állásidő drasztikusan csökkenni fog. A működő sorok és ezzel együtt az operátorok számát érdemes 8-9 körülre megválasztani, így a kívánt helymegtakarítás biztosan elérhető és a gyártás is kellően gyors lesz. A többi munkaállomás számát az előző két érték ismeretében könnyebb meghatározni, hiszen például az univerzális tesztgépek és az operátorok száma ideális esetben egyenlő. A célgépek száma és mérete lehetővé teszi azt, hogy ne kelljen őket új és drága eszközökre cserélni.

- III. Az utolsó vizsgálatok esetén beláttuk, hogy az ütemezés nagyon komoly hatással van az újrakonfigurálható rendszerre. A gyár mérnöksége is komolyan elgondolkozott azon, hogy az új rendszer esetén már teljesen új ütemezést fognak alkalmazni, céljuk a már korábban ismertetett évente 3-5 alkalommal történő szerelés. Nem elég azonban összevonni ezeket a feladatokat hanem a sorrendjüket is célszerű megvizsgálni (a szimuláció segítségével ez most már könnyen megtehető), és a legmegfelelőbb sorrend alapján kell a termelést megszervezni.

5.2. Az elért eredmények értékelése

A szimulációs modellt úgy építettem fel, hogy az jól reprezentálni tudja az újrakonfigurálható rendszer működését, emellett célorientált és könnyen kezelhető lett. A gyár munkatársai ezáltal egy olyan hatékony eszközhöz jutottak amely bizonyítja azt, hogy a rendszer létrehozható és képes az elvárásoknak megfelelően működni. A vizsgálatok megmutatták, hogy az újrakonfigurálható rendszer képes adott idő alatt elvégezni a szerelési feladatokat, és megfelelően ki tudja váltani a jelenleg működő 48 dedikált szerelősort. Ez egyben azt is jelenti, hogy a célkitűzéseknek megfelelően óriási helymegtakarítás érhető el: ha a rendszer a jelenlegi elképzelések szerint épül meg, akkor a becslések szerint akár 60-70%-os helymegtakarítás is elérhető, ezzel együtt pedig a gyártási költségek is drasztikusan csökkenni fognak.

A gyár mérnöksége a közeljövőben elkezdheti a gyártórendszer pontos megtervezését, ami azt jelenti hogy akár egy éven belül elkezdődhet a telepítés. A 2. fejezetben kitűzött célokat tehát sikerült elérni, azonban nem szabad elfelejteni hogy ezek mellett még rengeteg megoldatlan feladat maradt amivel érdemes a közeljövőben foglalkozni.

5.3. Nyitott kérdések a témával kapcsolatban

A gyártórendszer megépítése és a jelenlegi termelés átszervezése hatalmas befektetés még egy multinacionális vállalat számára is, ezért érdemes további részletes, célorientált tesztek futtatásával támogatni a tervezést. Minden felmerülő kérdést, még a néha jelentéktelennek tűnőket is érdemes megvizsgálni, hiszen a szimulációnak éppen az a legnagyobb előnye hogy ezeket gond nélkül meg lehet tenni, ezzel pedig egyre több hibát ki lehet küszöbölni. Amint megszületnek az első konkrét elképzelések a gyár részéről, érdemes az új adatokkal is futtatni a szimulációt, valamint a historikus adatok folyamatos frissítése is nagyon fontos. Az ütemezést az előző fejezetben ismertetett módon érdemes átalakítani és megvizsgálni a szimuláció segítségével. A rugalmas gyártósorokhoz hasonlóan ki lehet dolgozni mutatószámokat a rendszer rugalmasságára illetve újrakonfigurálhatóságára vonatkozóan. Ezek alapján minősíteni kell a rendszert, hogy egy átalakított terv esetén a kettőt össze lehessen hasonlítani, és ki lehessen választani a körülményeknek megfelelőt.

5.4. A feladat folytatása

A feladatot a TDK dolgozattal még nem szeretném lezárni, szeretném további témákkal, valamint az előző kérdésekre adott válaszokkal kiegészítve diplomatervként folytatni. A célom egy olyan komplex módszer kidolgozása, amely hatékonyan támogatja az újrakonfigurálható rendszerek tervezését illetve kivitelezését.

Irodalomjegyzék

- [1] ELMARAGHY, H. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems* 17 (2005), 261–276. 10.1007/s10696-006-9028-7.
- [2] ELMARAGHY, H. A. *Changeable and Reconfigurable Manufacturing Systems*. Springer, 2009.
- [3] KOREN, Y., HEISEL, U., JOVANE, F., MORIWAKI, T., PRITSCHOW, G., ULSOY, G., AND BRUSSEL, H. V. Reconfigurable manufacturing systems. *CIRP Annals - Manufacturing Technology* 48, 2 (1999), 527 – 540.
- [4] LANDERS, R., MIN, B.-K., AND KOREN, Y. Reconfigurable machine tools. *CIRP Annals - Manufacturing Technology* 50, 1 (2001), 269 – 274.
- [5] LAW, A. M., AND KELTON, D. W. *Simulation Modelling and Analysis*. McGraw-Hill Education - Europe, Apr. 2000.
- [6] LOTTER, B., AND WIENDAHL, H.-P. Changeable and reconfigurable assembly systems. In *Changeable and Reconfigurable Manufacturing Systems*, H. A. ElMaraghy, Ed., Springer Series in Advanced Manufacturing. Springer London, 2009, pp. 127–142.
- [7] PAGE, B., AND KREUTZER, W. *The Java simulation handbook*. Shaker, 2005.
- [8] PFEIFFER, A. *Novel Methods for Decision Support in Production Planning and Control*. PhD dissertation, Budapesti Műszaki és Gazdaságtudományi Egyetem, Budapest, 2007.
- [9] TSUBONE, H., AND HORIKAWA, M. A comparison between machine flexibility and routing flexibility. *International Journal of Flexible Manufacturing Systems* 11 (1999), 83–101. 10.1023/A:1008096724273.

- [10] WIENDAHL, H.-P., ELMARAGHY, H., NYHUIS, P., ZÄH, M., WIENDAHL, H.-H., DUFFIE, N., AND BRIEKE, M. Changeable manufacturing - classification, design and operation. *CIRP Annals - Manufacturing Technology* 56, 2 (2007), 783 – 809.
- [11] WOMACK, J. P., JONES, D. T., AND ROOS, D. *The Machine That Changed the World : The Story of Lean Production*. Harper Perennial, Oct. 1991.