



MISKOLCI EGYETEM

GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR

HATVANY JÓZSEF INFORMATIKAI TUDOMÁNYOK DOKTORI ISKOLA

**KITERJESZTETT MODELLEK ÉS MÓDSZEREK
ERŐFORRÁS-KORLÁTOS
TERMELÉSÜTEMEZÉSI FELADATOK MEGOLDÁSÁRA**

PHD ÉRTEKEZÉS

KÉSZÍTETTE:

DR. KULCSÁRNÉ FORRAI MÓNIKA

OKLEVELES MÉRNÖK-INFORMATIKUS

TÉMAVEZETŐ:

DR. ERDÉLYI FERENC

A MŰSZAKI TUDOMÁNY KANDIDÁTUSA

A DOKTORI ISKOLA VEZETŐJE:

PROF. DR. VÉGH JÁNOS

A FIZIKAI TUDOMÁNYOK DOKTORA

MISKOLC, 2017

NYILATKOZAT

Alulírott Dr. Kulcsárné Forrai Mónika büntetőjogi felelősségem tudatában kijelentem, hogy a Hatvany József Informatikai Tudományok Doktori Iskolába beadott PhD értekezés önálló munkám eredménye, az irodalmi hivatkozások egyértelműek és teljeseek.

Dátum: 2017. szeptember 4.

.....
Dr. Kulcsárné Forrai Mónika

Köszönetnyilvánítás

Köszönetet mondok témavezetőmnek Dr. Erdélyi Ferenc ny. egyetemi docensnek, aki kiemelkedő elméleti tudásával és gyakorlati tapasztalataival kutató munkámat irányította és támogatta, és a PhD értekezésem megírását hasznos tanácsokkal és javaslatokkal segítette.

Köszönöm a Miskolci Egyetem Alkalmazott Informatikai Intézeti Tanszék teljes kollektívájának támogatását, külön megköszönöm Dr. Kulcsár Gyula egyetemi docens, Prof. Dr. Tóth Tibor professor emeritus, Dr. Dudás László egyetemi docens (korábbi tanszékvezető), és Dr. Nehéz Károly egyetemi docens (jelenlegi tanszékvezető) kollégáim szakmai támogatását.

Köszönetet mondok a Hatvany József Informatikai Tudományok Doktori Iskola tanárainak és oktatóinak, akik hozzájárultak szakmai fejlődésemhez.

Köszönöm az ipari partnerek (Electrolux Lehel Hűtőgépgyár, Jászberény; Fehrer Hungaria Járműipari Kft., Mór) közreműködését és támogatását. Külön köszönöm Rigó István logisztikai csoportvezető együttműködését.

Köszönetemet fejezem ki családtagjaimnak biztatásukért és támogatásukért.

A PhD értekezésben ismertetett kutató munka az EFOP-3.6.1-16-00011 jelű „Fiatalodó és Megújuló Egyetem – Innovatív Tudásváros – a Miskolci Egyetem intelligens szakosodást szolgáló intézményi fejlesztése” projekt részeként – a Széchenyi 2020 keretében – az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

Summary

In my dissertation, I summarize the results of my research work on modeling and solving resource-constrained scheduling problems. My investigations were focused on modeling and practice-oriented solutions of the different classes of extended scheduling problems related to flexible make-to-order (MTO) manufacturing. The dissertation presents the analyzed discrete manufacturing-assembly systems, their associated production, preparation and logistical processes, the elaborated new resource-constrained scheduling models, the solving approaches, the main algorithms and the newly developed groups of scheduling objective functions.

In addition to the set of machine selection (assignment, allocation) and sequencing decision variables that are typical of classical scheduling problems, I also consider the constraints and decision variables that cover the tasks of shift assignment, management of preparation processes and auxiliary resources. These extensions cause considerable difficulty when solving scheduling problems. In each case, the allocation of resources and the integrated fine scheduling of the timely execution of operations and other (preparatory and logistical) activities were accomplished with self-developed novel solutions. These solving methods are based on the combined usage of heuristic and exact constructive algorithms and on rule-based scheduling procedures, as well as advanced multi-operator and multi-objective local search techniques with simulation supported evaluation.

In Chapter 1 of the dissertation, I describe the topic of my research work, the purposes of the research and the applied research methods. In Chapter 2, I review fundamental models for production scheduling related to the research.

Chapter 3 summarizes the production fine scheduling model developed and analyzed in the first section of my research. It is typically suitable for synchronizing discrete component manufacturing and assembly phases of large household appliance production. The most important features of the modeled resource-constrained manufacturing-assembly system are: alternative technological routes; limited availabilities of machines; capacity-constrained intermediate buffers; different processing, set-up and material handling times; as well as jobs with strict due dates. Solving this problem includes the assignments of jobs to resources, scheduling the jobs, and the assignments of shifts to resources.

Chapter 4 presents the modeling and solving of a complex resource-constrained production fine scheduling problem that is typical in the production of vehicle fittings and component parts. One of the features of the problem is that multiple shared accessible resources (machines, attachment points, carriers, and tools) have to be allocated simultaneously, and multiprocessing tasks have to be scheduled in order to fulfill production orders within strict due dates. Another feature of the problem is that a combined task of production scheduling and inventory control has to be solved simultaneously in such a way that the technological main processes and even the tool-preparatory assistant processes have to be fine scheduled by considering their interconnections. My new integrated solving approach is based on a two-level decision hierarchy. At the lower level, the preparation-oriented built-in scheduling sub-problem is

transformed into a resource environment that consists of parallel virtual machines whose number varies with time. In the transformed problem space, independent jobs appear with integer release times, integer due dates and unit processing times. The execution of these jobs is scheduled in order to minimize the maximal lateness. A new solving algorithm is able to produce an optimal solution of the sub-problem in polynomial running time. At the top level of the decision hierarchy, I use the solution of the sub-problem to meet the tool preparation constraint in the search algorithm that solves the full problem. The subsections of Chapter 4 describe in detail the most important features of the analyzed full production fine scheduling problem, the formulated optimization model, the approach of the developed solving method and the most important algorithms.

Further chapters of the dissertation summarize the new scientific results, their practical applications, and my publications in these topics. The dissertation is closed by the list of references. The appendices contain some typical running results referred to in the text of the dissertation.

Összefoglalás

A dolgozatomban összefoglalom az erőforrás-korlátos ütemezési feladatok modellezésével és megoldásával kapcsolatban végzett kutatómunkám eredményeit. Vizsgálataim az igény szerinti rugalmas gyártás (Make to Order, MTO) különbözően kiterjesztett ütemezési feladattípusainak modellezésére és gyakorlat-orientált megoldására irányultak. Bemutatásra kerülnek a vizsgált diszkrét gyártó-szerelő rendszerek, a hozzájuk kapcsolódó termelési, előkészítési és logisztikai folyamatok, a kidolgozott új erőforrás-korlátos ütemezési modellek, a megoldási koncepciók, a fontosabb algoritmusok és a kifejlesztett új ütemezési célfüggvény-csoportok.

A klasszikus ütemezési feladatokra jellemző gépválasztási (hozzárendelési, allokációs) és sorrendi döntési változók halmazán túlmenően figyelembe vesszem a műszakbeosztásokra, az előkészítési folyamatok megszervezésére, valamint a kiegészítő erőforrásokra (auxiliary resources) vonatkozó korlátozásokat és döntési változókat is. Ezek a kiterjesztések jelentős mértékben megnehezítik az ütemezési feladatok megoldását. Az erőforrások allokálását, az operációk és az egyéb (előkészítési és logisztikai) tevékenységek időbeli végrehajtásának integrált finomprogramozását minden esetben új, saját fejlesztésű megoldási módszerekkel valósítottam meg. Ezek a módszerek alapvetően heurisztikus és egzakt felépítő algoritmusok, szabályalapú ütemezési eljárások valamint szimulációval támogatott továbbfejlesztett többoperátoros és többcélú lokális keresési technikák kombinált alkalmazására épülnek.

A dolgozat 1. fejezetében ismertetem a kutatómunka témakörét, a kitűzött kutatási célokat és az alkalmazott kutatási módszereket. A 2. fejezetben röviden áttekintem a kutatáshoz kapcsolódó alapvető termelésütemezési modelleket.

A 3. fejezetben összefoglalom a kutatómunkám első szakaszában kidolgozott és vizsgált termelésprogramozási modelleket, amely jellemzően háztartási nagygépek diszkrét alkatrészgyártási és szerelési fázisainak szinkronizálására alkalmas. A modellezett erőforrás-korlátos gyártó-szerelő rendszer legfontosabb jellemzői közé tartoznak az alternatív technológiai útvonalak, a korlátozottan rendelkezésre álló gépek, a korlátozott méretű műveletközi tárolók, az eltérő műveletvégzési, átállítási és anyagmozgatási idők valamint a szigorú belső határidős munkák. A bemutatott probléma megoldása magába foglalja a munkák erőforrásokhoz rendelését, ütemezését és az erőforrások műszakbeosztásának meghatározását.

A 4. fejezetben bemutatom egy összetett – a járműipari szerelvények és részegységek gyártásánál tipikus – erőforrás-korlátos termelésprogramozási feladat modellezését és megoldását. A feladat egyik jellegzetessége, hogy többféle osztott hozzáférésű erőforrást (gépeket, csatlakoztatási pontokat, szerszámhordozókat és szerszámokat) kell egyidejűleg kiosztani és párhuzamos végrehajtású munkákat kell ütemezni szigorú határidős gyártási rendelések teljesítése érdekében. A probléma egy másik sajátossága, hogy a termelésütemezés és a készletgazdálkodás egy kombinált feladatát kell egyidejűleg megoldani úgy, hogy a technológiai főfolyamat mellett a szerszám-előkészítési mellék-folyamatokat is önállóan ütemezni kell. Az integrált megoldási koncepció egy kétszintű döntési hierarchiára épül. Az alsó szinten az előkészítésre vonatkozó beépülő részproblémát egy transzformációs eljárás segítségével leképezem egy olyan erőforrás-környezetre, amely időben változó számú, párhuzamosan működő virtuális gépekből áll. A transzformált problématerben egész-értékű

indítási és befejezési időpontokkal határolt, egységnyi műveleti idejű független munkák végrehajtását ütemezem a késések minimalizálására törekedve. Egy új megoldó algoritmus a részfeladat optimális megoldását polinomiális futási idő alatt képes előállítani. A döntési hierarchia felső szintjén a részprobléma megoldását használom fel a teljes feladat megoldását végző kereső algoritmusban a szerszám-előkészítési korlát betartása érdekében. A 4. fejezet alfejezetei részletesen ismertetik a vizsgált teljes termelésprogramozási feladat legfontosabb jellemzőit, a megfogalmazott optimalizálási modellt, a kifejlesztett megoldási módszer koncepcióját és a legfontosabb algoritmusokat.

Az értekezés további fejezetei az új tudományos eredményeket, azok gyakorlati alkalmazását, valamint a témában készített saját tudományos közleményeket foglalják össze. A dolgozatot a hivatkozott irodalmak listája zárja. A mellékletekben néhány jellemző futási eredményt helyeztem el, melyekre a dolgozat szövege hivatkozik.

Tartalomjegyzék

NYILATKOZAT.....	2
Köszönetnyilvánítás	3
Summary	4
Összefoglalás.....	6
Tartalomjegyzék.....	8
1 Bevezetés.....	10
1.1 A kutatás tématerülete	10
1.2 A kitűzött kutatási célok.....	11
1.3 Az alkalmazott kutatási módszerek	12
2 A kutatáshoz kapcsolódó alapvető termelésütemezési modellek áttekintése	14
2.1 Alapvető jellemzők.....	14
2.2 Ütemezési feladatok változatossága az erőforrások és operációk szempontjából	16
2.3 Ütemezési feladatok változatossága a munkák szempontjából	17
2.4 Ütemezési feladatok változatossága az ütemezési célok szempontjából.....	18
3 Diszkrét alkatrészgyártási és szerelési folyamatok termelésütemezési és műszakbeosztási feladatainak integrált megoldása	20
3.1 A vizsgált feladat ismertetése	20
3.1.1 Szerelést közvetlenül megelőző alkatrészgyártási folyamat ütemezése	20
3.1.2 A termelési folyamat jellemzői	20
3.1.3 A termelési folyamat technológiai és logisztikai objektumai	21
3.1.4 Termelési politika, célok, elvárások és kritériumok.....	23
3.2 Egy új kiterjesztett modell az ütemezési és műszakbeosztási feladat integrált kezelésére ...	24
3.3 Megoldási módszer.....	25
3.3.1 Megoldási koncepció.....	25
3.3.2 Az alkalmazott szimulációs technikával kapcsolatos követelmények.....	28
3.3.3 Az integrált termelési finomprogramozás keresési algoritmusai.....	31
3.4 Futási eredmények és illusztratív példák.....	33
3.4.1 A többcélú összehasonlító operátor működése.....	33
3.4.2 A célfüggvény-értékek alakulása a keresési folyamatban	35
3.4.3 A vizsgált modell szimulációjának fontosabb jellemző eredményei.....	39
3.5 Összefoglalás és következtetések	44
4 Diszkrét gyártási folyamatok és előkészítő tevékenységek integrált termelésprogramozása	46

4.1	A feladat ismertetése	46
4.1.1	Alkatrészgyártási folyamatok és előkészítő tevékenységek	46
4.1.2	A termelés-menedzsment követelményei	48
4.2	Az IPSIC feladat modellezése	49
4.2.1	A modellezés szerepe	49
4.2.2	Adatszerkezetek és proaktív szemléletű integrált megoldási koncepció	50
4.2.3	Az IPSIC feladat formális optimalizálási modellje járműipari üléselemek gyártásának finomprogramozására	54
4.3	A konfiguráció-előkészítési tevékenységek ütemezése	64
4.3.1	A konfiguráció-előkészítések ütemezési problémája	64
4.3.2	A probléma transzformációja és egy új kiterjesztett ütemezési modell definiálása	65
4.3.3	Időtartalék-orientált megoldási módszer	67
4.3.4	Az optimális megoldás elérésének bizonyítása	68
4.3.5	A beépülő részfeladat és a teljes IPSIC feladat kapcsolata	71
4.3.6	Egy JIT-orientált megoldási módszer	72
4.3.7	A konfiguráció-előkészítési ütemterv rugalmasságának növelése egy továbbfejlesztett JIT-orientált algoritmussal	74
4.3.8	Következtetések és alkalmazási lehetőségek	76
4.4	A termelés főfolyamatainak ütemezése	77
4.4.1	A konfiguráció-cserék ütemezési problémája	77
4.4.2	Továbbfejlesztett többcélú és többoperátoros keresési algoritmus	78
4.4.3	A szomszédsági operátorok működése és szerepe	81
4.4.4	A szimuláció szerepe	84
4.4.5	Igény szerint változtatható optimalizálási célfüggvény-rendszer	87
4.4.6	Egy valós alkalmazási példa és néhány jellemző futási eredmény	89
4.5	Összefoglalás és következtetések	96
5	Új tudományos eredmények	98
6	Az eredmények hasznosítása	100
	Saját publikációk	101
	Irodalomjegyzék	102
	Melléletek	106
	I. Melléklet: A célfüggvény-értékek változása az EasyForce kereső algoritmus működése során	106
	II. Melléklet: A prioritási sémák szerepe	113
	III. Melléklet: Tanaka-benchmark feladatok megoldása MOMOTS algoritmussal	118
	New Scientific Results	121

1 Bevezetés

1.1 A kutatás tématerülete

Napjainkban a termelő vállalatoknak és termelési hálózatoknak egyre fokozódó kihívásokkal kell szembenézniük. A globalizált piaci környezetben a versenyképesség növelése érdekében rendkívül fontos, hogy a termelő vállalatok minél gyorsabban alkalmazkodjanak a körülmények változásaihoz. Ennek érdekében a termelési folyamatok hatékonyságát és rugalmasságát, valamint a szállítókészséget kell javítani a készletek, a termelési költségek és a késések (határidő túllépések) alacsony szinten tartása mellett.

Ezeket a feladatokat a termelés-menedzsment operatív szintű funkciói oldják meg. A korszerű vállalatirányítási tevékenység kiterjed a vállalat működésével összefüggő összes területre. A modern termelési, menedzsment és informatikai paradigmákra alapozva egy vállalat egészének működését és annak folyamatait a tervezés, az előkészítés, a végrehajtás és az ellenőrzés szakaszaiból felépített többszörösen visszacsatolt szabályzórendszerek irányítják.

A gyártási rendszerek egy részében a termékegyedek és a munkahelyek térben és időben elkülönülnek. Az ilyen diszkrét termelési folyamatok (pl. az előgyártmány-gyártás, alkatrészgyártás, előszerelés és szerelés) tervezése és irányítása főként többszintű, hierarchikusan egymásra épülő rendszerben valósul meg [6]. A magasabb szinten alkalmazott modellek hosszabb időszakra vonatkoznak és kevésbé részletesek és pontosak. Az alacsonyabb szinten a felsőbb szintről kapott eredmények (tervezési és irányítási döntések) alapján készített egyre részletesebb és egyre rövidebb időszakokra vonatkozó modellek segítségével hozzák meg a döntéseket. Előfordulhat, hogy az alsó szinteken megoldhatatlan feladat jelenik meg, ilyen esetekben visszacsatolással új vagy módosított megoldást kell kérni a felettes szintről [45].

A nagy rendszerekben a termelési feladatok dekomponálása érdekében elméleti szempontok alapján általában elkülönül a hosszú távú tervezés, a közép távú ütemezés valamint a rövid távú termelésprogramozás feladatköre [46]. A gyakorlatban a termékek tulajdonságai (pl. komplexitás, típusváltozatok stb.), a technológiai folyamatok jellemzői (pl. útvonalak és kapacitások stb.) és a termelési volumenek nagymértékben befolyásolják az alkalmazott szintek számát [8].

A termelésstervezés és –ütemezés elméleti eredményei általában közvetlenül nem alkalmazhatók a gyártás operatív irányítására, mivel az elkészített tervek összevontan kezelt erőforrásokra alapozott nagyvonalú megoldásokat alkotnak. A termelésprogramozás (fine scheduling) minden fontos részletre kiterjedő pontos végrehajtási finomprogramot készít rövid időhorizontra (pl. heti, napi, műszakonkénti bontásban). Az elkészített termelési finomprogramok megvalósítása érdekében a komplex döntéshozatalnak ki kell terjednie a termelés főfolyamatain túl (pl. gyártás, szerelés) a legfontosabb kapcsolódó mellék (pl. logisztikai) és segéd (pl. szerszámellátó) folyamatokra is.

A gyártásirányítás összetett irányítási funkcióhalmaz, amely valós időben irányítja és felügyeli a gyártórendszerben zajló folyamatokat. A gyártásirányítás az előre megtervezett, és

a belső rendelkezésekben megfogalmazott aktuális célokat igyekeznek megvalósítani, alkalmazkodva a fellépő bizonytalanságokhoz, és törekedve az eredmények adott feltételek (korlátozások) melletti optimalizálására. A gyártásirányítási rendszer gondoskodik a gyártási feladatok megfogalmazásáról, finomprogramozásáról, kiosztásáról, elindításáról, a kivitelezés irányításáról, valamint a végrehajtás felügyeletéről és minőségének biztosításáról [5]. A gyártásirányítás meghatározó funkciója a felettes termelésstervezési szinttől kapott középtávú termelési tervek rövidtávú finomprogramokká alakítása. A finomprogramozás során meg kell vizsgálni a szükséges technológiai tervek, anyagok és erőforrások rendelkezésre állását, meg kell vizsgálni a korlátozásokat és alternatív lehetőségeket. Ezt követően kerülhet csak sor az operációk és erőforrások egymáshoz rendelésére, továbbá az egyes munkák indítási sorrendjének és időpontjainak meghatározására. A gyártásirányításnak kell gondoskodnia az ütemezett feladatok végrehajtásához szükséges anyagok (alapanyag, komponensek, beépülő részegységek), erőforrások (személy, gép, szerszám, eszköz, stb.) és információk (tervek, programok, utasítások, adatok stb.) rendelkezésre állásáról is.

A diszkrét gyártási folyamatok – a terméktől függően – sokfélék és technológiájuk is nagyon eltérő lehet, ezért a gyártási finomprogramok gyors és optimalizált (automatikus vagy fél-automatikus) elkészítése megfelelő modelleket, algoritmusokat és implementált szoftvereket igényel. A finomprogramozási funkció hatékony megvalósításának fontos feltétele a felmerülő ütemezési feladatok pontos megfogalmazása és a megoldások értékelése.

Az ütemezési feladatok változatossága megnehezíti olyan modellek és szoftverek kidolgozását, amelyek könnyen és gyorsan adaptálhatók adott termelési rendszer gyártásirányításának támogatására. Az elméleti eredmények gyakorlati bevezetése ezért általában jelentős szakértői munkát igényel. Ennek hatékony támogatását az ütemezési feladatok modelljeinek, a megoldó módszereknek és a szoftver alkalmazások fejlesztésének kutatási eredményei támogatják.

1.2 A kitűzött kutatási célok

A diszkrét termelési folyamatok rövid távú (üzem, műhely, cella, gépcsoport, gyártósor vagy gép szintű) részletes ütemezési feladatainak megoldásához szükség van az adott termelési folyamat lehető legpontosabb modellezésére és hatékony ütemező rendszerekre, melyek a termelés változó feltételrendszeréhez és kritériumaihoz igazodva, végrehajtható és optimum közeli megoldásokat adnak.

A kutatás alapvető feladatai:

1. Az igény szerinti (Make To Order, MTO) rugalmas gyártás hatékonyság-növelésének vizsgálata, a követelmények és lehetőségek felmérése, analízise.
2. Ütemezési modellek kutatása és fejlesztése erőforrás-korlátos diszkrét termelési folyamatok rövid távú (napi és heti) ütemezésére az operatív termelés-menedzsment funkciók támogatására.
3. Ütemezési és döntéstámogatási algoritmusok fejlesztése.

A feladatok megoldásának megalapozása érdekében vizsgáltam az egyutas (Flow Shop, FS), a többutas (Job Shop, JS), valamint a párhuzamos gépekkel kibővített rugalmas (Flexible Flow Shop, FFS, Flexible Job Shop, FJS) ütemezési modellek alkalmazható változatait.

A kitűzött kutatómunka hatóköre kiterjed azokra a modellekre, amelyekben szerepet kapnak:

- az alternatív technológiai tervek,
- a párhuzamos gépi alternatívák,
- az erőforrások rendelkezésre állásainak időkorlátai,
- az emberi erőforrások korlátozó feltételei (műszakbeosztások),
- a változtatható és többszintű sorozatnagyságok,
- a műveletközi tárolók kapacitáskorlátai,
- az anyagmozgatási idők,
- a gépek és munkahelyek be- és átállítási műveletei (set-up),
- több, különböző teljesítmény-mutató (célfüggvény) együttes (multi-objective) figyelembevétele.

A nemzetközi szakirodalomban elérhető ismert ütemezési modellek és módszerek számos esetben nem adnak a felsorolt speciális követelményekkel kibővített feladatok esetében elfogadható megoldást, ezért az ilyenformán megfogalmazott erőforrás-korlátos, kiterjesztett diszkrét ütemezési feladatok kutatása mind elméleti mind gyakorlati szempontból fontos.

A figyelembe vehető módszerek – a feladat összetettsége és kombinatorikus jellemzői miatt – főként a heurisztikus (szabályalapú, felépítő) algoritmusok, a mesterséges intelligencia (keresési) módszerek, valamint a szimulációs modellek eszköztárából tevődnek össze. A feladatok fent vázolt sajátosságai indokolják a tudományos célú alap kutatás elméleti és a kísérleti fejlesztés tapasztalatainak valamilyen kombinált alkalmazását.

A kitűzött kutatási feladatok új modellek, algoritmusok és továbbfejlesztett megoldási módszerek kidolgozását, valamint a megoldások jóságának értékelését foglalják magukba. Az alkalmazott kutatás célja az elméleti eredmények adaptálása konkrét, megvalósított, vagy megvalósítandó rendszerekre (tudástranszfer alapú fejlesztés).

Az értekezés a kutatási eredmények alkalmazására ismerteti három különböző valós diszkrét gyártó-szerelő rendszer kiterjesztett erőforrás-korlátos ütemezési modelljét és annak tulajdonságait, bemutatja az ütemezési feladat megoldási koncepcióját, a fontosabb algoritmusokat, és a termelési teljesítményt mérő mutatószámok definícióját.

1.3 Az alkalmazott kutatási módszerek

A termelésütemezés és finomprogramozás végrehajtása, kezelése dinamikus modelleket igényel. A feladatok bonyolultsága és összetettsége miatt az ismert modellek bővítése és új modellek kidolgozása egymáshoz szorosan kapcsolódó tudományterületek eredményeire támaszkodva valósítható meg. A modellezés során alapvetően a termelésinformatika, a diszkrét matematika, az ütemezés-elmélet, valamint a mesterséges intelligencia modelljeinek, módszereinek és eredményeinek kombinált alkalmazására támaszkodtam. Kezdetben a szakirodalomban publikált modelleket tanulmányoztam és megvizsgáltam néhány új, a

gyakorlatban felmerült probléma sajátosságait. Ebben nagy segítségemre voltak a Miskolci Egyetem Alkalmazott Informatikai Intézeti Tanszék K+F+I projektjeiben résztvevő kollégáim. A fő kutatási irányok kitűzésében témavezetőm Erdélyi Ferenc javaslataira támaszkodtam. A vizsgált problémakört a tanulmányozott valós ipari feladatok igényei tovább konkretizálták. A megfelelő megoldásokat a megismert konkrét vállalati igényekre figyelemmel kerestem.

A kutatómunkám során a felmerült részfeladatokhoz kapcsolódó ismereteket összegyűjtöttem. Az ismert eredményekre alapozva kerestem alkalmas kiindulási modelleket. Ezt követően fogalmaztam meg az ütemezési feladatok és részfeladatok problémás, még meg nem oldott elemeit.

A szakirodalom már példák sokaságán keresztül igazolta, hogy a vizsgálatra ebben az értekezésben kitűzött ütemezési feladatoknál jóval egyszerűbb feladatok esetében sem lehet polinomiális számítási idő-költségű algoritmust készíteni optimális ütemezési megoldás megtalálására. Ezért elsősorban heurisztikus és meta-heurisztikus megoldási módszerekkel foglalkoztam. Olyan megoldó eljárás-mintákat kerestem, vizsgáltam, módosítottam és bővítettem, amelyek más feladatokban már eredményesek voltak.

A kiválasztott feladatok sajátosságaihoz illeszkedő új modellváltozatokat dolgoztam ki, és ezeket szimulációval elemeztem. A tapasztalatok alapján módosítottam, finomítottam és pontosítottam a modellek részleteit, a feladatmegoldás koncepcióját és egyes eljárás-elemeit. Megfelelő eredmények elérése után szigorúbb követelményrendszerrel és kiterjesztett célok kitűzésével bővítettem a modelljeimet.

A kidolgozott modellek implementálásában nagy segítségemre volt intézetünk termelésinformatikai kutató csoportja. Mérnök- és gazdaságinformatikus hallgatók, TDK, projekt-feladat, szakdolgozat és diplomamunka keretében – Kulcsár Gyula vezetésével – sok alkalmazás-fejlesztési részfeladatot oldottak meg az általam kidolgozott új modelleken. Így például eljárások módosítása, szoftver alkalmazások futási sebességének növelése, más információ-technológia részletkérdések megoldása nem tartoztak szorosan kutató munkám feladatai közé, azokat a témakörrel foglalkozó társaim önálló kutatómunka keretében vizsgálták és vizsgálják jelenleg is. (pl. párhuzamos algoritmusok, GPGPU alapú programozás, in-memory computing stb.)

A tézisek érvényességét – a mesterségesen létrehozott adatokon végzett futtatási tesztek kedvező tapasztalatait követően – valós problémák ipari környezetben történő megoldásával igazoltam. Ehhez konkrét termelési rendszerek műszaki specifikációi és ipari adatbázis adatok kerültek felhasználásra, ipari partnerek bevonásával és engedélyével.

A kapcsolódó K+F+I projektek tapasztalatait is felhasználva, a tudományos eredményeket konferenciákon mutattam be és folyóiratokban is publikáltam.

2 A kutatáshoz kapcsolódó alapvető termelésütemezési modellek áttekintése

2.1 Alapvető jellemzők

Általános értelemben az ütemezés elvégzendő munkák időbeli kiosztását (hozzárendelését és időzítését) jelenti rendelkezésre álló erőforrások között. A munkák és az erőforrások típusai nagyon sokfélék lehetnek és együttesen erősen függnak az aktuálisan vizsgált szituációtól. Az ütemezési feladatok kombinatorikus optimalizálási problémákhoz vezetnek, amelyekben az előzetesen meghatározott célfüggvény vagy célfüggvények szélsőértékét (minimumát vagy maximumát) eredményező megoldás megkeresésére törekszünk az előírt korlátfeltételek betartása mellett.

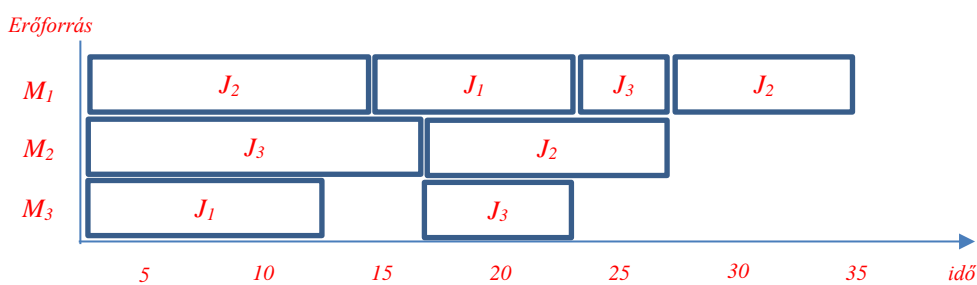
Az ütemezési problémákban a „munka” (job) rendszerint elvégzendő műveletek (operációk) halmaza. A termelésütemezési feladatokban a munka adott számú munkadarab együttesét is jelenti. Az adott munkadarab-halmaz egyedein előre definiált műveleteket kell elvégezni a technológiai tervben megadott módon. Az erőforrások képesek elvégezni a szükséges műveleteket a munkadarabokon, ezáltal végrehajtják a munkákhoz tartozó operációkat. Az ütemezési feladatok esetében a valós erőforrások modellje absztrakt virtuális entitásokkal jellemezhető. Ezek az entitások jelentik az elsődleges, másodlagos és egyéb kiegészítő erőforrásokat. (Ilyenek például a szerszámgépek, technológiai berendezések, robotok, szakmunkások, operátorok, dolgozók, munkahelyek, szerszámok, készülékek, vezérlő programok stb.).

Egy ütemezési feladat végső megoldása az „ütemterv”, amely a munkák operációinak a kiválasztott erőforrásokon (erőforrásokkal) történő időbeli végrehajtását definiálja. Az ütemtervet (megoldást) leggyakrabban Gantt diagramok jelenítik meg grafikus formában. A grafikus elemek sokfélék lehetnek (vonalak, téglalapok, különböző feliratozási stílusok, színek, kitöltési effektusok stb.). Elrendezését tekintve két alaptípust szokás megkülönböztetni:

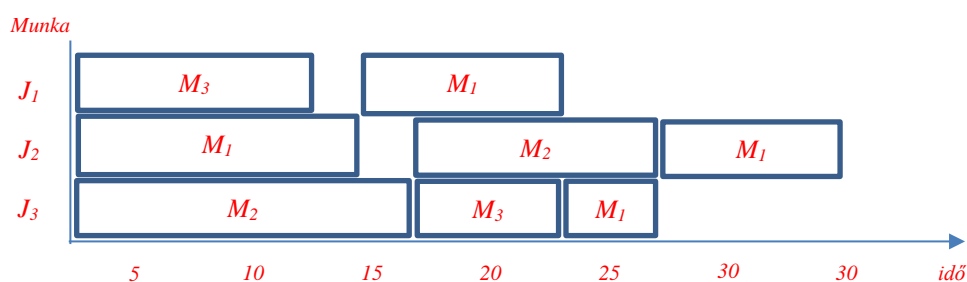
- Az erőforrások állapotára vonatkozó diagram (Resource-oriented chart) függőleges tengelyén az erőforrások, vízszintes tengelyén az idő van feltüntetve. A diagram téglalapokból áll, egy téglalap az adott erőforráson egy adott munka adott operációjának végrehajtását jelenti. A rendezőelvet az erőforrásokon egymás után következő operációk grafikus ábrázolása jelenti.
- A munkákra vonatkozó diagram (Job-oriented chart) függőleges tengelyén a munkák szimbólumai, vízszintes tengelyén az idő van feltüntetve. A diagram téglalapokból áll, egy téglalap az adott munka egy adott operációjának adott erőforráson történő végrehajtását jelenti. Az egyes munkák egymás után következő operációinak áttekinthető ábrázolása jelenti a rendezőelvet.

Az 1. ábra példaként egy ütemtervnek két különböző aspektusát jeleníti meg. A téglalapok hossza (műveleti idő) változatlan, a kezdési és befejezési időpontok rendre ugyanazok csak az elrendezésben és a feliratokban van eltérés.

Erőforrás-orientált Gantt diagram



Munka-orientált Gantt diagram



1. ábra: Ütemterv reprezentálása Gantt diagrammal.

A gyártásirányítás alapvető feladata a felettes termelésstervezési szinttől kapott középtávú termelési tervek rövidtávú feladatokra bontása, a feladatok ütemezése. Az ütemezés során meg kell vizsgálni a szükséges technológiai tervek és erőforrások rendelkezésre állását, fel kell térképezni a korlátozásokat és lehetőségeket. Ezt követően megtörténhet az operációk és erőforrások egymáshoz rendelése, továbbá az egyes munkák indítási sorrendjének és időpontjainak meghatározása [23], [22].

Az ütemezéssel kapcsolatos döntések meghozatala alapvetően prediktív és reaktív módon történhet. A gyakorlatban jelenleg inkább elterjedt prediktív ütemezés meghatározott időközönként, adott időszakra előre készít aktuális gyártási ütemtervet [45], [44], [38]. Egyre nagyobb azonban az igény a bizonytalanságok és a váratlan események gyors kezelésére. Ezt a célt szolgálja a reaktív ütemezés, melynek során a rendszerben bekövetkező események hatására valamilyen döntési logika alapján kerül kiválasztásra a szituációnak leginkább megfelelő beavatkozási tevékenység [47], [48], [4], [37], [13], [20], [31].

A diszkrét gyártási folyamatok sokfélék, technológiáik és korlátozásaik nagyon eltérők lehetnek [3], [6], [9], ezért eredményes irányításuk ismétlődő gondos modellezést igényel. Az irányítási funkció hatékony megvalósításának egyik fontos feltétele a felmerülő ütemezési feladatok minél pontosabb megfogalmazása és megoldása.

Az ütemezési feladatok osztályozására a szakirodalomban többféle javaslat található. Az egyik legelterjedtebb szempontrendszer az $\alpha|\beta|\chi$ formalizmus, amelyet Graham és társai javasoltak először 1979-ben [18]. Azóta számos szimbólummal bővült a formális leírás [10], azonban az alap elképzelés változatlan maradt. Ezt felhasználva és továbbfejlesztve egy valós termelésprogramozási feladat megfogalmazása során három alapvető körülményt kell megvizsgálni és a kapcsolódó kérdéseket pontosan megválaszolni. Ezek a következők:

1. Az erőforrás-környezet definíciója. Ez magában foglalja az ütemezési feladatban szereplő erőforrások (gépek, munkahelyek, dolgozók stb.) jellemző tulajdonságait és a közöttük lévő kapcsolatrendszerrel, különös tekintettel a munkákhoz tartozó operációk (műveletek) végrehajtásának jellemzőire.
2. A munkák definíciói. A munkák tulajdonságain kívül ez tartalmazza a munkákra vonatkozó előírásokat, korlátozásokat, végrehajtási jellemzőket és megengedett alternatívákat.
3. Az aktuális „termelési politika” definíciója. Ez prioritások, szabályok, célfüggvények, menedzser indexek rendszere, amely az ütemezés célját (goal), irányítási környezetét írja elő. Rendszerint számszerűsített teljesítménymutatók formájában.

2.2 Ütemezési feladatok változatossága az erőforrások és operációk szempontjából

A témakör elméletével foglalkozó kutatók többsége főként operációkutatási szempontok szerint vizsgálja az ütemezési feladatokat [10], [38], [39]. Az erőforrás-környezet szempontjából legegyszerűbb az egygépes (Single Machine) ütemezési modell típus, amelyben minden munka esetében egyetlen operációt kell végrehajtani, és csupán egy gép (erőforrás) terhelhető [27].

A párhuzamos gépes modellek (Parallel Machines) alapvető jellemzője, hogy minden munka esetében csupán egyetlen operációt kell végrehajtani, de több gép dolgozhat párhuzamosan különböző munkákon. A gépek képességeit figyelembe véve három további kategóriát különböztethetünk meg [10], [38], [12]:

- teljes mértékben egyenértékű gépek csoportja (Identical Parallel Machines);
- egymástól különböző állandó intenzitás-értékkel működő gépek csoportja (Uniform Parallel Machines);
- munkától függő intenzitás-értékekkel működő gépek csoportja (Unrelated Parallel Machines).

A szakirodalomban sok cikk ad részletes áttekintést a párhuzamosan működő gépek ütemezési modelljeiről, azonban Weng és társai [50] valamint Lamothe és társai [26] felhívták a figyelmet arra, hogy a korlátozott gyártási képességeket, átállási időket és másodlagos erőforrás-korlátokat együttesen kezelő modellek száma sokkal kisebb.

A műhelymodellek esetén több munka, több operáció és több gép tartozik a rendszerhez. Minden munka több operációból állhat. Minden egyes operációt egy adott gép végezhet el. Az operációk száma lehet munkánként eltérő vagy azonos (General Shop). Különböző előírás vonatkozhat a munkák operációinak számára és sorrendjére [10]:

- minden munka esetében azonos (Flow Shop),
- kötött, de munkánként eltérő lehet (Job Shop),
- nem korlátozott (Open Shop),
- a felsorolt alaptípusok kombináltan egyszerre is előfordulhatnak (Mixed Shop).

A rugalmas műhelymodellekben (Flexible Flow Shop, Flexible Job Shop stb.) a rugalmas jelző (flexible) az ütemezési modellekhez kapcsolva arra utal, hogy míg az alap műhelymodellekben egy adott operáció végrehajtására egyetlen gép alkalmas csupán, addig a

rugalmas modellben az adott operációt a gépek egy meghatározott csoportjába (halmazába) tartozó bármelyik gép elvégezheti. Ezáltal az alap műhelymodell kibővül operációhoz kapcsolódó gépválasztási feladattal, ugyanakkor továbbra is alapvető szerepet játszik a munkák gépenkénti sorrendjének és indítási időpontjának meghatározása. A gépcsoport gépei egymással párhuzamosan működő egyenértékű, részben azonos vagy különböző intenzitás- és más paraméterértékekkel rendelkező gépek lehetnek. Jellegzetes példái a rugalmas műhelymodelleknek a rugalmas egyutas (Flexible Flow Shop, FFS) és a rugalmas többutas (Flexible Job Shop, FJS) modell típusok [2], [32], [40], [49], [15], [11].

A kiterjesztett rugalmas modellek (Extended Flexible Flow Shop, Extended Flexible Job Shop stb.) esetén vannak a rendszerben olyan erőforrások (gépsorok, munkahelyek), amelyek összevontan több operációt tudnak végrehajtani. Az operáció helyett nagyobb egységek (technológiai lépések, végrehajtási lépések) tekinthetők az ütemezés alapegységeinek [23], [24], [s11].

A vázolt modell típusok további speciális képességekkel bővíthetők. Ezek közé tartoznak a többcélú gépeket is integráló modell típusok (Multi-Purpose Machine, MPM), melyekben egy adott gép többféle operáció elvégzésére alkalmas (több halmazhoz tartozik). A rugalmas gyártórendszerekben gyakran egy adott gép több különböző szerszámot, készüléket, programot stb. használhat, így átállítást követően többféle operáció elvégzésére is alkalmas különböző időkben (átlapolódás nélkül).

Az ütemezési modellekkel szemben támasztott követelmények között szerepel az igényelt erőforrásokra vonatkozó aktuális korlátozások minél pontosabb definiálása. Ebbe a kategóriába tartoznak például az erőforrások (pl. gépek, személyek, eszközök) időbeli rendelkezésre állására vonatkozó előírások [2], az erőforrások (pl. megmunkáló gépek, műveletközi tárolók) kapacitáskorlátai, az erőforrások (pl. szerszámok, készülékek) alkalmazhatóságára vonatkozó korlátozások.

A korlátozottan rendelkezésre álló párhuzamos gépek ütemezési modelljeivel és módszereivel viszonylag sok cikk foglalkozik. Ezekről részletes összefoglalást adnak például a [17] és [19] publikációk. Ehhez a témakörhöz szorosan kapcsolódik az értekezésben szereplő modellek és módszerek többsége. Az értekezés erőforrás-környezet szempontjából szorosan kapcsolódik a rugalmas gépcsoportokból (stage) összekapcsolt gyártórendszerek ütemezési problémáihoz is, amikor a gépek rendelkezésre állása nem folyamatos. Ilyen feladatokról ad jó áttekintést például a [2] és [35] cikk.

2.3 Ütemezési feladatok változatossága a munkák szempontjából

A legtöbb esetben a munkák operációi nem szakíthatók meg, azonban vannak olyan szituációk, amikor megengedett az operációk végrehajtásának felfüggesztése majd egy későbbi időpontban való újraindítása. Ez utóbbi esetben az ütemező döntési hatásköre jelentősen kibővül. Hasonló változást eredményez az is, ha megengedett a munkák kisebb egységekre bontása és/vagy nagyobb egységekbe való összevonása. Újabb elvárásokat generál, ezáltal további modellbővítést indokol a munkák végrehajtásához kapcsolódó minőségi jellemzők figyelembevételének igénye is (pl. közbenső minőségellenőrzés után a művelet ismételt végrehajtása szükséges). További jelentős kiterjesztést igényel a változtatható intenzitás-

paraméterek alkalmazhatóságának és a helyettesítő komponensek rendelkezésre állásának figyelembe vétele.

A modellekkel szemben támasztott követelmények között szerepel a munkákra vonatkozó aktuális korlátozások minél pontosabb definiálása. Ebbe a kategóriába tartoznak például:

- az indításra és/vagy befejezésre vonatkozó időbeli előírások [27],
- a munkák között fennálló megelőzési relációk [14],
- a munkák gépekhez rendelésére vonatkozó korlátozások [33], [29],
- a munkák operációinak megkezdését lehetővé tevő átállítási/beállítási időtartamok [1]
- a logisztikai feladatok kezelése [28], stb.

Komoly elvárást támaszt az ütemezési modellekkel szemben a rendszerben jelenlévő párhuzamos végrehajtású munka (Multiprocessor Task, MPT): Ilyen esetben egy párhuzamos végrehajtású munka rendelkezik legalább egy olyan operációval, amelynek végrehajtásához egyszerre egynél több erőforrásra (gépre, dolgozóra, szerszámra stb.) van szükség a teljes műveleti idő alatt (egyidejűleg több erőforrást foglal).

2.4 Ütemezési feladatok változatossága az ütemezési célok szempontjából

A termelés lokális (munkahelyi), rendszer (műhely) szintű és globális (vállalati) céljai (pl. stabilitás, kihasználtság, készletek, átfutási idők, csúszások, minőség, piaci pozíció, profit, hatékonyság javítása stb.) igen eltérőek, esetenként ellentmondóak is lehetnek. A gyorsan változó piaci környezet és az arra történő gyors reagálás az egyes szinteken felállított célok felülvizsgálatát és megváltoztatását is eredményezheti. Fontos elvárás tehát, hogy a modern ütemezési modellek – a termelési politika változó céljainak megfelelően – többkritériumos szemléletűek legyenek és támogassák az időben változó célok elérését.

A szakirodalomban publikált ütemezési modellek többsége csupán egyetlen elsődleges célfüggvény optimalizálása koncentrált. Az igény szerinti rugalmas gyártás szempontjából kiemelkedően fontos többcélú optimalizálási szemlélet sokkal kevesebb ütemezési modellben jelenik meg [34], [43].

Az ütemezési feladatban alkalmazható optimalizálási célfüggvények igen sokfélék lehetnek. Ezek a célok bizonyos esetekben a teljesítménymutatók (pl. költségek, késések, csúszások, befejezési időpontok, átfutási idők, várakozási idők, gépátállítások, készletszintek stb.) minimalizálását követelik meg. Más célok esetén a teljesítménymutatók (pl. gépi és emberi erőforrások kihasználása, áteresztőképesség, gyártási hatékonyság stb.) maximalizálása a követelmény. A célfüggvények gyakran egymástól sem függetlenek, kapcsolatrendszerük bonyolult, komplex feladatok esetében nehezen modellezhető [8]. Az ilyen feladatoknak csak kivételes esetekben van olyan megoldása, amely az összes célfüggvény szempontjából egyszerre tekinthető optimálisnak. A korszerű ütemezési modelleknek lehetővé kell tenniük a változó célfüggvény-rendszerek definiálását. Ez a kiindulási alapja annak, hogy hatékony és rugalmas megoldási módszerek alkalmazásával kompromisszumos kvázi-optimális végrehajtható megoldás (vagy megoldás halmaz) elfogadható idő alatt előállítható legyen.

A gyakorlat szempontjából fontos elvárás, hogy nemcsak optimum-közeli, hanem robusztus ütemtervet kell előállítani több, prioritással ellátott cél esetén. Robusztus ütemterveknek nevezhetjük az ütemtervek olyan csoportját, amely a reaktív (előre nem vagy csak nagy bizonytalansággal tervezhető eseményekre adandó válasz) tevékenységeket a lehető legjobban támogatja (pl. a KPI mutatók értéke valamilyen időátlag definíció szerint optimális).

Az ütemezési feladatok és a szükséges modellek változatossága megnehezíti olyan szoftverek kidolgozását, amelyek könnyen és gyorsan adaptálhatók adott termelési rendszer gyártásirányításának támogatására. A korszerű gyártásirányítást támogató szoftverek (Manufacturing Execution System, MES) ISA-95 alapú modellje a következő négy funkcionális fő komponenszt definiálja [42]: (1) termelésirányítás, (2) üzemfenntartás, (3) minőségbiztosítás, (4) készlet-gazdálkodás. A termelésirányítás (termelés-menedzsment) szolgáltatásai között kiemelt szerepet kap az ütemezés és az újraütemezés [16]. Ezek támogatása érdekében erőteljes nemzetközi kutató munka folyik világszerte. Az elméleti eredmények gyakorlati bevezetése azonban további jelentős implementációs szakértői munkát igényel.

3 Diszkrét alkatrészgyártási és szerelési folyamatok termelésütemezési és műszakbeosztási feladatainak integrált megoldása

3.1 A vizsgált feladat ismertetése

3.1.1 Szerelést közvetlenül megelőző alkatrészgyártási folyamat ütemezése

A téma kutatásának motivációját az Electrolux Lehel Kft. jászberényi hűtőszekrény és fagyasztóláda gyárában felmerült feladatok adták. Különböző egymást követő gyártási fázisok között nagyon hasonló gyártásszinkronizálási kihívás mutatkozott. Ilyen például a hűtőszekrény-test vákuumszívása és előszerelése, a hűtőszekrény-ajtó habosítása és szerelése, valamint a fagyasztóláda-belsőkad gyártása és szerelése. Az általánosított feladatra (a konkrét folyamatok és berendezések megnevezése és darabszáma nélkül) az alábbiak jellemzők:

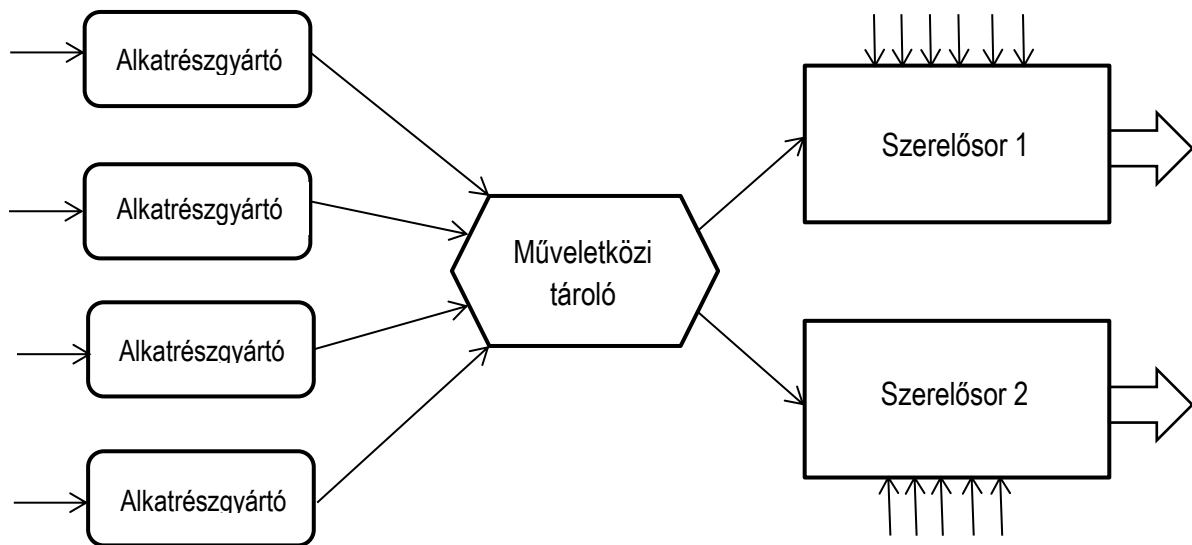
Adott egy diszkrét termelési rendszer, melyben adott számú (kettő vagy több) párhuzamosan működő gyártósoron folyik a végtermékek összeszerelése. A kiindulási és közbülső termékek alapvetően három nagy anyagcsoportból állnak össze. Az első ilyen anyagcsoportba tartoznak az alapanyagok, az alkatrészek és a részegységek. A másik két anyagcsoportba a segédanyagok és az egyéb anyagok tartoznak. A végtermék alkatrészeinek (komponenseinek) egy meghatározott részét helyben gyártják, a többi típust külső beszállító partnerektől szerzik be. A továbbiakban egy kiválasztott alkatrésztípus gyártásának ütemezési feladatát részletezem. A feladat a többi – helyben gyártott – alkatrésztípusra is megfogalmazható a paraméterek aktualizálásával (pl. a gépek darabszáma, időadatok stb.). A kiszállítási tervek alapján a végszerelés aktuális termelési terve ismert. Ezt az integrált vállalatirányítási (Enterprise Resources Planning, ERP) rendszer MRP (Manufacturing Resources Planning) modulja készíti el. A gyártásirányítás szempontból kiemelt fontosságú a szerelést közvetlenül megelőző alkatrészgyártás megfelelő ütemezése.

3.1.2 A termelési folyamat jellemzői

A modelltípushoz tartozó konkrét gyártórendszerben két technológiai szerelősor áll rendelkezésre. A szerelősorokon szerelik össze a végtermékeket ($p=1, 2, \dots, N_p$), melyeknek egyik fődarabját előzetesen négy párhuzamosan működő alkatrészgyártó gép valamelyikén munkálják meg. A szerelősorok és az alkatrészgyártó gépek között egy közös műveletközi tároló van kialakítva. A technológiai lépések végrehajtási sorrendje természetesen kötött (alkatrészgyártás \rightarrow szerelés). Ez a két technológiai lépés adja a gépek csoportosításának rendező elvét. A végtermék gyártási folyamat vizsgált fázisainak elvi vázlatát a 2. ábra szemlélteti.

A rendszerben P_p ($p=1, 2, \dots, N_p$) végtermék gyártható. Minden egyes P_p termékbe pontosan egy darab, helyben gyártott alkatrész épül be. Az alkatrészek E_e ($e=1, 2, \dots, N_E$) halmaza ismert.

Az alkatrészek és a termékek technológiai specifikációi egyértelműen meghatározzák az adott műveletek végrehajtására használható (megfelelő képességekkel rendelkező) technológiai berendezések halmazait, és ezek engedélyezett kombinációjaként adódó technológiai útvonalakat. Az ütemezési feladatban ezeket az előírásokat szigorúan figyelembe kell venni. Ismernek tekinthető a rendszer anyagáramlási gráfja (nem minden alkatrész gyártható minden gépen, és nem minden termék szerelhető minden szerelősoron). Az alkatrészgyártó gépekről egy adott munkához tartozó sorozat már elkészült elemei kisebb egységekben (akár darabonként is) továbbíthatók. Ezáltal ugyanannak a sorozatnak különböző munkadarabjai különböző helyeken (gépeken vagy tárolóban) lehetnek. Az átfutási idők pontosabb számítása érdekében az anyagmozgatási időket is be kell vonni a modellezésbe.



2. ábra: A feladatban szereplő gyártó-szerelő rendszer elvi vázlata.

A szerelősorok működését a késztermékekre vonatkozó szerelési ütemterv határozza meg, amely a vevői igények alapján készül. A finomprogramozási funkció feladatai közé tartozik a vázolt diszkrét alkatrészgyártási és szerelési fázisok működésének összehangolása (szinkronizálása) azáltal, hogy az alkatrészgyártás mindenkor termelési finomprogramját úgy kell megtervezni, hogy az adott aktuális szerelési ütemterv fennakadások nélkül megvalósítható legyen.

3.1.3 A termelési folyamat technológiai és logisztikai objektumai

Az ütemezési feladat szempontjából a termelési folyamat berendezései közül a következő objektumok a főszereplők:

Alkatrészgyártó gépek csoportja MG_1 (1. gépcsoport):

- M_1 (1. gép)
- M_2 (2. gép)
- M_3 (3. gép)
- M_4 (4. gép)

Szerelősorok csoportja MG_2 (2. gépcsoport):

- M_5 (5. gép)
- M_6 (6. gép)

Műveletközi alkatrésztároló B_1 (1. buffer):

- a két gépcsoport között kialakított, közösen használt műveletközi alkatrésztároló.

Az ütemezési feladat megoldása során a megengedett technológiai útvonalakat a mindenkori alkatrész és gép páros aktuálisan engedélyezett összerendelési lehetőségeiből kiindulva kell meghatározni minden egyes rendelési sorozatra, figyelembe véve az előzetesen célállomásként megjelölt szerelősort. Ez egy fontos követelmény, amit az ütemező rendszernek teljesíteni kell.

Feltételezhető, hogy a különböző gépek (munkahelyek) a különböző alkatrészeken vagy termékeken a műveletvégzést eltérő sebességgel, különböző idők alatt valósíthatják meg. Bizonyos alkatrésztípusok több gépen történő egyidejű gyártására is lehetőség van (pl. különböző rendelések ugyanarra a terméktípusra vonatkoznak), más típusoknál ugyanez esetleg nem áll fenn.

A gépeket különböző alkatrész vagy termék gyártása között át kell állítani, ennek bizonyos időigénye van. Az átállítás alatt a gép nem végezhet műveletet. Az átállítási idők nagysága függ a konkrét géptől és az egymást váltó terméktípusok sorrendjétől. Általános esetben az átállítási idők nem szimmetrikusak, vagyis két termék gyártása közötti átállítás időtartama ellentétes sorrendet feltételezve nem feltétlenül azonos. Léteznek olyan terméktípusok is melyek között nem szükséges átállítani a gépeket. A terméktípusok tehát egyértelműen besorolhatók diszjunkt halmazokba (osztályokba) és definiálhatók a halmazok közötti átállítási idők, míg a halmazon belüli átállítási idők rendre nullának tekinthetők.

Előidejű (prediktív) tervezésnél az aktuális ütemezési időszakra (pl. egy hétre) vonatkozóan a szerelősorok tervezett rendelkezésre állási időintervallumainak pontos listája (pl. 8 órás műszakbeosztása) gépenkénti bontásban rendelkezésre áll. Ezek az adatok az aktuális megrendelések részletes adataival együtt (terméktípus, darabszám, szerelősor-azonosító, szerelési sorozat indítási időpontja) az ERP rendszer által generált szerelősori termelési terv részét képezik. Az alkatrészgyártó gépek használata szintén műszakbeosztáshoz kötött (figyelembe véve a működtető személyzetet), de ezek kijelölése már az ütemező döntési hatáskörébe tartozik. Ezek alapján, a műszakbeosztás egyrészt korlátozási feltétel (a szerelősorok esetében), másrészt döntési változó (az alkatrészgyártó gépek esetében).

Az ütemezési feladatban fontos szerepet játszik a műveletközi tároló. Az ütemezési modell kidolgozásakor a következő jellemzőket is figyelembe kell venni:

- a szerelősorokon folyamatos munkavégzést kell biztosítani az ERP rendszer által előírt termelési finomprogramnak megfelelően,
- a műveletközi készletek nagyságát befolyásolja:
 - a választott technológiai útvonalak mentén egymás után következő gépek műveleti és átállítási időinek nagysága és ezek aránya,
 - az indított rendelési sorozatnagyságok,
 - a sorozatok indításának ütemezett sorrendje,
 - a gépek közötti anyagmozgatás során alkalmazott egységgrakományban elhelyezhető munkadarabok száma (logisztikai sorozatnagyság),

- a gépekről a tárolókba valamint onnan a gépekhez való szállítások időadatai,
- a gépek műszakbeosztásai,
- a műveletközi tároló kapacitása terméktípusonként eltérő (típusonként változnak a befoglaló méretek),
- több gép egyszerre tölti és üríti különböző intenzitással és különböző típusú alkatrészekkel a tárolót,
- ha a tároló megtelik, akkor blokkolja az előtte lévő gépeket, ezért azokat le kell állítani, majd később újra kell azokat indítani (az újraindításnak jelentős időszükséglete lehet).

3.1.4 Termelési politika, célok, elvárások és kritériumok

A vázolt erőforrás-korlátos ütemezési feladat megoldása során olyan megoldások megtalálása a cél, melyek a szigorú korlátozásokat nem sértik meg és a termelés-menedzsment által a termelési politika keretében megfogalmazott kritériumok tekintetében a „lehető legjobb” számszerűsített teljesítménymutatókat (performance indicators) eredményezik. A teljesítménymutatók nagyon sokfélék lehetnek, és időnként megváltozhatnak követve a termelési rendszer változó irányítási céljait. Több szempont együttes figyelembe vétele változtatható kritériumrendszer mellett (multi-objective optimization) nehéz feladat.

A szigorú technológiai, anyagmozgatási, tárolási és rendelkezésre állási korlátozások nagy számára és erősségére tekintettel a túlterhelt időszakokban megengedett a határidők átlépése (puha korlátok). Ilyenkor az ütemezési modellben a zéró csúszásokra vonatkozó korlátozások helyett minimalizálandó büntető-függvények jelennek meg a célfüggvények között.

A bemutatott ütemezési feladat megoldásával kapcsolatban további termelési elvárások is megjelenhetnek. Ilyenek például:

- A szerelősorokra előírt termelési tervek csúszások nélküli megvalósulása a legnagyobb prioritású. Ennek érdekében minden szükséges alkatrésznek és részegységnek időben meg kell érkezni a szerelősorokra.
- A gépek átállításának jelentős idő és költség vonzata van, ezért az alkatrészgyártás a lehető legkevesebb átállással valósuljon meg.
- A gépek leállítása és újraindítása időigényes és hibák (selejt) forrása, ezért törekedni kell a műszakon belüli folyamatos munkavégzésre. A műveletközi tároló telítődése következtében kialakuló nemkívánatos várakozásokat a lehető legkisebb értéken kell tartani, ezért fontos a gyártás stabilitásának megőrzése minél alacsonyabb készletszint fenntartása mellett.
- Az alkatrészgyártásban közreműködő dolgozók műszakbeosztását a gépek terhelésével összehangolva úgy kell megtervezni, hogy a behívott (beosztott) dolgozók folyamatosan tudjanak dolgozni (költségcsökkentés).

3.2 Egy új kiterjesztett modell az ütemezési és műszakbeosztási feladat integrált kezelésére

A szakirodalomban fellelhető nagyon sok modell ellenére a fent megfogalmazott és vizsgált probléma sajátosságai miatt nem sorolható be közvetlenül egyik ismert ütemezési feladatkategóriába sem. Erőforrás-környezet szempontjából az *Extended Flexible Flow Shop (EFFS)* modellek kategóriájára vezethető vissza. A korábban publikált legjobb modellek jellemzőit felhasználva [23], és kibővítve a vizsgált feladat sajátosságainak megfelelő elemekkel egy továbbfejlesztett feladattípust definiáltam, amely a következő szimbolikus formában írható le:

$$FF2, M_g, Q_{i,m}, Set_{px,py,m}, Cal_m, B_p / R_i^{O2}, D_i, A_{i,g}, Tr_{mx,my} / f_1, f_2, \dots, f_K \quad (1)$$

ahol a szimbólumok jelentése a következő:

- $FF2$ – Az operációk (műveletek) száma kettő és sorrendjük kötött. A munkadarabok által felkeresett gépek sorozata (technológia útvonal) eltérő lehet (Flexible Flow Shop);
- M_g – Funkciók szerint rendezett gépek halmazai (gépcsoportok), amelyekhez egy vagy több párhuzamosan működő gép vagy gépsor tartozik;
- $Q_{i,m}$ – A gépek/gépsorok munkáktól függő termelési sebességekkel működhetnek;
- $Set_{px,py,m}$ – Munkadarab-osztályok sorrendjétől és géptől függő átállítási időadatok;
- Cal_m – A gépekhez tartozó műszakok (rendelkezésre állási időintervallumok);
- B_p – Közös használatú, korlátozott méretű műveletközi alkatrésztároló, melynek tárolási kapacitása függ a tárolandó alkatrészek típusától;
- R_i^{O2} – A munkák legkorábbi indítási időpontjai (indításra vonatkozó időbeli korlátozások). A szerelési folyamat esetében a termelési terv definiálja;
- D_i – A munkák legkésőbbi befejezési időpontjai (teljesítési határidők);
- $A_{i,g}$ – Munkánként definiálható a műveletvégzésre alkalmas gépek halmaza gépcsoportonkénti bontásban;
- $Tr_{mx,my}$ – Gépek közötti anyagmozgatási idők;
- f_1, f_2, \dots, f_K – A kijelölt minimalizálandó célfüggvények listája (egyszerre több célfüggvény is előírható, K a célfüggvények aktuális száma).

Az ütemezési modellben a gyártásirányítás által támasztott igényeket a következő célfüggvények együttes kompromisszumos optimalizálása fejezi ki:

- f_1 : minimális legyen a határidőt túllépő sorozatok száma,
- f_2 : minimális legyen a csúszások időösszege,
- f_3 : minimális legyen a legnagyobb határidő-túllépés (csúszás),
- f_4 : minimális legyen a gépátállítások száma,
- f_5 : minimális legyen a gépátállítások időösszege,
- f_6 : minimális legyen a gépek átlagos kihasználatlansága, (maximális legyen a gépek átlagos kihasználtsága, mivel: $kihasználatlanság = 100 - kihasználtság [\%]$).
- f_7 : minimális legyen a sorozatok átlagos átfutási ideje,
- f_8 : minimális legyen a műszakbeosztások költségeinek összege.

Az ütemezési modellben a műszakbeosztások költségeinek összege helyett, egy új mutatószámot a *súlyozott műszakszámot* használom célfüggvényként. Az új teljesítménymutató számításának lényege a következő:

- A műszakokat kategóriákba (osztályokba) sorolom.
- Minden osztályhoz hozzárendelek egy relatív időegyenérték mutatót. A munkanap délelőtt értéke egy egész (1,00). Ez a referenciaérték, és ehhez viszonyítva adható meg a további kategóriák értéke (az 1. táblázat tartalmaz egy illusztratív példát a kategóriákra és a konkrét értékekre).
- Az adott ütemtervre vonatkozó *súlyozott műszakszámot* az egyes betervezett műszakok kategória szerinti relatív időegyenértékének összegzésével állítom elő.

1. táblázat: Illusztratív példa a műszakok kategóriáira és a relatív időegyenértékekre.

<i>Műszak kategória</i>	<i>Relatív időegyenérték</i>
Munkanap délelőtt	1,00
Munkanap délután	1,25
Munkanap éjszaka	1,50
Nem munkanap délelőtt	1,75
Nem munkanap délután	2,00
Nem munkanap éjszaka	2,50

A tényleges műszakköltség a súlyozott műszakszám és a műhely gazdasági modelljéből kiolvasható fajlagos költségmutató szorzataként adódik. Így az f_8 célfüggvény megfogalmazható úgy, hogy minimális legyen a súlyozott műszakszám.

Az ütemezési modell lényeges új elemei közé tartozik:

- a műszakok menedzselése (korlátozó feltételek és döntési változók),
- a műveletközi tároló irányítása,
- a munkadarab-sorozatok logisztikai szemléletű bontása kisebb (csoportokra),
- a célfüggvények körének jelentős bővítése és rugalmassá tétele.

Mindezek együttes (integrált) kezelése jelentősen segíti az ipari gyakorlati igények kielégítését.

3.3 Megoldási módszer

3.3.1 Megoldási koncepció

A feladat kombinatorikus tulajdonságai miatt a megoldás koncepcióját heurisztikus felépítő algoritmusok és keresési meta-heurisztikák kombinált alkalmazására alapoztam. Az elméleti globális optimum keresése helyett, olyan megoldási módszereket fejlesztettem, amelyek nagyméretű feladatok esetében is elfogadható időn belül több kritérium együttes figyelembevételével kompromisszumosan jó ütemtervet képesek előállítani.

Az alkatrészgyártás finomprogramozási feladatának megoldása során a szerelősorokra előírt termelési terv belső rendelkezéseinek teljesítését kell biztosítani. Ehhez a szükséges munkák

elvégzésére alkalmas gépek lekötését (terhelését) és az elvégzendő műveletek indítási időpontjait kell úgy megtervezni, hogy a szigorú korlátozások megsértése nélkül a megfogalmazott célok elérhetőek legyenek.

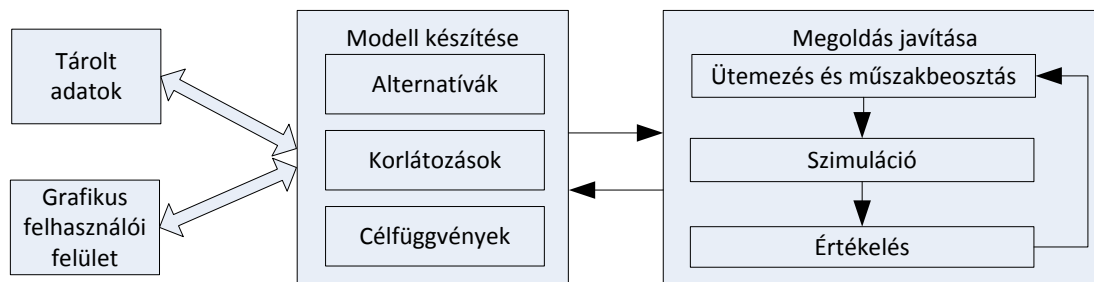
A feladatban a bemenő adatok definiálják az aktuális gyártási erőforrás-környezetet, a belső rendeléseket, a korlátozásokat és a célfüggvényeket. Ezekből kiindulva kell elkészíteni az alkatrészgyártás részletes termelési finomprogramját. A termelési finomprogram tartalmazza, hogy:

- melyik gépen,
- mikor kell az átállítást/felkészítést elkezdni,
- mennyi idő alatt kell azt elvégezni.
- milyen alkatrészfajtából,
- mikortól kezdve,
- mennyi idő alatt,
- mennyit kell gyártani.

Az alkatrészgyártásra vonatkozó munkák (job-ok) a szerelősorok késztermékre vonatkozó termelési programjával összhangban a kívánt alkatrésztípusok, az igényelt darabszámok, az előírt célállomások (szerelősorok) és a megkövetelt határidők együttesét definiálják. Egy alkatrésztípusra vonatkozó munka határidejét az érintett sorozat szerelősoron való indításának tervezett időpontja adja meg. Mivel a gépek szakaszosan dolgoznak, egy adott munka határideje szigorú értelemben a sorozat első munkadarabjára vonatkozik. A sorozat többi munkadarabjához saját belső határidőt rendelnek, melynek pontos értéke a szerelősor adott típusra vonatkozó ciklusidejéből és az aktuális műszakbeosztásából számítható ki. Ennek megvalósítására egy olyan algoritmust készítettem, amely a szerelősorok működését úgy szimulálja, mintha a szükséges összes alkatrész időben rendelkezésre állna. Az alkatrész helyett egy termelési vagy logisztikai csoport is használható, amely az alkatrész egész számú (paraméterben megadott) többszörösét jelenti.

Az ütemezési munka (job) fogalom ebben a modellben adott számú egyforma munkadarabon előre meghatározott műveletek végrehajtását jelenti. Mivel a szerelősorhoz generált belső rendelések nem bonthatók meg, de az azonos típusra vonatkozó rendelések összevonhatók, így az ütemezés alapegysége célszerűen maga a belső rendelés. Minden egyes belső rendelésnek egy önálló munkát (job-ot) feleltet meg. A gyártórendszer és az anyagmozgató rendszer lehetővé teszi, hogy valamely gépen futó munkadarab-sorozat munkadarabjai egyesével vagy kisebb csoportokban átszállításra kerüljenek a következő gépre vagy a műveletközi tárolóba. Ennek következtében adott munkának különböző munkadarabjain különböző műveletek is végrehajthatók egyidejűleg, ezzel jelentősen csökkenthető az átfutási idő. Adott munka (munkadarab-sorozat) adott gépen nem szakítható meg más munkával, de a műveletvégzés szünetelhet azonos sorozat két munkadarabja között, ha a műveletközi tároló megtelt és a következő gyártási fokozat sem tudja a munkadarabokat közvetlenül fogadni (pl. mert blokkolt vagy más munkával terhelt).

A megoldási módszer alapját egy iteratív keresési módszerbe beágyazott szimulációs algoritmus adja (3. ábra).



3. ábra: A megoldási módszer elvi vázlata.

Az első fázis az előkészítés: Az input adatok betöltése. Ezt követően egy részletes vizsgálat következik, amelynek során feltárulnak az összerendelhetőségi szabályok (pl. melyik gép milyen munkadarabokat tud gyártani stb.).

A második fázis a döntési változók értékeinek beállítása: Ütemterv és műszakbeosztás készítése. Egy lehetséges megoldás reprezentációja alapvetően két fő döntési változócsoporthal adható meg. Ezek a következők:

1. Minden egyes munka első műveletének (alkatrészgyártás) alkalmas géphez rendelése, valamint a munkák gépenkénti végrehajtási sorrendjének meghatározása.
2. Az alkatrészgyártó gépek (1. gépcsoport gépei) műszakbeosztásának kialakítása a kialakuló terheléseknek megfelelően. Az engedélyezett műszakok (rögzített méretű időintervallumok) listájának adott géphez rendelése.

A döntési változók az alkatrészgyártásra vonatkoznak. A szerelősorok termelési terve adott az input adatok között. Ezeket közös ütemtervbe kell beilleszteni mivel a termelési folyamat együttesen határozzák meg. Más megközelítésben ez azt jelenti, hogy a teljes ütemterv egy része kötött, nem módosítható. Az ütemezési folyamat eredményeképpen elkészül egy lehetséges termelési ütemterv.

A harmadik fázis a termelési ütemterv alapján a termelési folyamat szimulációja: Az ütemterv végrehajtását jellemző időpontok és időtartamok számítását egy szimulációs eljárás végzi el. Az algoritmus figyelembe veszi a korábban ismertetett korlátozásokat és működési jellemzőket (pl. a gépek műszakbeosztását, a gépeken a munkák adott sorrendje által meghatározott átállítási időket, a gép és a munka összerendelések alapján számítható műveleti időket, a műveletközi tároló készletszintjének alakulását stb.). A szimuláció eredményeit felhasználva az ütemterv kibővül a pontos időadatokkal, ennek következtében elkészül a termelési finomprogram.

A negyedik fázis a generált megoldás minősítése: A szimuláció által számított időadatok és más változók értékeinek felhasználásával egy értékelő komponens kiszámítja a vizsgált ütemtervre (megoldásra) vonatkozó aktuális célfüggvény-értékeket.

A 2.-4. fázis ismételt végrehajtásával hatékony keresési meta-heurisztikára alapozva iteratív módon egyre jobb megoldás-változatok állíthatók elő. „Elegendően jó” célfüggvény-érték kombinációk elérésekor vagy elegendően nagy keresési idő után az iteráció leállítható.

3.3.2 Az alkalmazott szimulációs technikával kapcsolatos követelmények

Az értekezésben vizsgált ütemezési feladatok kiterjesztett modellje a klasszikus feladatokhoz képest speciális szimulációt igényel. A „jó” szimulátor a megoldásesélyes ütemtervekhez tartozó célfüggvény-értékek gyors kiszámításához szükséges.

Adott ütemterv által meghatározott termelési folyamat rész-tevékenységeihez kapcsolódó (kezdő és záró) időpontok és időtartamok értékeinek kiszámítása az ismert EFFT modellel leírható gyártási rendszerek esetében is viszonylag gyorsan elvégezhető [23]. Egy ilyen szabályalapú algoritmus az „összefogó” technológiai lépésekre vonatkozó sorrendet követve bejárja az érintett gépcsoportokat, azokon belül tetszőlegesen sorrendben veszi a párhuzamosan működő gépeket, majd a konkrét gépeken az ütemterv által előírt végrehajtási sorrendben kiszámítja a műveletekhez kapcsolódó időpontokat és időtartamokat (figyelembe véve a korábban már kiszámított időpont értékeket is). A szabályalapú számítási módszer azért nagyon gyors, mert nem kell a különböző gépeken folyó munkák feladatai (munkadarabokon végzett műveletek) között esemény-vezérelt módon „átkapcsolgatni” (process context switch) a szimulációs folyamatot, helyette elegendő a számítás megfelelő sorrendjének megadása, mert ugyanazt a hatást fejt ki.

A hivatkozott szabályalapú módszer azonban az általam vizsgált új modellek esetében a kapacitáskorlátos műveletközi tárolók vagy a munkánként eltérő (job dependent) műveleti sorrendtervek (pl. Job Shop, Open Shop) esetében már nem használható.

A feladat megoldására javasoltam a rendelkezésre álló szimulációs algoritmus [23] továbbfejlesztését. A végrehajtás-vezérelt új algoritmusnak képesnek kell lenni az idő múlásának követésére és ennek megfelelően a gyártórendszerben szereplő összes technológia berendezés és műveletközi tároló együttes kezelésére. A fejlesztés során komoly kihívást jelentett az is, hogy a modellben a munkák a gépek között kisebb csoportokban (logisztikai egységekben) is mozgathatók. Ennek megoldására kétféle sorozatnagyság definiálását javasoltam. Az egyik a munkához (termelési job-hoz) tartozó *ütemezési sorozatnagyság* (teljes alkatrészhalmoz). Ezt az egységet az ütemezéskor egyben használom és alapegységként kerül be az ütemtervbe. A másik a *logisztikai sorozatnagyság*, amely az ütemezési sorozatnagyságtól kisebb is lehet, méretét alapvetően az anyagmozgatási rendszer és az irányítási stratégia határozza meg. Az ütemezési sorozatnagyság a logisztikai sorozatnagyság egész számú többszöröse. A modellben a logisztikai sorozatnagyság bemeneti paraméterként jelenik meg. A szimulációban a számítás és nyomkövetés a logisztikai sorozatnagyságra vonatkozik, majd az ezekkel kapott időadatokból állnak össze a munkákra vonatkozó származtatott értékek.

Ebben a szemléletben minden ütemezési sorozatnagyság egy önálló nyilvántartási egység J_j ($j = 1, 2, \dots, N_j$). Minden egyes J_j egy halmaz, melyhez ugyanolyan típusú alkatrészek tartoznak, ezek ugyanazt az útvonalat követik, és ugyanazokon a gépeken közvetlenül egymás után kerülnek megmunkálásra. Ugyanakkor minden J_j halmaz kisebb méretű diszjunkt U_{jl} ($l=1, 2, \dots, z_j$) részhalmazokból áll. Ezek az U_{jl} alkatrészhalmozok önállóan mozgathatók a gyártórendszerben. A megvalósított szimuláció bemutatásához az U_{jl} kétindexes jelölés helyett az egyszerűbb U_i egyindexes jelölést használom. Az általánosság megszorítása nélkül itt az i index az összes logisztikai egységet figyelembe véve egy globális sorszámot jelöl.

A megvalósított szimuláció feldolgozási egysége a „végrehajtási lépés”. Ez egy adott munka adott logisztikai egységének (a hozzá tartozó munkadarabokkal) adott gépen való

megmunkálását jelenti, ide értve az esetleges előkészítést (set-up) és a kapcsolódó anyagmozgatási (rakodás, szállítás, tárolás) műveleteket is. Ezekkel a követelményekkel a szimuláció legfontosabb lépései a következők:

- A modell-objektumok létrehozása, inicializálása, a kapcsolatrendszer kialakítása.
- Az aktuális végrehajtási lépés kiválasztása.
- A kiválasztott végrehajtási lépés szimulációja, az érintett modell-objektumok frissítése.

A szimuláció legfontosabb modell-objektumai a következők: belső rendelés (production order), munka (job), logisztikai egység (unit), végrehajtási lépés (task), gép (machine), alkatrésztároló (buffer), ütemterv (schedule), termelési finomprogram (fine schedule), célfüggvény-értékeket tároló objektum (KPIs, key performance indicators).

A fentiekben vázolt végrehajtás-vezérelt módszer célja a végrehajtási lépésekhez kapcsolódó időadatok kiszámítása. (Minden egyes gép esetében ismert a hozzá rendelt U_i logisztikai egységek végrehajtási sorrendje.) Fontos megválaszolandó kérdés a következő:

A műveletközi tároló kapacitáskorlátja hogyan befolyásolja a termelési folyamatot?

Ennek megválaszolása érdekében a szimuláció során az összes tevékenységet megfelelő sorrendben kell „lejátszani”. Ezt a sorrendiséget a szimulációban résztvevő modell-objektumok („szereplők”) együttes viselkedése határozza meg.

A megvalósított szimuláció valódi párhuzamos számítást nem tartalmaz, ezért minden végrehajtási szituációban ki kell választani a következő aktív végrehajtási lépést. Az algoritmus minden gépen nyilván tart egy terhelési listát és egy mutatót, amely kijelöli a következő logisztikai egységet. A gépek és a mutatott egységek párokat alkotnak. Egy ilyen páros (M_m , U_i) akkor válik kiválasztásra alkalmas végrehajtási lépéssé, ha az összes indítási feltétele teljesül. Az indítási feltételek a következők:

- A gépet nem blokkolja a tároló (új követelmény).
- A gép befejezte az előző feladatát (klasszikus követelmény).
- A logisztikai egységen végzett korábbi művelet sikeresen befejeződött (új követelmény).

Az algoritmus az események időbeli sorrendiségét úgy biztosítja, hogy a jelöltek közül a legkorábban indítható végrehajtási lépést választja ki, majd kiszámítja a kezdési időpontját (ST_{mi}), az előkészítés (set-up) idejét ($SetT_{mi}$), a műveleti idejét ($ProcT_{mi}$) és a befejezési időpontját (CT_{mi}).

A műveleti időt ($ProcT_{mi}$) a logisztikai egységhez tartozó munkadarabok száma (q_i) és a gép aktuális munkadarab-típushoz tartozó termelési intenzitása (pr_{mi}) együtt határozza meg. Adott U_i egység tényleges indítási időpontját (ST_{mi}) a hozzárendelt M_m gépen a következő értékek befolyásolják:

- a gépen a tároló telítődése által kikényszerített blokkolt időszak vége (BT_{mi}),
- a logisztikai egység legkorábbi indítási időpont-korlátja (r_i),
- a logisztikai egység előző műveletének befejezési időpontja (ct_{pi}),
- a logisztikai egység anyagmozgatási ideje az előző és az aktuális gép között (mt_{ipm}),
- a gép előző műveletének befejezési időpontja (ct_{mh}),
- a gép terméktípus-sorrendtől függő átállítási ideje ($sett_{mhi}$),
- a gépen engedélyezett műszakok listája (CAL_m).

$$\begin{aligned}
a_{mi} &\leftarrow ct_{pi} + mt_{ipm}; \\
SetT_{mi} &\leftarrow sett_{mhi}; \\
ProcT_{mi} &\leftarrow q_i / pr_{mi}; \\
ST_{mi} &\leftarrow \max(a_{mi} - SetT_{mi}, ct_{mh}, r_i - SetT_{mi}, BT_{mi}); \\
CT_{mi} &\leftarrow ST_{mi} + SetT_{mi} + ProcT_{mi}; \\
Load &(ST_{mi}, CT_{mi}, CAL_m);
\end{aligned}$$

4. ábra: A végrehajtási lépések időadatainak egyszerűsített számítása.

Az időadatok számításának egyszerűsített leírását a 4. ábra foglalja össze. A bemutatott algoritmus részlet azt az esetet írja le, amikor megengedett a gép átállításának megkezdése a logisztikai egység tényleges géphez érkezésének (a_{mi}) időpontja előtt (a gép konfigurálásához a munkadarabnak nem kell jelen lennie).

Az időadatok számítását követően a szimbolikusan jelölt $Load$ függvény elhelyezi az U_i logisztikai egység feldolgozásának számított időtartamát az M_m gép idővonalán, figyelembe véve a gép CAL_m engedélyezett műszakbeosztását (rendelkezésre állási időintervallumait). Ezt úgy végzi el, hogy a gép idővonalán balról jobbra haladva megkeresi az $[ST_{mi}, CT_{mi}]$ intervallumot egyben befogadó első szabad időkeretet. Ennek akkor van jelentősége, ha a gép aktuálisan vizsgált műszakjának végéhez közeledve valamely művelet már nem fér bele egyben. Ekkor lép a következő engedélyezett műszakra és addig folytatja a vizsgálatot, amíg be nem tudja illeszteni egy alkalmas időintervallumra megbontás nélkül. Ez a függvény a modellnek azt a tulajdonságát kezeli, hogy a logisztikai egységek feldolgozása a gépeken (végrehajtási lépés) nem szakítható meg.

A szimulációban a végrehajtási lépés lejátszása kiterjed a műveletközi tároló kezelésére is. A számított kezdési időpontban a logisztikai egység munkadarabjait kiveszi a bemeneti forrásból, majd a számított befejezési időpontban beteszi a kimeneti tárolóba. A modellben az alkatrészgyártás esetében a bemeneti tároló virtuális a kimeneti tároló valós, míg szerelésnél éppen fordított. A virtuális tároló mérete nem korlátozott, a valós tárolónak beállított kapacitáskorlátja van. Ha a valós tároló megtelik a munkadarabok az alkatrészgyártó gépeken maradnak és blokkolják a további feldolgozást mindaddig, amíg a tároló készletszintje egy paraméterben megadott szint alá nem süllyed. Ekkor a gépek újra dolgozhatnak. Ha a tároló kiürül, a szerelősor anyagellátása megszűnik, és a szerelés leáll. A szerelősor akkor is leáll, ha a tároló nem üres, de a kívánt alkatrésztípusból nincs annyi készleten amennyi a szerelés folyamatos biztosításához minimálisan szükséges. Ez kialakulhat például akkor is, ha a tároló más alkatrészekkel van tele és a kívánt típus gyártása emiatt blokkolt.

A műveletközi tároló különböző alkatrésztípusra vonatkoztatott kapacitáskorlátja eltérő lehet. Például a befoglaló méretbeli különbségek miatt egyik típusból 1500 darabot, egy másik típusból 1200 darabot tud tárolni akkor, ha egyidejűleg csak egyféle termékkel van teletöltve. Ezért a szimuláció során a tároló kapacitásának százalékos kihasználtságát veszem figyelembe a ténylegesen tárolt darabszámok helyett. Az alkatrésztípusra vonatkozó kapacitás egy darabra vetített százalékos értéke alapján a vegyes tárolás esetében is pontosan meghatározható a tároló aktuális állapota (hány százalékig van megtöltve). A berakás és kiszedés során a mozgatott logisztikai egységek méretét szintén százalékos értékre konvertálva vonja ki, illetve adja hozzá a tároló aktuális százalékos kihasználtságához az algoritmus.

Amint látható az értekezésben vizsgált folyamatban az alkatrésztároló erőteljesen kihat az ütemterv teljesítmény-mutatóira. Ezért nagyon fontos a megfelelő ütemterv keresése során a

vizsgált gyártórendszer minél pontosabb szimulációja, másrésztől viszont azt is figyelembe kell venni, hogy a szimuláció sebessége alapvetően befolyásolja a teljes ütemezési (termelésprogramozási) folyamat keresési algoritmusának futási idejét.

3.3.3 Az integrált termelési finomprogramozás keresési algoritmus

A kiterjesztett modell ütemtervének optimalizálásához szükségessé vált a rendelkezésre álló meta-heurisztikus kereső eljárások továbbfejlesztése. Ehhez a következő követelményeket fogalmaztam meg.

Az ütemezési feladat döntési változóinak értékét egy továbbfejlesztett többoperátoros és többcélú kereső algoritmus (*MOMOTS*) állítja be (5. ábra).

```

MOMOTS
{
  s0 ← Kezdeti megoldás készítése;
  s* ← s0;
  Taboo_List ← NULL;
  while ( Leállási feltétel nem teljesül )
  {
    while ( Szomszédság kiterjesztésének feltétele teljesül )
    {
      Nc ← Az aktuális szomszédsági operátor kiválasztása (Priority_List);
      s ← Szomszédos megoldás készítése ( s0, Nc);
      if ( A Taboo_List nem tartalmazza ( s ) )
      {
        A Taboo_List bővítése új elemmel ( s );
        if ( Taboo elemek száma > megengedett érték )
          A Taboo_List legkorábban felvett elemének törlése;
        if ( A szomszédság kiterjesztésének első eleme ( s ) ) sk ← s;
        else if ( s < sk ) sk ← s;
      }
    }
    s0 ← sk;
    if ( sk < s* ) s* ← sk;
  }
  return s*;
}

```

5. ábra: Többoperátoros, többcélú kereső algoritmus.

Induláskor a kezdeti s_0 ütemterv heurisztikus felépítő szabályok kombinált alkalmazásával készül ($s_0 \leftarrow$ Kezdeti megoldás készítése). Ennek menete a következő:

- A munkákat sorba rendezi a belső határidők szerint nem csökkenő sorrendbe (EDD).
- Ebben a sorrendben haladva, minden munkához az alkalmas gépek halmazából egyenletes valószínűséggel véletlenszerűen választ egy gépet.
- A kiválasztott gépen a már beütemezett munkák mögé helyezi el az aktuális munkát.
- A gépek összes rendelkezésre állási időintervallumát (műszakját) engedélyezi (bekapcsolja).

A keresési folyamat során a kezdeti s_0 ütemtervből kiindulva megengedett módosítások ismételt végrehajtásával készül el a legjobbnak ítélt s^* ütemterv. Az iteratív javítás egy közbenső lépése során, az aktuális kiterjesztés (szomszédság) s_0 bázismegoldásából kiindulva az algoritmus paraméterben definiált számú kiterjesztett (szomszédos) s ütemtervet készít. A szomszédok generálását az aktuálisan kiválasztott szomszédsági N_c operátorok alkalmazásával valósítja meg. A rendelkezésre álló ütemező kereső algoritmusának szomszédsági operátorait az új követelményeknek megfelelően javasoltam kibővíteni.

A szomszédsági operátorok alapvetően a munkákhoz rendelt útvonaltípusra, a konkrét gépekre, a végrehajtási sorrendekre, és a műszakokra vonatkozó döntési változók értékeit módosítják különböző mértékben. A szomszédsági operátorok működési algoritmusai a korábban leírtaknak megfelelően csak az alkatrészgyártó gépekre vonatkozó ütemtervrészeket változtatják meg, a szerelősorokét változatlanul hagyják.

Az implementált új, továbbfejlesztett módosító operátorok a következők:

- N_1 egy véletlenszerűen kiválasztott munkát kiemeli az ütemtervből és máshová illeszti be véletlenszerűen választott gép és pozíció szerint.
- N_2 a belső határidőt túllépő munkák közül véletlenszerűen kiemeli egyet az ütemtervből és máshová illeszti be.
- N_3 egy véletlenszerűen választott gépen megváltoztatja a munkák végrehajtási sorrendjét (véletlen szám által kijelölt hosszú párcsere-láncolatot végez el).
- N_4 egy véletlenszerűen választott gépen felcseréli két véletlenszerűen választott szomszédos munka sorrendjét.
- N_5 a belső határidőt túllépő munkák közül véletlenszerűen kiválaszt egyet és egyet előre lépteti a végrehajtási sorrendben.
- N_6 egy véletlenszerűen választott gépen engedélyez egy véletlenszerűen választott letiltott műszakot.
- N_7 egy véletlenszerűen választott gépen letilt egy véletlenszerűen választott engedélyezett műszakot.
- N_8 egy véletlenszerűen választott gépen engedélyez egy véletlenszerűen választott korábban letiltott műszakot és a rákövetkező korábban engedélyezett műszakot letiltja.

A módosító operátorok kiválasztását egy fontossági sorrendet kijelölő prioritáslista (*Priority_List*) és kvázi-véletlenszám generátor együttműködése határozza meg. A prioritáslista az operátorok kiválasztási valószínűségét írja le.

Ha a módosító operátorok által készített valamelyik szomszédos s ütemterv szerepel a tabulistán (*Taboo_List*), akkor az algoritmus azt nem értékeli ki (már ismeri), ellenkező esetben felkerül a tabulistára, és ha a megengedett tabuelemek száma elérte a maximális értéket, akkor a legkorábban felvett listaelem törlődik. A tabulista azt a célt szolgálja, hogy a keresési folyamat ne ragadjon bele egy lokális optimumba, hanem át tudjon rajta haladni és meghatározott lépésszámon belül ne térhessen vissza oda.

A szimulációt és a célfüggvények értékének kiszámítását követően, ha az aktuális ütemterv jobb, mint az adott kiterjesztés addigi legjobb ütemterve ($s < s_k$), akkor az aktuális ütemterv megjegyzésre kerül ($s_k \leftarrow s$). A kiterjesztés legjobb ütemterve lesz a következő lépés

kiterjesztésének a kiindulási bázisa ($s_0 \leftarrow s_k$), és ha ez a megoldás jobb, mint a keresés során megtalált legjobb megoldás ($s_k < s^*$), akkor ez kerül megjegyzésre ($s^* \leftarrow s_k$).

A több összetevőből felépülő célfüggvény-rendszer kezelésére a szakirodalomban elérhető *relatív minősítésre* alapozott matematikai modellt használtam [24]. A módszer alapelve az, hogy két különböző megoldás összehasonlításakor az egyik megoldásnak a másikkhoz viszonyított (relatív) jóságának számértéke alapján dönthető el, hogy melyiket tekinthető jobb megoldásnak.

A modell formális leírása a következő:

$$f_k : S \rightarrow \mathfrak{R}^+ \cup \{0\}, \forall k \in \{1, 2, \dots, K\}, f_k \rightarrow \min \quad (2)$$

$$D : \mathfrak{R}^2 \rightarrow \mathfrak{R}, D(a, b) = \begin{cases} 0, & \text{ha } \max(a, b) = 0 \\ \frac{b-a}{\max(a, b)}, & \text{egyébként} \end{cases}, \quad (3)$$

$$F : S^2 \rightarrow \mathfrak{R}, F(s_x, s_y) = \sum_{k=1}^K (w_k \cdot D(f_k(s_x), f_k(s_y))), \quad (4)$$

ahol:

- S – a megoldások halmaza;
- f_k – a k . célfüggvény minimalizálandó alakban megadva;
- K – a célfüggvények száma;
- s_x, s_y – két különböző megoldás;
- w_k – az f_k célfüggvény fontosságát kifejező nem negatív érték (prioritás);
- $F(s_x, s_y)$ – az s_y megoldás s_x megoldáshoz viszonyított relatív jósága.

Az $F(s_x, s_y)$ előjeles függvényérték felhasználásával az alapvető relációs operátorok értelmezése kiterjeszhető az S -beli s_x és s_y megoldásokra a következőképpen:

$$(s_y ? s_x) := (F(s_x, s_y) ? 0) \quad (5)$$

ahol ? tetszőleges relációs operátort helyettesíthet (<, ≤, >, ≥, =, ≠).

A lehetséges ütemtervek halmazán értelmezett relációs operátorok felhasználásával a különböző ütemtervek páronkénti összehasonlítása és értékelése egyszerűen megoldható.

3.4 Futási eredmények és illusztratív példák

3.4.1 A többcélű összehasonlító operátor működése

Az összehasonlító operátor lényege, hogy minden egyes célfüggvény szempontjából meghatározza a két valós vektor elemenkénti eltérésének relatív (a nagyobb értékhez viszonyított) mértékét, majd ezeket az eltéréseket összesíti a teljes vektorra.

2. táblázat: 1. példa a többcélú összehasonlító operátor működésére.

Prioritások	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
	1	1	1	1	1	1	1	1
Célfüggvények	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$f_k(s_x)$	0	0	0	17	680	9,45	1353,97	26,15
$f_k(s_y)$	0	0	0	17	635	9,53	1525,63	26,30
$D(f_k(s_x), f_k(s_y))$	0	0	0	0	-0,0662	0,0084	0,1125	0,0057
$w_k * D(f_k(s_x), f_k(s_y))$	0	0	0	0	-0,0662	0,0084	0,1125	0,0057
$F(s_x, s_y)$	0,0604							

A 2. táblázatban vázolt 1. példában az első négy célfüggvény értéke mindkét ütemterv esetében megegyezik. Az f_5 szempontjából az s_y jobb megoldás mint s_x , viszont f_6 , f_7 és f_8 szempontjából éppen fordított a helyzet. A célfüggvények egyformán fontosak, ezért prioritásértékük rendre 1-re van beállítva. A vizsgált két ütemterv jósága nagyon közel van egymáshoz, ez abból is látható, hogy $F(s_x, s_y)$ értéke közel van a nullához. Mivel $F(s_x, s_y) > 0$ így $s_y > s_x$, vagyis ebben a szituációban az s_x ütemterv tekinthető jobb megoldásnak.

3. táblázat: 2. példa a többcélú összehasonlító operátor működésére.

Prioritások	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
	5	5	10	5	5	5	1	10
Célfüggvények	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$f_k(s_x)$	0	0	0	17	680	9,45	1353,97	26,15
$f_k(s_y)$	0	0	0	17	635	9,53	1525,63	26,30
$D(f_k(s_x), f_k(s_y))$	0	0	0	0	-0,0662	0,0084	0,1125	0,0057
$w_k * D(f_k(s_x), f_k(s_y))$	0	0	0	0	-0,3309	0,0420	0,1125	0,0570
$F(s_x, s_y)$	-0,1194							

Az összehasonlító módszer figyelembe veszi egyrészt az eltérések relatív mértékét ($D(f_k(s_x), f_k(s_y))$), másrészt a célfüggvények előírt fontosságát is (w_k). A 3. táblázatban szereplő 2. példában a célfüggvény-értékek változatlanok, viszont a célfüggvények fontosságát különböző prioritás-értékek fejezik ki. Ezeket figyelembe véve, az $F(s_x, s_y) = -0,1194$ eredmény adódik, melyből következik, hogy mivel $F(s_x, s_y) < 0$ így $s_y < s_x$, vagyis az adott szituációban az ütemező a definiált kritérium-rendszer alapján az s_y ütemtervet tekinti jobb megoldásnak.

4. táblázat: 3. példa a többcélú összehasonlító operátor működésére.

Prioritások	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
	5	5	10	5	5	5	3	10
Célfüggvények	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$f_k(s_x)$	0	0	0	17	680	9,45	1353,97	26,15
$f_k(s_y)$	0	0	0	17	635	9,53	1525,63	26,30
$D(f_k(s_x), f_k(s_y))$	0	0	0	0	-0,0662	0,0084	0,1125	0,0057
$w_k * D(f_k(s_x), f_k(s_y))$	0	0	0	0	-0,3309	0,0420	0,3376	0,0570
$F(s_x, s_y)$	0,1057							

Érdemes megfigyelni, hogy ha a w_7 értékét 1-ről megnöveljük 3-ra (az átfutási időket nagyobb mértékben akarjuk figyelembe venni), a nagyon közeli megoldások megítélése alapvetően megváltozik (4. táblázat). Mivel $F(s_x, s_y) > 0$ így $s_y > s_x$, vagyis ebben a szituációban az s_x ütemterv tekinthető jobb megoldásnak.

Amennyiben ezeket a döntéseket az ütemező algoritmusnak egy közbenső lépésben kell meghoznia, látható módon ugyanazokat az ütemterveket a prioritásoktól függően másképpen ítéli meg, és más-más irányban folytatja a keresést. A prioritásértékek finomhangolásával a felhasználó viszonylag egyszerű módon képes kifejezni aktuális elvárásait.

3.4.2 A célfüggvény-értékek alakulása a keresési folyamatban

Vegyük a következő viszonylag egyszerű példát:

A szerelősorok egy heti termelési terve ismert. Mindkét szerelősor naponta két (nappalos, délutános) műszakban dolgozik. Adottak az input adatok (ezeket itt nem részletezem). A belső rendelések száma 20, a helyben gyártandó munkadarabok száma összesen 8300. Minden rendelés önálló munkát jelent, és a logisztikai egység mérete legyen 1 munkadarab. Tételezzük fel, hogy a négy alkatrészgyártó gép közül kettő nem dolgozik, mert felújítás alatt állnak. A műveletközi tároló átlagos kapacitása legyen 1500 munkadarab. A célfüggvény-rendszer összetevői és azok fontosságát kifejező prioritásértékek az 6. ábrán láthatók.

Ütemezési célfüggvények	Prioritás
Késő rendelések száma	5
Késések összege [perc]	5
Max. késés [perc]	10
Átállások száma	5
Átáll. összeitő [perc]	5
Átl. gépkihaszn. [%]	5
Átl. átfut. idő [perc]	1
Súly. műszakszám	10

6. ábra: Célfüggvények és prioritásértékek.

A keresési algoritmus paraméterei a következők:

- A lépések maximális száma: 200.
- Adott lépésben generált kiterjesztett megoldások maximális száma: 200.
- A tabulista maximális elemszáma: 500.

A legjobb megoldást eredményező lépés sorszáma 52, ezen belül a legjobb kiterjesztés sorszáma 197. A megvizsgált ütemtervek száma 2892. Az algoritmus futási ideje: 1 perc 53 másodperc. Az alkalmazott futási tesztkörnyezet fontosabb jellemezői a következők:

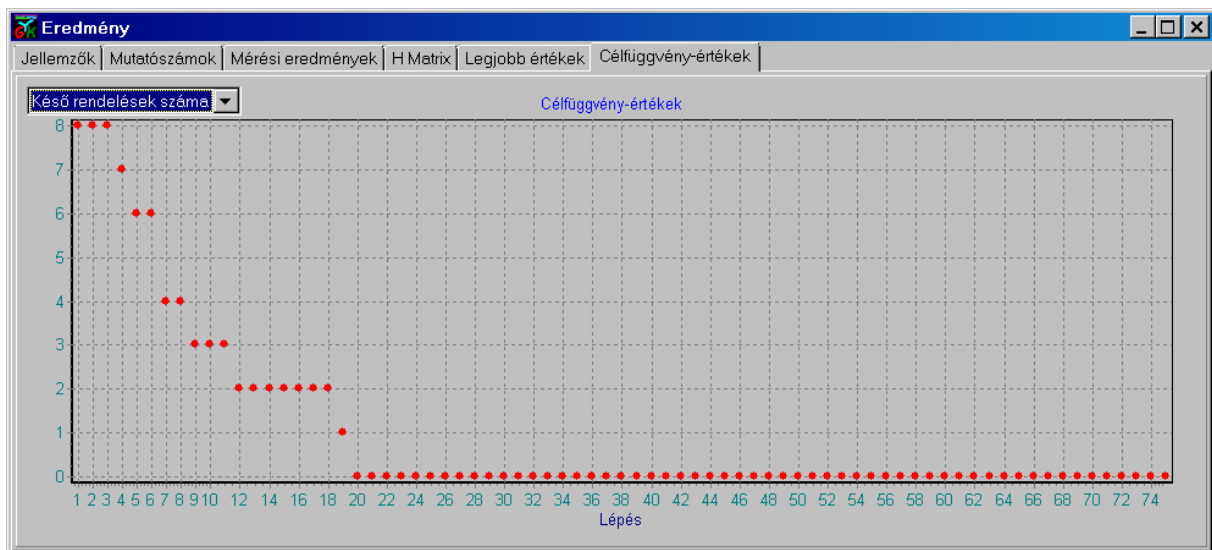
- Intel Core 2 DUO T9550 2,66 GHz CPU,
- 4 GB RAM,
- Microsoft Windows 7 Enterprise (32 bit) OS.

A kiindulási s_0 ütemterv és a keresés során legjobbnak ítélt s^* ütemterv célfüggvény-értékei az 5. táblázatban láthatók. A heurisztikus algoritmus által készített kezdeti megoldáshoz képest a kereső algoritmus minden célfüggvény tekintetében jelentős mértékben tudott javítani a teljesítmény-mutatókon.

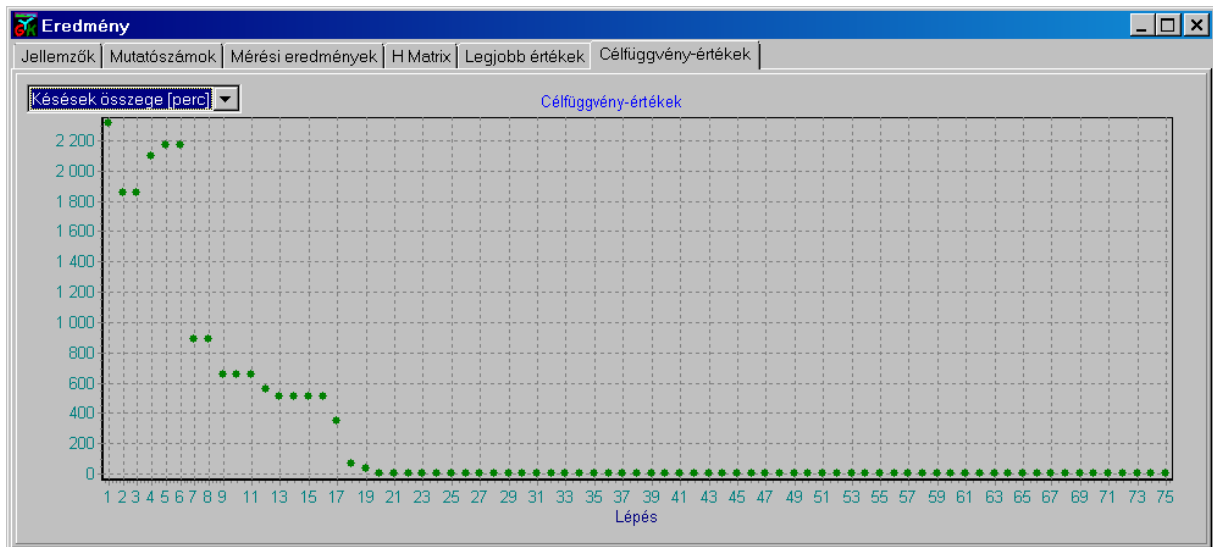
5. táblázat: A kiindulási megoldás és a legjobbnak ítélt megoldás összehasonlítása.

Prioritások	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8
	5	5	10	5	5	5	3	10
Célfüggvények	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$f_k(s_0)$	9	4414,93	794	19	850	59,68	1666,53	40,25
$f_k(s^*)$	0	0	0	17	635	14,62	1391,8	22,45
$D(f_k(s_0), f_k(s^*))$	-1	-1	-1	-0,1053	-0,2529	-0,7550	-0,1649	-0,4422
$w_k * D(f_k(s_0), f_k(s^*))$	-5	-5	-10	-0,5263	-1,2647	-3,7751	-0,1649	-4,4224
$F(s_0, s^*)$	-30,1534							

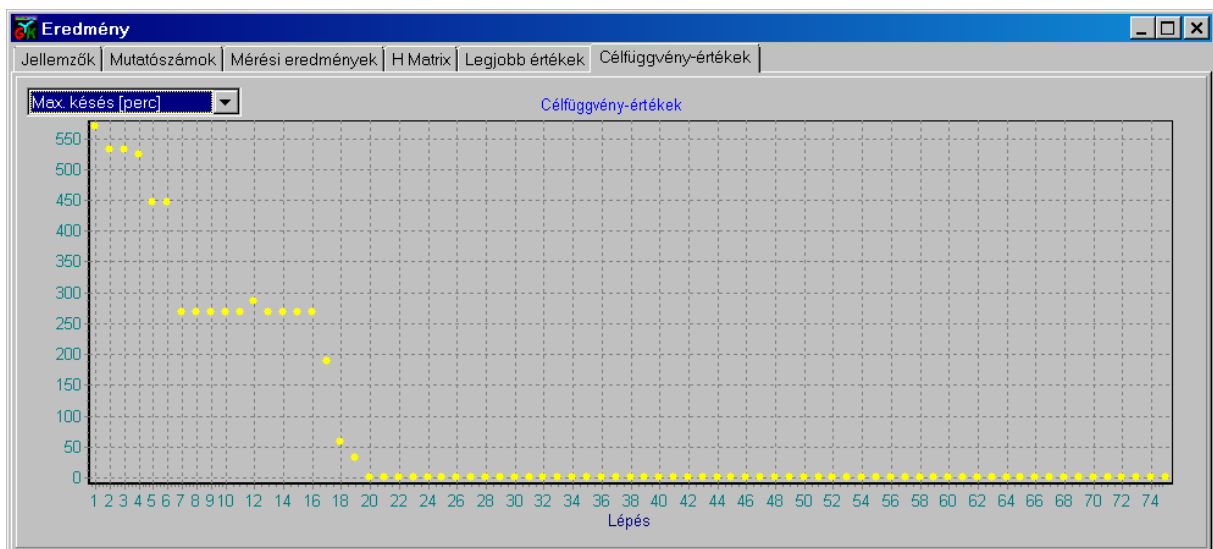
A 7.- 14. ábrák a célfüggvény-értékek változását mutatják a keresés során megtett lépések függvényében. A diagramok az adott lépésen belül megvizsgált kiterjesztett ütemtervek részleteit nem szemléltetik, az adott lépéshez a kiterjesztés legjobb (s_k) megoldásának értékeit rendeli hozzá.



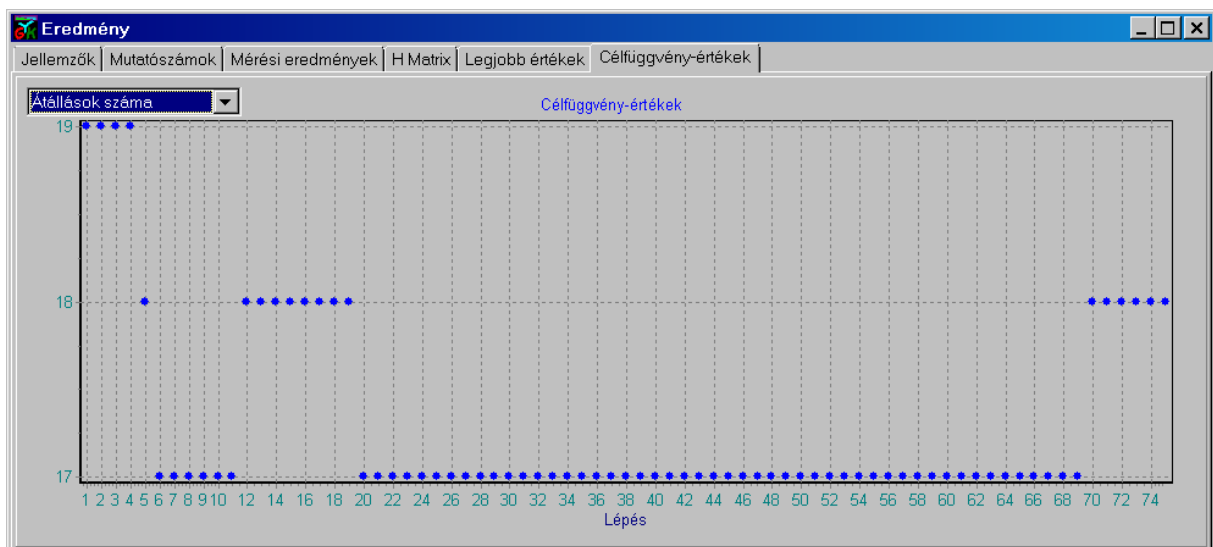
7. ábra: A határidőt túllépő sorozatok számának változása.



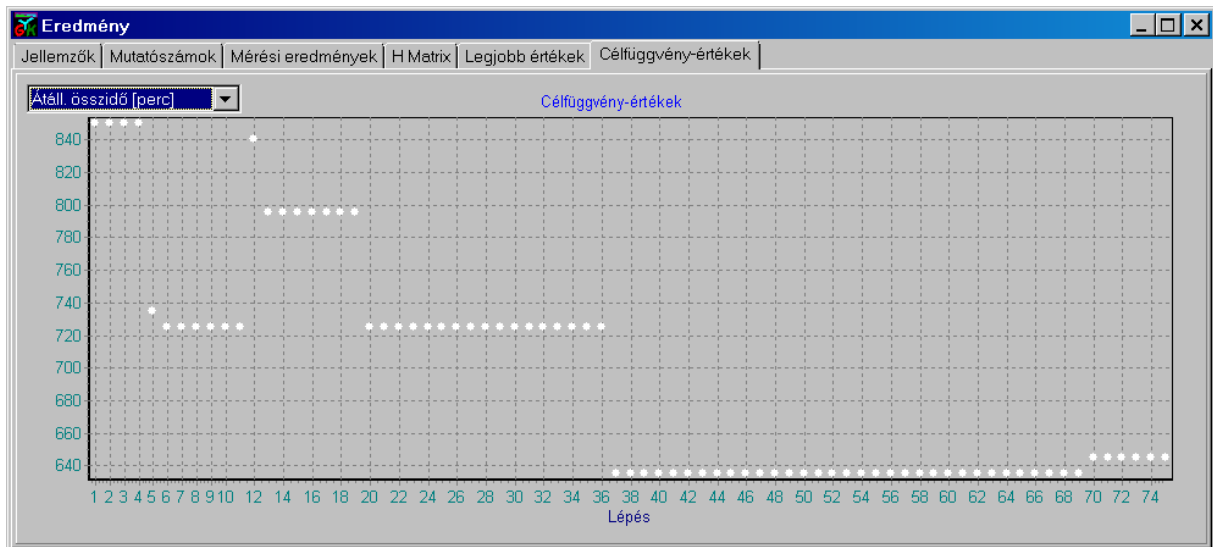
8. ábra: A csúszások időösszegének változása.



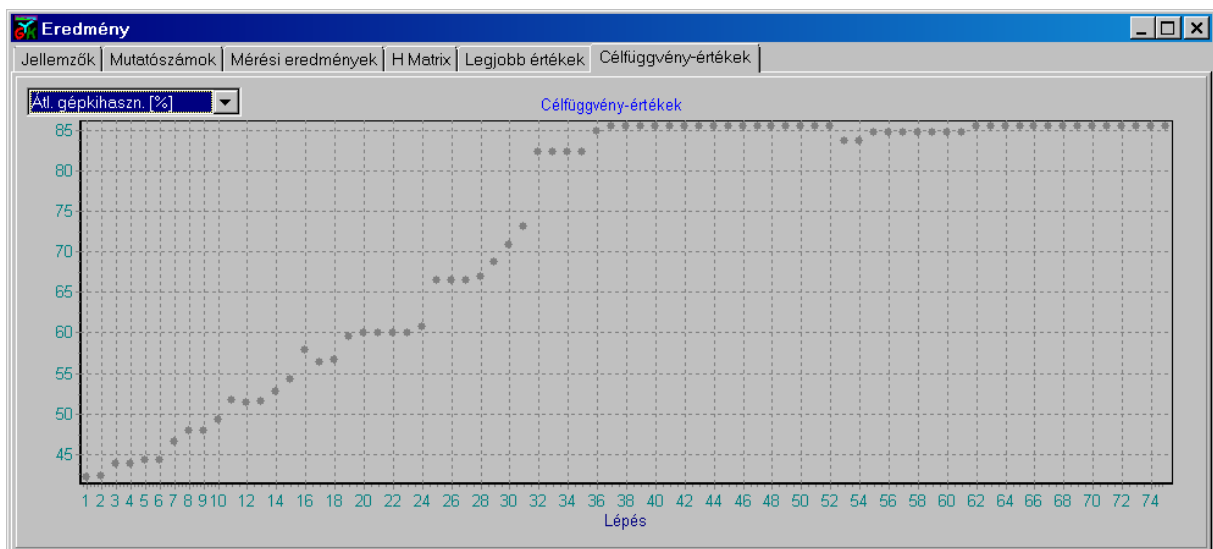
9. ábra: A legnagyobb határidő-túllépés (csúszás) változása.



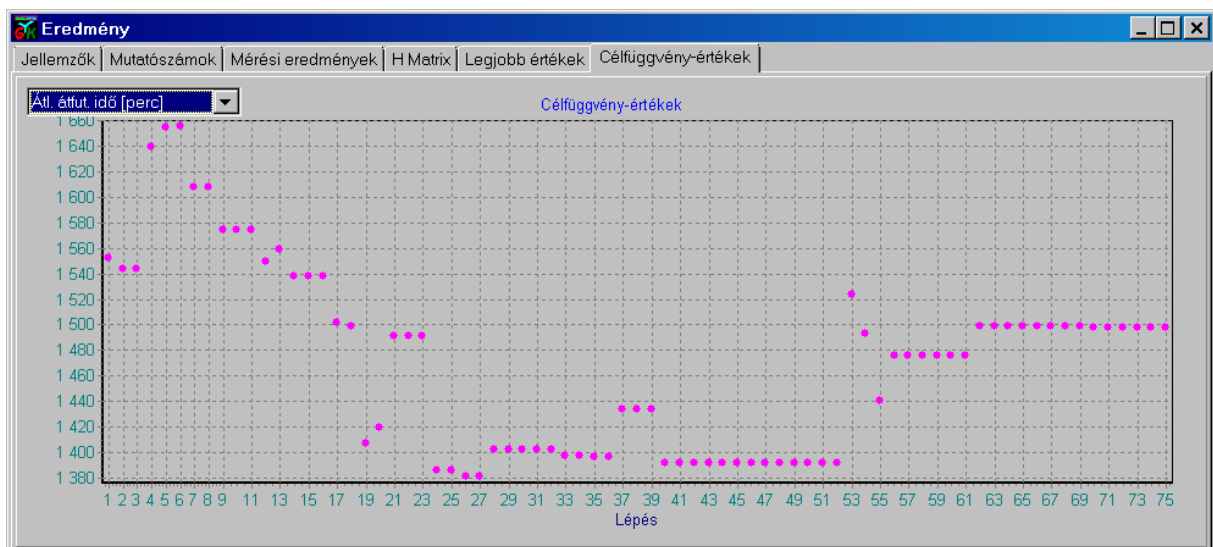
10. ábra: A gépátállások számának változása.



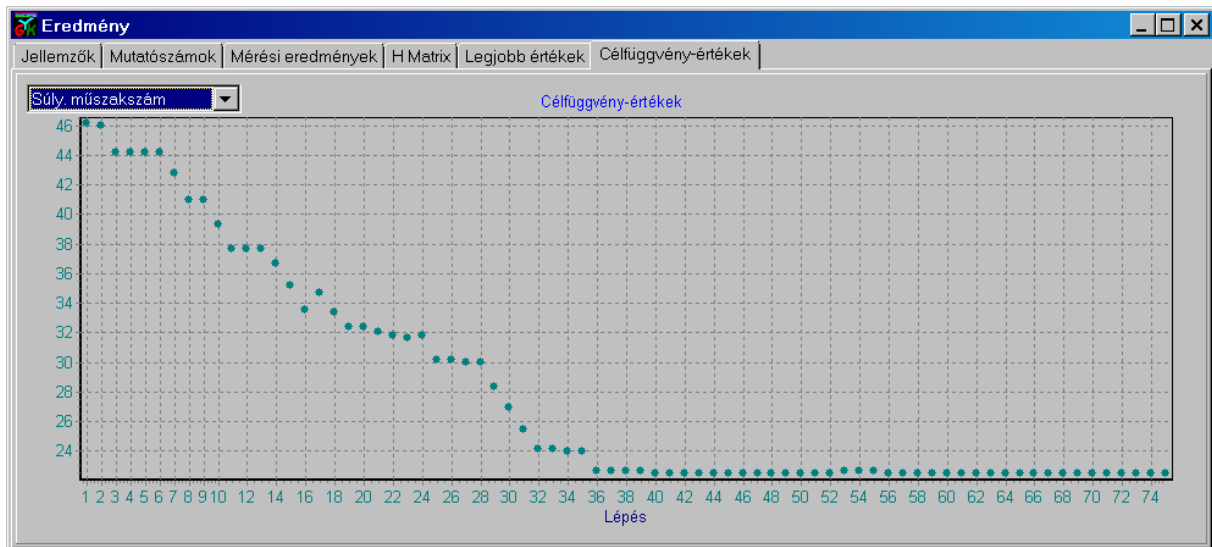
11. ábra: A gépátállítások időösszegének változása.



12. ábra: A gépek átlagos kihasználtságának változása.



13. ábra: A sorozatok átlagos átfutási idejének változása.

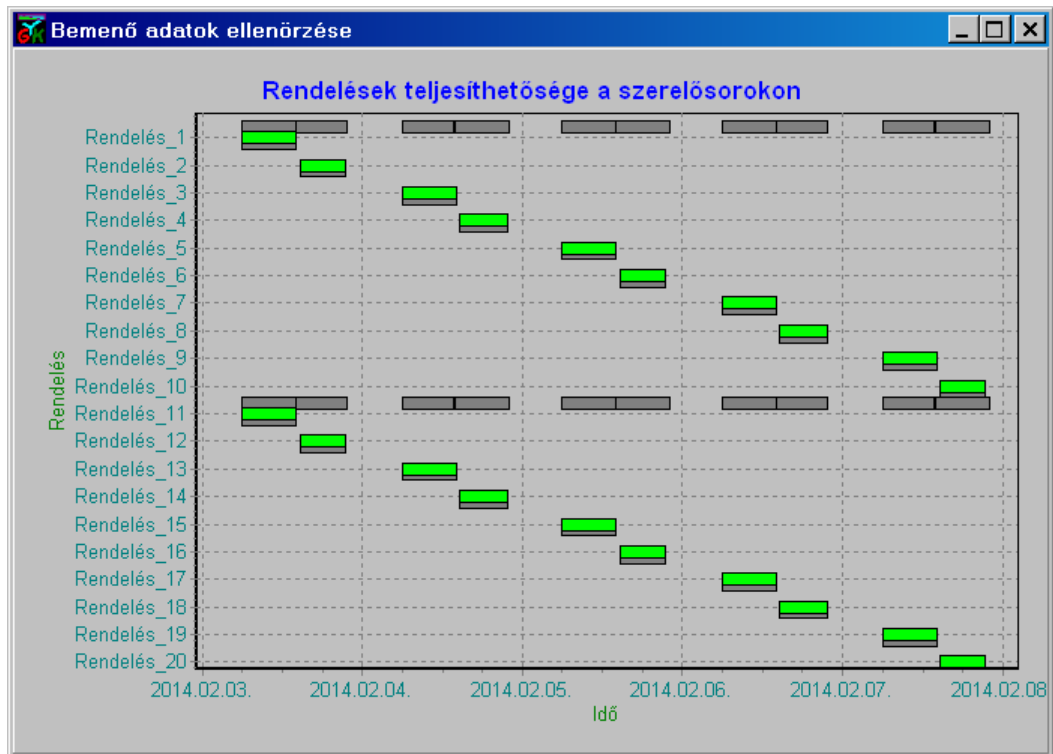


14. ábra: A súlyozott műszakszám változása.

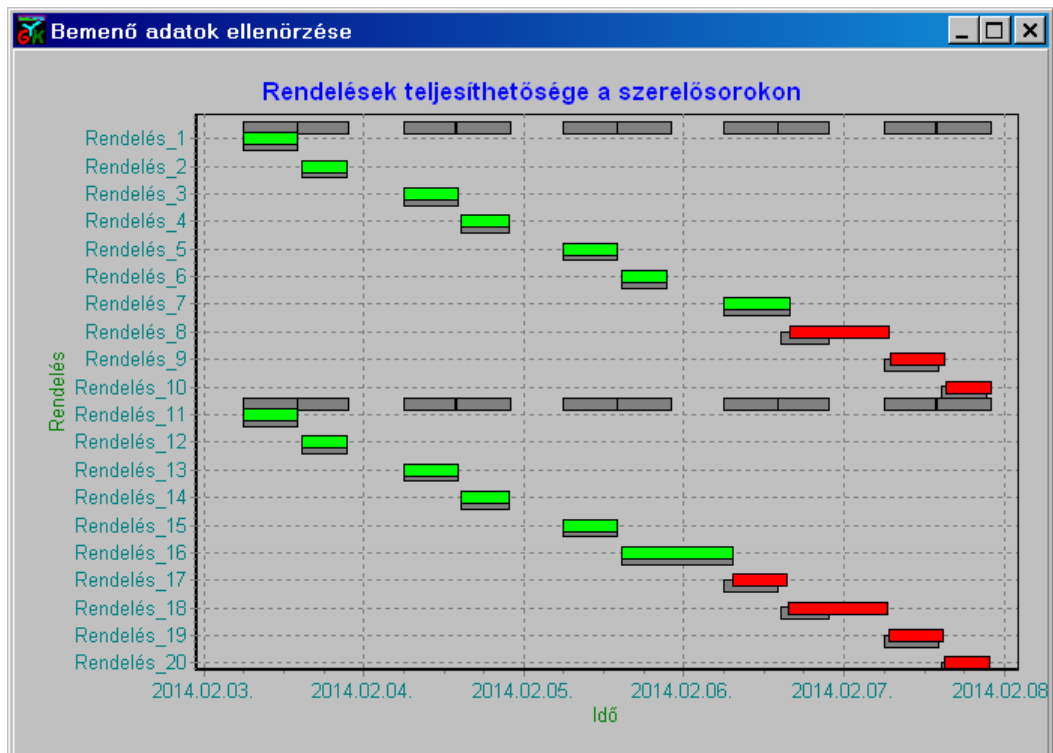
Az 52. lépéstől kezdve a csúszásokra és a műszakszámra vonatkozó célfüggvények értékei nem változtak, míg az átállásokra, az átfutási időkre és a gépkihasználatra vonatkozó célfüggvények értékei romlást mutattak. A megtalált legjobb megoldás az 52. lépés után összességében már nem javult.

3.4.3 A vizsgált modell szimulációjának fontosabb jellemző eredményei

Az értekezésben vizsgált modell fontos tulajdonsága, hogy a szerelés termelési terve bemenő adatként meghatározza a teljes termelési folyamat teljesítményének tartományait. Emiatt az input termelési terv megvalósíthatóságát ellenőrizni kell. Egy megadott termelési terv szimulációs ellenőrzésének Gantt-diagramja a 15. ábrán látható. Ennek légyege, hogy a szerelési folyamat időbeli végrehajthatósága ellenőrizhető azt feltételezve, hogy a szükséges alkatrészek megfelelő időben rendelkezésre állnak. Erre azért van szükség, mert ha a bemenetként megadott szerelésre vonatkozó termelési terv a szimuláció szerint nem valósítható meg, akkor még időben lehet változtatást kérni a felsőbb termelési tervezési szinttől. A 15. ábra egy teljesíthető, míg a 16. ábra egy eleve teljesíthetetlen termelési tervet ábrázol. A diagram a teljesíthető rendeléseket zöld színnel, a csúszással induló rendeléseket piros színnel jeleníti meg. Az egyszerűbb átláthatóság kedvéért a rendelések elsődlegesen a szerelősor szerint, másodlagosan pedig az indítási sorrend szerint rendezve kapták meg a belső azonosítójukat (Rendelés_1, ..., Rendelés_20).



15. ábra: Teljesíthető szerelési (termelési) terv.



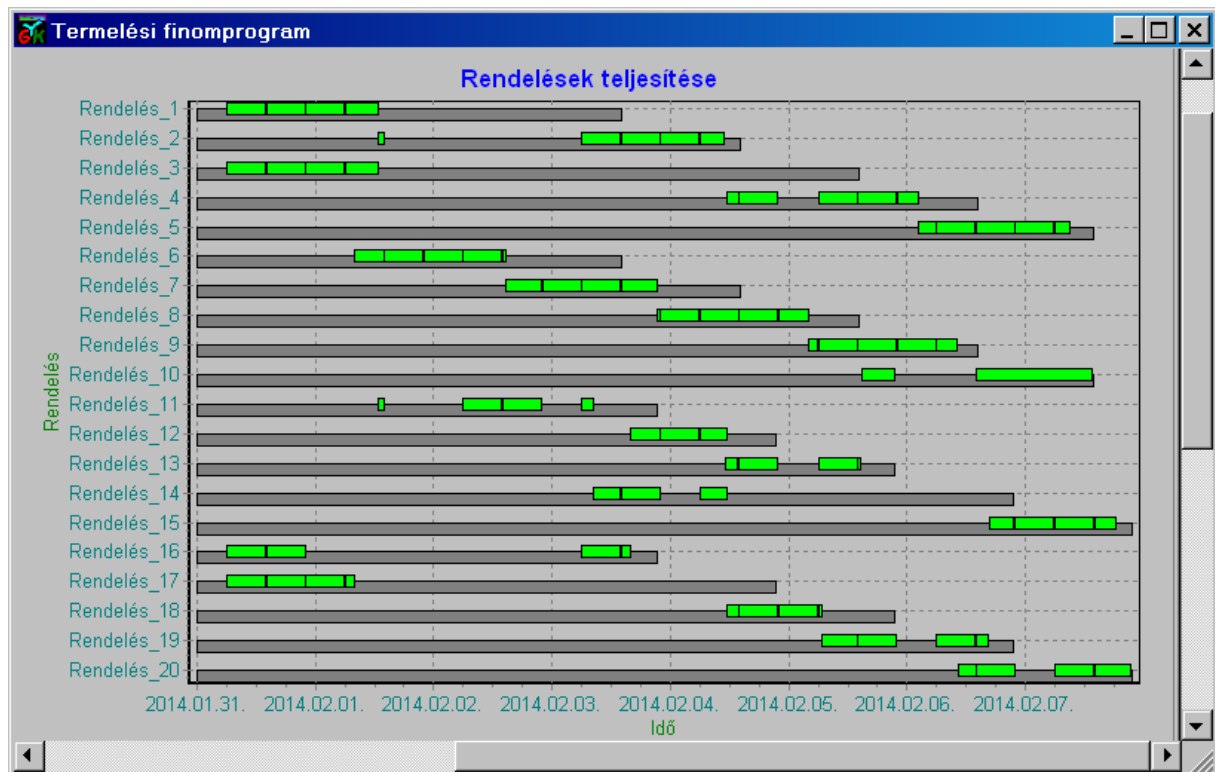
16. ábra: Teljesíthetetlen szerelési (termelési) terv.

A következő példában a 15. ábrán feltüntetett teljesíthető szerelési terv adja a belső rendeléseket. A gyártórendszerben négy alkatrészgyártó gép szolgál ki két szerelősort. A bemenő adatok között szereplő termelési sebességekre az jellemző, hogy az alkatrészgyártás műveleti idői rendre jóval nagyobbak mint a szerelési műveletek időtartamai. A szerelősorok

folyamatos működésének feltétele, hogy a szükséges alkatrészek gyártását jóval a szerelési folyamat megkezdése előtt el kell kezdeni és jelentős méretű alkatrésztárolóra van szükség.

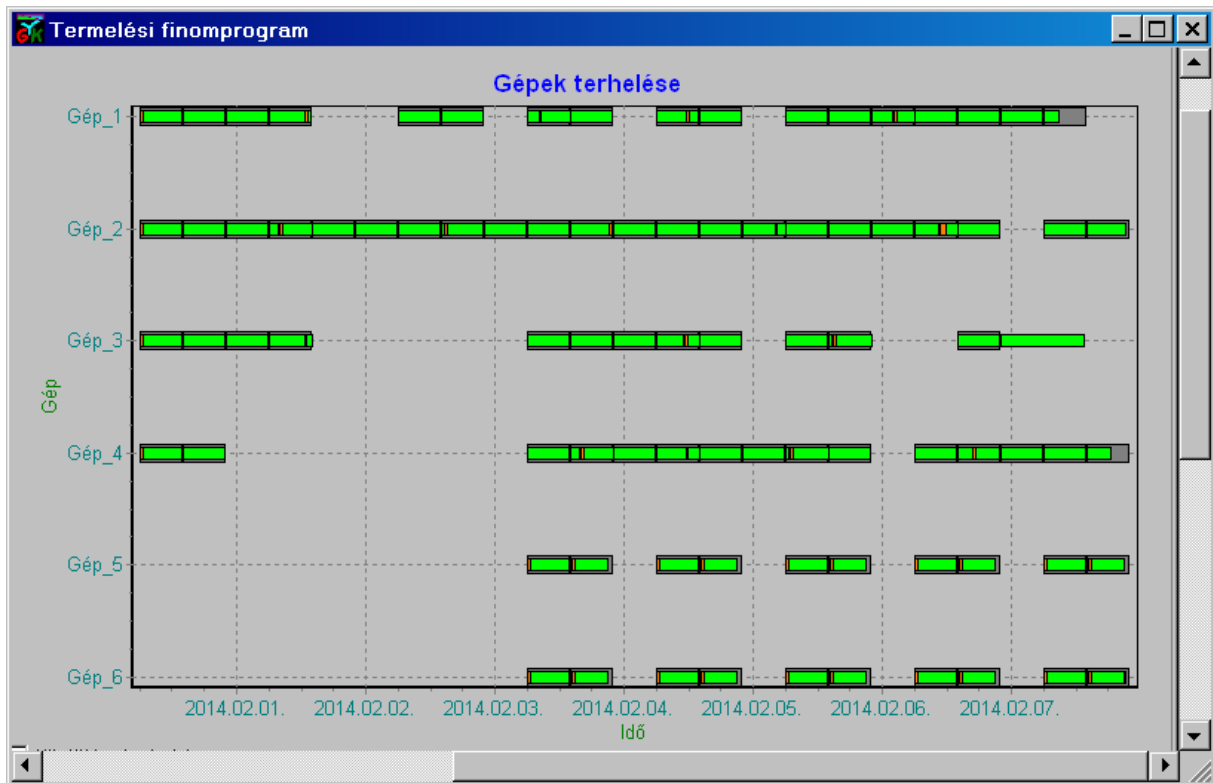
A MOMOTS algoritmusra alapozott ütemező szoftver által készített megoldás szimulációjának fontosabb eredményeit a 17.-21. ábrák foglalják össze.

A 17. ábrán látható a rendelt alkatrészsorozatok gyártásának idődiagramja. Ezen az ábrán a szürke téglalapok a határidőig tartó időablakot, a zöld téglalapok pedig a tényleges munkavégzést szimbolizálják.



17. ábra: A belső rendelések teljesítésének idődiagramja.

A 18. ábrán látható gép-terhelési diagramból kiolvasható, hogy a szerelősorok (Gép_5, Gép_6) egy hetes, napi két műszakos várakozásmentes működését a négy alkatrészyártó gép (M_1, ..., M_4) akkor tudja biztosítani, ha már az előző hét utolsó munkanapjától kezdve (a hétvégét is beleértve) előre dolgozik. Ezen az ábrán a szürke téglalapok a tervezett műszakokat jelentik, a zöld téglalapok a tényleges munkavégzést, a narancssárga téglalapok pedig a gépátállítást mutatják.



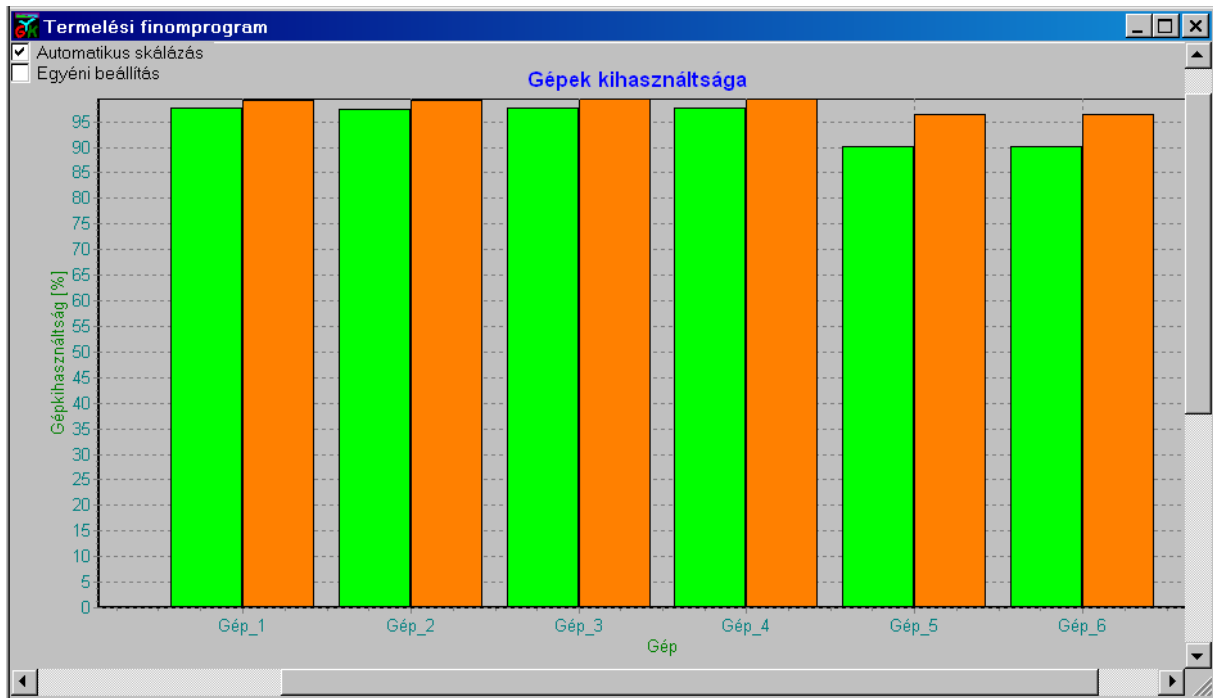
18. ábra: A gépek terhelése.

A 19. ábra jeleníti meg példaként az ütemező által kialakított műszakbeosztást, amely a Gép_3 alkatrészgyártó gépre vonatkozik.

Gépek:	Műszakok:
BK_MAXI Gép_1	<input checked="" type="checkbox"/> Péntek 1. műszak 2014.1.31.6:0-13:59
BK_NORMAL Gép_2	<input checked="" type="checkbox"/> Péntek 2. műszak 2014.1.31.14:0-21:59
BK_3 Gép_3	<input checked="" type="checkbox"/> Péntek 3. műszak 2014.1.31.22:0-5:59
BK_4 Gép_4	<input checked="" type="checkbox"/> Szombat 1. műszak 2014.2.1.6:0-13:59
E_1 Gép_5	<input type="checkbox"/> Szombat 2. műszak 2014.2.1.14:0-21:59
E_2 Gép_6	<input type="checkbox"/> Szombat 3. műszak 2014.2.1.22:0-5:59
	<input type="checkbox"/> Vasárnap 1. műszak 2014.2.2.6:0-13:59
	<input type="checkbox"/> Vasárnap 2. műszak 2014.2.2.14:0-21:59
	<input type="checkbox"/> Vasárnap 3. műszak 2014.2.2.22:0-5:59
	<input checked="" type="checkbox"/> Hétfő 1. műszak 2014.2.3.6:0-13:59
	<input checked="" type="checkbox"/> Hétfő 2. műszak 2014.2.3.14:0-21:59

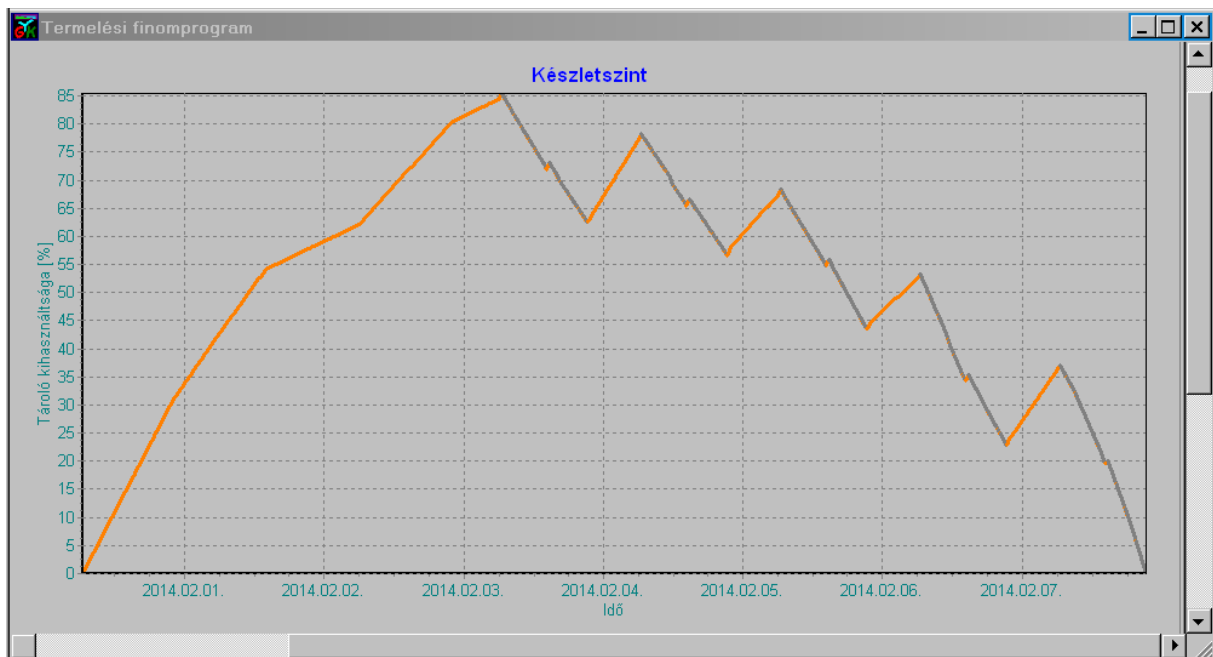
19. ábra: Adott gép ütemezett műszakbeosztása.

A vizsgált termelési finomprogram a 20. ábrán látható gépkihasznátságokat eredményezi. Az oszlopdiagram a tervezett műszakok időösszegének százalékos kihasználtságát mutatja. A narancssárga oszlopok értékeinek számítási algoritmus a gépátállításokat hasznos műveletként vette figyelembe, míg a zöld oszlopok értékeiben a gépátállítások vesztéséidőként vannak figyelembe véve (csak a tényleges operációk időösszege számít hasznos időnek).

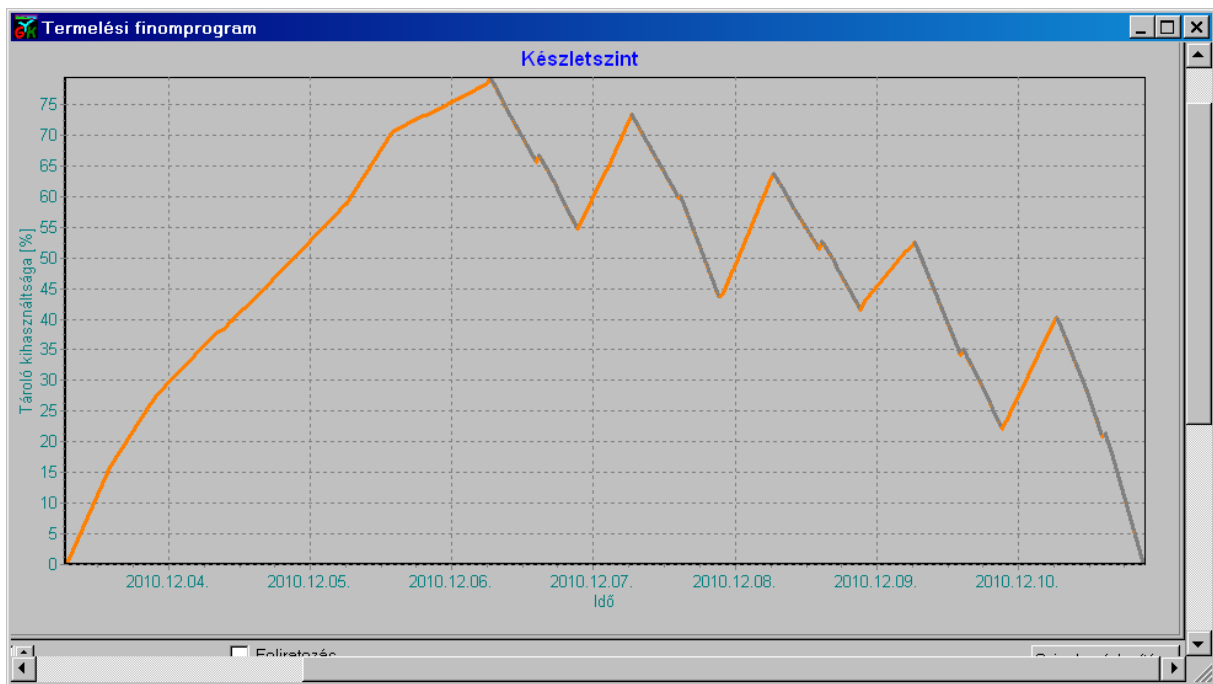


20. ábra: A gépek kihasználtsága az ütemezett műszakokban.

A 21. ábrán látható a tároló készlet szint-diagramja, melyből kiolvasható, hogy a gyártórendszer jelentős készletet halmoz fel már a szerelési folyamat megkezdése előtt (puffer hatás). A tároló készlet szintje rohamosan csökken azokban az időszakokban, amikor a szerelősorok dolgoznak, annak ellenére, hogy szereléssel egyidejűleg az alkatrészgyártó gépek is működnek (töltik a tárolót). Jól látható az ábrán az is, hogy az éjszakai műszakokban, amikor a szerelősorok állnak, a készlet szintje szinte periodikusan emelkedik („fűrészfog-jelleget” mutat).



21. ábra: Az alkatrész tároló készlet szintjének alakulása az ütemezett időintervallumban.



22. ábra: Az alkatrészraktár készletszintjének alakulása 80 %-os kapacitáskorlát mellett megtalált legjobb ütemterv esetében.

A tároló kapacitás-korlátjának csökkentésével egyre nehezebbé vált az ütemezési feladat megoldása határidő túllépés nélkül. A tároló kapacitását az eredeti érték 80 %-ra lehetett lecsökkenteni az adott szituációban, ennél kisebb érték mellett az ütemező nem tudta megoldani a feladatot. A 80 %-os kapacitáskorlát mellett megtalált legjobb ütemterv készletdiagramja a 22. ábrán látható. A célfüggvények közül a keresés során csak a határidő túllépést minimalizáló és az átfutási időt minimalizáló komponensek prioritását állítottam nullától nagyobb értékre, a többi nulla értéket kapott, mert a súlyozott műszakszám és az átállások minimalizálása ebben a szituációban hátráltatta a feladat megoldását.

3.5 Összefoglalás és következtetések

Kutatómunkám első szakaszában az igény szerinti rugalmas gyártás egy kiterjesztett, speciális termelésütemezési feladatosztály modellezésével és megoldásával foglalkoztam. A modellezett erőforrás-korlátos gyártó-szerelő rendszer legfontosabb jellemzői közé tartoznak az alternatív technológiai útvonalak, a korlátozottan rendelkezésre álló gépek, a korlátozott méretű műveletközi tárolók, az eltérő műveletvégzési, átállítási és anyagmozgatási idők valamint a szigorú belső határidős munkák. A feladat modellezése és a megoldási módszer kidolgozása során a változó vevői igényekhez és az erőforrás korlátokhoz rugalmasan alkalmazkodó ütemezési módszereket részesítettem előnybe. A megoldások (ütemtervek) előállítására egy továbbfejlesztett végrehajtás-vezérelt szimulációt magába foglaló többcélú, többoperátoros keresési algoritmust tartalmazó szoftver alkalmazást használtam. Az értekezés 3. fejezetében ismertettem a vizsgált feladat modelljének legfontosabb jellemzőit, valamint a megoldási módszer koncepcióját, legfontosabb algoritmusait. Illusztratív példák és futási eredmények bemutatásával körvonalaztam a valós ipari mintaadatokon elvégzett vizsgálatokat

és az elért eredményeket. A *MOMOTS* kereső algoritmus hatékonyságát klasszikus ütemezési benchmark feladatokon végzett tesztelések eredményei is alátámasztják (III. Melléklet).

Az integrált ütemezés és műszakbeosztás tervezési (optimalizálási) feladatot keresési feladatként fogalmaztam meg, melyben definiáltam a döntési változókat, a kiterjesztett korlátozásokat és a megfogalmazott menedzser célokat. Javaslatot dolgoztam ki a rendelkezésre álló szoftver alkalmazás továbbfejlesztésére. Az alkalmazott szimulációs algoritmus magába zárja és elrejtja a termelési folyamat teljesítmény-mérésének sajátos részleteit, ezáltal könnyebben kezelhetővé teszi a problémát. A szimuláció futási sebessége alapvető fontosságú a megoldási koncepció hatékonyságának szempontjából. A jelenlegi (egyetlen szalon futó) változat egy-két perces futási időt eredményez a teljes tervezési (keresési) folyamatra vetítve. A jövőben érdemes további kutatásokat folytatni a szimuláció sebességének növelése érdekében.

A feladat bonyolultsága és a jó megoldást befolyásoló tényezők sokasága miatt a klasszikus heurisztikus felépítő algoritmusok (a különböző szélsőséges szituációk esetében) nem alkalmazhatók. A többcélú keresési módszerek – rugalmasságuknak köszönhetően – képesek alkalmazkodni a változó feltételekhez és célfüggvény-rendszerekhez. Több célfüggvény egyidejű figyelembe vétele nagyon fontos, mert lehetővé teszi a gyakorlati igények beépítését a modellbe (pl. súlyozott műszakszám számítása). Meg kell azonban jegyezni, hogy a másodlagos (kiszegítő) célok indokolatlanul nagy prioritásértéke jelentősen megnehezítheti a keresési folyamatot. A vizsgált ütemezési feladatokban például a zéró határidő túllépés (csúszás) az elsődleges cél. A többi (másodlagosnak tekinthető cél) igazából akkor válik fontossá, ha több (esetleg sok) ütemterv is teljesíteni tudja az elsődleges célt. A kiegészítő kritériumok a keresés kezdeti szakaszában nehezíthetik is a hatékony keresést kritikus feladatok esetében. További kutatási terveim között szerepel annak vizsgálata is, hogy a kereső algoritmusok működése közben milyen lehetőségei vannak a célfüggvény-rendszerek szisztematikus és/vagy bizonyos határok közötti rugalmas változtatásának.

A vizsgált modell bonyolultsága meglepően nagy eredmény-szóródást okoz. A generált próbafeladatokon végzett kísérletek azt mutatták, hogy különböző szituációkban a gépátállítások minimalizálásának kritériuma, a gépkihasznátságok maximalizálásának kritériuma és a súlyozott műszakszámok minimalizálásának kritériuma ronthatja a határidők betartásának esélyét, míg más esetekben ugyanezek növelhetik is azt. Ez a hatás nagymértékben függ a rendelések tartalmától, a gépek képességeitől és a tárolóra vonatkozó előírásoktól is.

Még tovább nehezíti az ütemezési feladatok megoldását, ha a klasszikus ütemezési feladatokra jellemző gépválasztási és sorrendi döntési változók halmazába bekerülnek a műszakbeosztások előírására vonatkozó döntési változók is. Ez az új kiterjesztés a modell és a megoldási módszer szempontjából további kihívásokat támasztott. A megoldáshoz meg kellett oldani a modellben a döntési változókat leíró objektumok bővítését, a keresési algoritmusban pedig a megoldást módosító (szomszédsági) operátorok újraértelmezését és kibővítését.

A bemutatott modell és megoldási módszer alkalmas a konkrét gépek helyett műhelyek vagy üzemek (esetleg termelési hálózatok) termelési folyamatainak magasabb szintű ütemezésére és összehangolására is. Ennek lényege, hogy a modell entitásai egy magasabb absztrakciós szintre emelhetők, és ezáltal az alkatrészgyártó gép gyártó műhelyt vagy beszállítót, a szerelősor szerelő műhelyt, az alkatrésztároló pedig közbenső vagy beszállítói raktárt jelent.

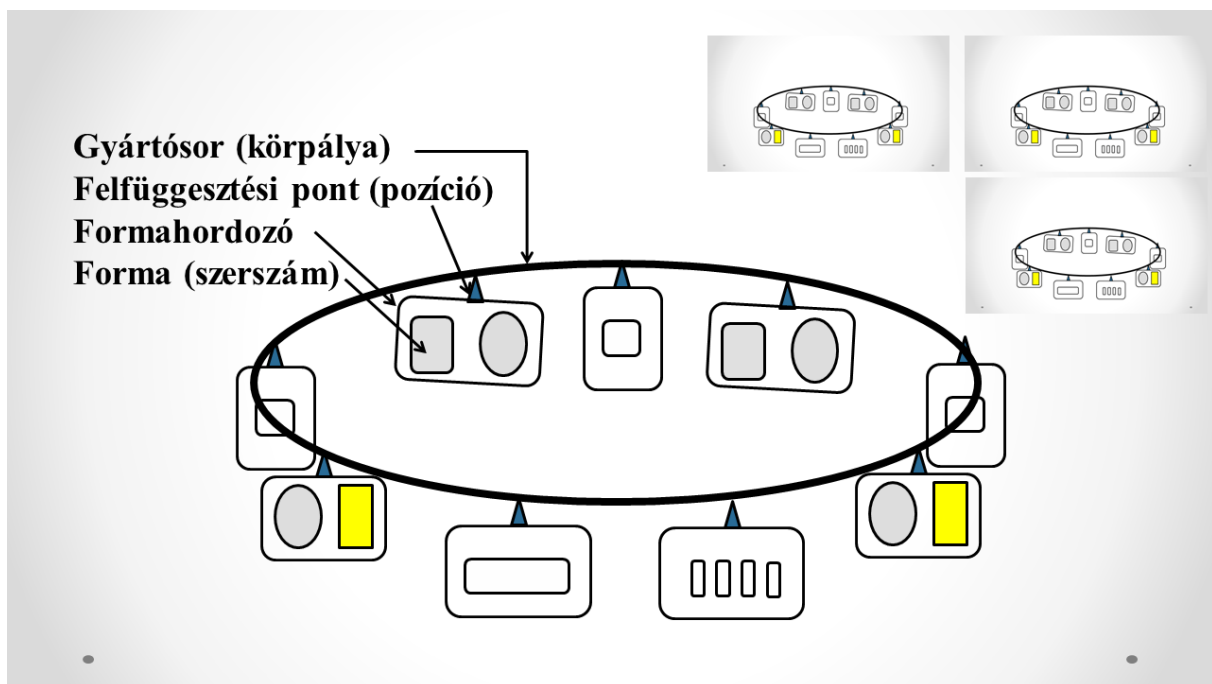
4 Diszkrét gyártási folyamatok és előkészítő tevékenységek integrált termelésprogramozása

4.1 A feladat ismertetése

4.1.1 Alkatrészgyártási folyamatok és előkészítő tevékenységek

A téma kutatásának motivációját a Fehrer Hungaria Járműipari Kft. móri telephelyén felmerülő termelésütemezési feladatok nehézségei adták. A konkrét ipari feladat világossá tette, hogy vannak olyan ütemezési feladatok, amelyeknél a termelési finomprogramozás megfelelő minőségű elvégzéséhez a gyártási főfolyamat mellett a gyártás-előkészítési mellékfolyamat ütemezése is feltétlenül szükséges. A gyártástechnológia követelményeinek elemzése azt is világossá tette, hogy a főfolyamatok és az előkészítő mellékfolyamatok ütemezése egymástól kölcsönösen függő feladatok megoldását követeli meg.

A vizsgált járműipari gyártóműhelyben üléselemeket készítenek különböző márkájú és típusú személyautókhoz. A vállalat egyidejűleg vesz részt több járműipari beszállítói hálózatban. A vevők (jármű-összeszerelő vállalatok) által generált, adott terméktípusra és darabszámra vonatkozó termék-lehívásokat (aktuális rendeléseket) szigorúan előírt szoros határidőre kell teljesíteni.



23. ábra: A vizsgált termelési rendszer elvi vázlata

Az üzem az üléselemek előállítását körpálya kialakítású gyártósorokon valósítja meg (23. ábra). Az üzem több gyártósort (pályákat) üzemeltet, amelyek közösen végzik a rendelések teljesítését. Adott terméktípus általában több pályán is gyártható. Minden egyes pálya saját dedikált konstans sebességgel működik, ezáltal adott számú teljes kört (ciklust) tud elvégezni

egy adott időintervallumban (műszakban). Minden egyes pálya egyedileg adott számú felfüggesztési ponttal (pozícióval) rendelkezik, melyekhez csak adott típusú formahordozó csatlakoztatható. Ezek a formahordozók – kialakítástól függően – egy vagy két oldalasak lehetnek. A formahordozó bal és jobb oldalához speciális szerszámok (formák) rögzíthetők a technológia által meghatározott szabályok szerint. Adott pálya felfüggesztési pontjaihoz csatlakoztatott formahordozók a hozzájuk erősített formákkal együtt mozognak. A formákat ipari robotok töltik fel alapanyaggal a kijelölt állomáson. Egy teljes kör bejárása során különböző állomásokon több művelet elvégzésével készül el a formában (speciális szerszámban) a termék. A következő ciklusban újra megtöltik a robotok a formát alapanyaggal és folytatódik a ciklus.

A technológiai folyamattervek szigorú szabályok formájában írják elő, hogy

- milyen terméktípust,
- melyik pályán,
- milyen pozícióban,
- milyen típusú formahordozón,
- a formahordozó melyik oldalán,
- mely formákkal és
- a formahordozó másik oldalán milyen más terméktípussal együtt lehet gyártani.

A pályák külön-külön definiált időbeosztás (műszakrend) szerint működnek, amelyben nyolc órás időintervallumokat (műszakokat) tekintenek alapegységnek. Az ütemezési döntések hatásköre a műszakrend módosítására nem terjed ki. A menedzsment az ütemezési időszakra pályánként előírja az aktív műszakok listáját. Tovább növeli a korlátozások komplexitását, hogy minden egyes műszakban pályánként csak külön-külön előírt darabszámú pozícióban végezhető el változtatás (csere). Ennek alapegysége egy formahordozóból és a hozzá kapcsolódó formából (vagy formákból) álló úgynevezett „konfiguráció”. Adott konfiguráció levétele egy aktív pozícióból és egy előkészített másik konfiguráció felhelyezése ugyanabba a pozícióba jelent egy teljes cserét (konfiguráció-csere).

A pályák kialakítása különböző, ezáltal a műszakonként teljesíthető teljes körök (ciklusok) száma is eltérő, valamint a pozíciók száma és típusaik is különbözőek. A termékek gyártásához rendszerint típusonként eltérő darabszámú forma és formahordozó áll rendelkezésre. Adott terméktípus esetében a formahordozók közül a megkövetelt tulajdonságok vizsgálata alapján választhatók ki az alkalmasak. A formák alapértelmezés szerint egy kör alatt egy darab adott típusú termék előállítására alkalmasak és a formahordozón egy oldalt foglalnak el. A formák egy része azonban alkalmas többféle termék együttes gyártására is. Ezek összetett formák, amelyek több részből ún. „fészekből” állnak össze. A többfészkes formák különböző vagy azonos típusú termékek együttgyártását támogatják. Létezik olyan forma is, amely kisebb átalakítással alkalmassá tehető különböző típusú termékek gyártására. Ezeket ún. közös használatú (konkurens hozzáférésű) formának nevezik. Egy közös forma egy fizikai objektumot jelent, azonban logikailag többféle terméktípus saját formacsoportjában is egyidejűleg nyilván van tartva.

A termeléshez szükséges konfigurációk előkészítését (szét- és összeszerelését) szakképzett dolgozók végzik. Az előkészítési tevékenység időigényes, így a pályák együtteséből álló üzemben egy adott műszakban az elvégezhető konfiguráció-előkészítések száma szigorú

kapacitáskorláttal határolt. Ez a kapacitáskorlát műszakról-műszakra változik a rendelkezésre álló személyzet műszakbeosztásának függvényében.

A termelésütemező rendszer döntési szabadsága arra korlátozódik, hogy a szigorú korlátozásokat betartva a cserélési lehetőségeket kihasználva igény szerint megtervezze a pozíciókban aktívan használandó konfigurációk összetételét műszakonkénti bontásban. A fent bemutatott példa jól érzékelteti, hogy a fő- és mellékfolyamatok ütemezését – a kölcsönkapcsolatok miatt – együtt célszerű megoldani.

4.1.2 A termelés-menedzsment követelményei

A vázolt erőforrás-korlátos, többrétegű ütemezési feladat megoldásának a lényege tehát az, hogy a mindenkori aktuális bemenő adatok alapján olyan ütemtervet kell készíteni, ami az alkalmazott gyártástechnológiából és az erőforrások rendelkezésre állásából eredő szigorú korlátozások betartása mellett a termelés-menedzsment által támasztott termelési kritériumok tekintetében a lehető legjobb számszerű teljesítménymutatókat eredményezi.

A bemutatott termelési folyamat finomprogramozási (részletes ütemezési) feladatának megoldásával kapcsolatban a legfontosabb követelmények (menedzser elvárások) a következők:

- A gyártási rendelések teljesítése az egyedileg definiált határidők túllépése (csúszás) nélkül valósuljon meg.
- A gyártás a lehető legkevesebb átállással (konfiguráció-cserével) valósuljon meg.
- A gyártás a lehető legkevesebb előkészítő tevékenységgel (konfiguráció-összeszereléssel) valósuljon meg.
- El kell kerülni a túl nagy készletek felhalmozását.
- Az ütemezési időhorizont végén biztosítani kell a terméktípusonként (cikkszámokként) külön-külön előírt minimális (tartalék) készlet szintet.
- A gyártósorok (pályák) kihasználtságát maximalizálni kell.

A megfogalmazott elvárások egymástól nem függetlenek és azok egymásnak részben ellentmondanak (pl. a készlet szintek minimalizálása és a gyártósorok kihasználtságának maximalizálása). A megfogalmazott feladat összetett sajátosságai miatt nem sorolható be közvetlenül egyik ismert ütemezési feladatkategóriába sem. Mondhatjuk azt is, hogy a termelésütemezés és a készletgazdálkodás egy kombinált új feladatát kell modellezni és megoldani. A továbbiakban ennek a feladattípusnak a jelölésére az IPSIC (Integrated Production Scheduling and Inventory Control) rövidítést használom.

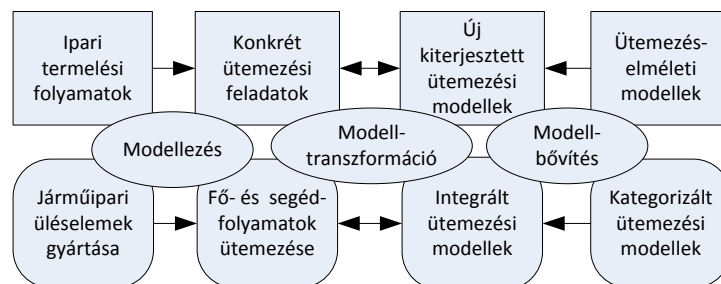
A vizsgált vállalatnál a bemutatott problémát a kutatás megkezdésekor emberi döntéshozatallal kezelték. Termelés-tervező mérnökök hozták meg a szükséges döntéseket és készítették a termelési tervet több fázisban. A tervezés időhorizontja rendszerint egy hetes időintervallum volt, menet közben azonban viszonylag sokszor kellett korrigálni az előzetes tervet. Ez nagymértékben nehezítette a folyamatok irányítását. A termelési terv megvalósításához szükséges konfigurációk előkészítésének ütemezését külön feladatként kezelték az előzetes termelési terv ismeretében. A feladatok „kézi” megoldása alapvetően MS Excel környezetben történt, melyhez az aktuális rendelések adatait ERP rendszerből kérdezték

le. A termelésstervezési hatékonyság és átláthatóság növelése érdekében a cég többször próbálkozott szoftveres megoldás bevezetésével, azonban a feladattípus sajátosságai miatt korábban nem találtak hatékonyan alkalmazható szoftvert. A kereskedelmi forgalomban elérhető ütemező alkalmazások konfigurálásával nem sikerült a feladatot megoldani. Ezek után döntött úgy a cég vezetése, hogy kutatás-fejlesztési projekt keretében új modell és megoldás létrehozását támogatja.

4.2 Az IPSIC feladat modellezése

4.2.1 A modellezés szerepe

A kutatás alapvető célja az volt, hogy kidolgozzam az IPSIC probléma részletes modelljét és annak megoldási algoritmusait annak érdekében, hogy ezek alapján implementálható legyen egy olyan termelésprogramozó szoftver, amelyik képes részletes termelési finomprogramot készíteni a mindenkori rendelések teljesítésére, valamint meg tudja valósítani a gyártási főfolyamatok és az előkészítő tevékenységek összehangolását a szigorú erőforrás-korlátos gyártórendszerben.



24. ábra: A modellezés szerepe az ipari feladatok megoldásában.

Az IPSIC feladat megoldása érdekében két irányból közelítem meg a problémát (24. ábra). Az egyik irány a vizsgált ipari folyamat megismerésével kezdődik. A folyamatok vizsgálata során alapvetően az ütemezési alapegységek, a műveletek, a technológiai lépések/fázisok/útvonalak, a végrehajtási jellemzők, az erőforrások és képességeik, a lehetséges alternatívák és korlátozások jellemzőit kell összegyűjteni és rendszerezni. Ezek után kerülhet sor a konkrét ütemezési feladatok modellezésére. Definiálni kell az erőforrás-környezetet, amely az erőforrásokat, az operációkat és azok kapcsolatait foglalja magába. Pontosán meg kell határozni az ütemezési alapegység fogalmát, valamint a végrehajtási jellemzőit és korlátozásait. Ezek után a termelés-menedzsment és a gyártásirányítás elvárásait és követelményeit ütemezési célok formájában kell kifejezni a hozzájuk tartozó fontossági mutatókkal (prioritásokkal) együtt. Mindezek együtt definiálják a konkrét ütemezési feladat részletes modelljét.

A másik irány a szakirodalomban elérhető operáció-kutatási és ütemezés-elméleti modellek vizsgálatával és azok alkalmazási feltételeinek elemzésével kezdődik. A szakirodalomban nagyon sokféle ütemezési modell található. Tematikusan szerkesztett könyvek (pl. [10], [38], [39]) és különböző témakörökre koncentrááló összefoglaló szócikkek (pl. [1], [4], [30]) segítik a tájékozódást. Erőforrás-környezeti nézőpontból a vizsgált ipari IPSIC feladathoz a

párhuzamos erőforrások modelljei (Parallel Machines/Resources) állnak a legközelebb. Ennek a témakörnek a részleteit jól összefoglalják pl. a [12], [21], és [36] cikkek. Az optimalizálási célok szempontjából a modellek döntő többsége egyetlen optimalizálási célfüggvénnyel foglalkozik. A direkt megrendelésre gyártás esetében főként valamilyen határidő-orientált teljesítménymutató kerül a középpontba. Az ütemezési modellek csak egy szűkebb halmazára jellemző a gyakorlati igények szempontjából fontos több-kritériumos (multi-objective) szemléletet (pl.: [7], [34], [30], [41], [24]).

Az operáció-kutatási és matematikai szemlélettel megalkotott elméleti ütemezési modellekben a konkrét gépek, a dolgozók és egyéb erőforrások műveletvégző képességei, időben változó rendelkezésre állásai, a valós folyamatokhoz kapcsolódó egyéb folyamatok (pl. logisztikai műveletek, gyártástechnológiai előkészítő tevékenységek stb.) gyakran túlzottan le vannak egyszerűsítve, vagy teljesen figyelmen kívül vannak hagyva. Ezek a modellek kiindulási alapként felhasználhatók, de a konkrét ipari problémák igényei miatt jelentős továbbfejlesztésre és kiterjesztésre van szükség.

A kiterjesztett modellek és a valós problémák összekapcsolásának egyik lehetséges módszere a problémater transzformációjára alapozott szemantikai megfeleltetés. Ennek lényege, hogy a valós feladat tényleges döntési változóinak halmazát (vagy annak egy rész-halmazát) a kapcsolódó korlátozásokkal együtt egy célszerűen megkonstruált (pl. szimulációs) eljárás visszavezeti egyszerűbben kezelhető döntési változók halmazára és azok korlátozásaira. Ezt a transzformált problémát a hozzá pontosan illeszkedő kiterjesztett modell oldja meg és az így kapott eredményből inverz transzformációval kiadódik az eredeti feladat megoldása.

Az értekezés további részében ennek a kétirányú szemléletnek az alkalmazását mutatom be a fentebb vázolt IPSIC probléma megoldásán keresztül.

4.2.2 Adatszerkezetek és proaktív szemléletű integrált megoldási koncepció

A szakirodalomban számos szakcikk foglalkozik a korlátozottan rendelkezésre álló párhuzamos gépek ütemezési modelljeivel és módszereivel. Ezekről jó áttekintés adnak például a [17], [19], és [35] cikkek. Az általam vizsgált IPSIC probléma sajátosságaihoz pontosan illeszkedő modellt azonban nem találtam.

A vizsgált körülmények között egy üzem terméktípusonkénti termelési intenzitása csak körülményesen és időigényesen módosítható az 4.1.1. alfejezetben leírt korlátozások miatt. Így a változatos megrendelések kiszolgálása igen komoly termelésprogramozási feladatot jelent. A beérkezett konkrét megrendelések alapján a termelési finomprogram jellemzően egy-két hetes időintervallumra készíthető el előre, mert a szükséges információk hosszabb időhorizontra nem állnak rendelkezésre vagy nagyon bizonytalanok. A termelési rendszer átállítási „lomhasága” miatt szükséges a különböző terméktípusokra egyedileg kalibrált tartalék készlet szintet fenntartani a szállítókészség javítása érdekében. Ezek a készletek adják az igényelt termékmennyiségek egy részének biztonsági fedezetét azokban az átmeneti időszakokban, amikor a vevői (független) igényekből kialakuló terhelésingadozások és terhelési csúcsok miatt a rendszert több terméktípus tekintetében egyidejűleg kellene átállítani a megfelelő termelési intenzitásra, de a „konfigurációk csereszámára” vonatkozó korlátozások ezt nem teszik lehetővé. Mivel az aggregált termelésstervezési modellek csak a gyártandó mennyiségek és

időtartamok összerendelésére képesek, így a technológiából adódó speciális (szigorú) finom korlátozásokat nem tudják kezelni. Ilyenkor nélkülözhetetlen a finomprogramozási modell korrekciós képessége, amely lehetővé teszi a szükséges gyártásirányítási beavatkozások (konfiguráció-cserék) időbeli végrehajtásának megtervezését.

A vizsgált termelésprogramozási feladat megoldásakor a gyártási főfolyamatok technológiai jellemzői és korlátozásai mellett különös figyelmet kell fordítani az előkészítési segédfolyamatok (konfigurációk előkészítésének) jellemzőire és korlátozásaira is.

Nyilvánvaló, hogy a vázolt IPSIC feladat az NP-nehéz (NP-hard) feladatkategóriába tartozik, mivel már az egyik egyszerűsített speciális esete – a szakirodalomban az $I // \sum w_i T_i$ szimbolikus jelöléssel leírt feladat – is egy jól ismert NP-hard feladat. Ebben a $I // \sum w_i T_i$ feladatban csupán egyetlen gép szerepel, és ezen egyedi prioritással (súllyal) és határidővel rendelkező munkák végrehajtását kell úgy ütemezni, hogy a határidő-túllépések (csúszások) súlyozott összege minimális legyen [39]. Mint ismert, az NP-hard feladatok esetében nem lehet polinomiális futási idejű, egzakt optimumot garantáló megoldó algoritmust készíteni [10].

Az IPSIC feladat NP-nehéz tulajdonságára tekintettel a fejlesztés során heurisztikus és meta-heurisztikus keresési technikákra koncentráltam, hogy olyan megoldási módszert dolgozzak ki, amely nagyméretű (pl. a vizsgált valós ipari feladatban 900 megrendelést tartalmazó) feladatok esetében is a gyakorlat számára elfogadható futási időt biztosít. Ennek megfelelően a korábban sikeresen alkalmazott (az értekezés 3. fejezetében bemutatott) megoldási koncepcióból indultam ki.

A vizsgált ipari IPSIC feladatban szereplő és egymáshoz kapcsolódó termelésütemezési és készletgazdálkodási feladatokat egy új integrált termelésprogramozási feladat formájában fogalmazom meg. Ez az integrált feladat (a továbbiakban termelésprogramozási feladat) többcélú kombinatorikus optimalizálási problémához vezet, amelyet proaktív szemléletű meta-heurisztikus keresési módszerrel oldok meg.

Az „integráció” ebben az esetben azt jelenti, hogy a vevői megrendelések kiszolgálását (Make to Order) és a tervezett belső készletszintek elérését/betartását célzó (Make to Stock) gyártási tételeket nem választom szét, hanem együtt kezelem, továbbá a termelési fő- és segédfolyamatok időbeli végrehajtását is együtt tervezem meg.

Ezt a stratégiát a korlátozások nagyobb komplexitása teszi szükségessé. A munkaigényes gyártás-előkészítéssel járó összetett tevékenység pedig indokolja, hogy termékorientált átállásra lehetőleg minél kevesebbszer kerüljön sor.

A választott stratégia fontos jellemzője, hogy a főfolyamatra vonatkozó termelésütemezés nem munka-orientált (job pool generation) szemléletet követ. Ebből következően előre megtervezett termelési sorozatnagyság-kalkulációt sem használ. Az új megközelítésben tehát nem használom a klasszikus operációkutatási modellek „munka” (*job*) fogalmát, hanem az igények (demand rate) valamint a korlátozások célfüggvények által vezérelt összhangja határozza meg a kvázi dinamikus sorozatnagyságokat. A termelési ütemtervben nem a munkákhoz (job-okhoz) kapcsolódó operációk időbeli végrehajtását és az azokhoz szükséges erőforrások allokálását írom le, hanem az erőforrások időbeli használatát (működését) írom le az ütemezési időhorizonton és ebből adódik ki a termékek gyártásának változó intenzitású időbeli lefutása.

A gyártórendszer kiinduló állapotának leírására olyan bemeneti adatszerkezetet használok, amelyből kiolvashatók a gyártósorok pozícióiban található konfigurációk, a pozícióhoz nem

csatlakoztatott előszerelt konfigurációk, az egyéb erőforrások és azok rendelkezésre állásai, valamint az aktuális nyitó készletszintek terméktípusonkénti bontásban. A kiinduló állapot mellett további két fontos bemeneti adatszerkezetet alkalmazok, amelyek két különböző típusú igény megfogalmazására szolgálnak. Ezek segítségével a felhasználó betöltheti egyrészt az aktuális vevői rendelések listáját, másrészt előírhatja az elvárt záró készletszintek mértékét. Minden egyes vevői igény önálló gyártási rendelést generál, amely előírja az igényelt termék típusát, darabszámát, a teljesítés határidejét és prioritását. Annak érdekében, hogy a termelési ütemterv az aktuális rendelések hiánytalan teljesítésén túlmenően megfelelő záró készletszinteket eredményezzen, vagyis mindkét igénytípust kielégítse, az optimalizálási modell terméktípusonként egyedileg előírható sávban próbálja tartani a záró készletszintet. A felhasználó előírhatja a sávok alsó és felső határait és az ütemező rendszer a sávokból való „kimozdulásokat” igyekszik minimalizálni.

A kétféle igényhalmaz teljesítését célzó termelési ütemterv definiálja minden műszakban a gyártósorok minden egyes pozíciója számára az aktívan használandó konfigurációt. Minden konfiguráció – fizikai egységként – magába foglalja a dedikált formahordozót és az arra rászertelt formákat, ezen túlmenően informatikai egységként a felszerelt formákban gyártandó terméktípusokat is definiálja. Erre azért van szükség, mert bizonyos esetekben ugyanaz a forma többféle termék gyártására is használható.

A termelési ütemterv leírja a termelési rendszer egymást követő diszkrét állapotait egy állapot-idő táblázatban. Az induló állapothoz képest növekményes rendszerben egy listában megadom a beavatkozások (konfiguráció-cserék) által okozott állapotváltozásokat. Így a termelési ütemterv és a konfiguráció-cserék sorozata egymással teljesen egyenértékűvé válik az induló állapotot ismerve. Ennek köszönhetően a termelésütemezési feladat megoldási (keresési) folyamatában egyetlen elsődleges döntési változótípust használok, ez pedig a konfiguráció-csere, amely előírja, hogy:

- melyik gyártósoron,
- melyik pozícióban,
- melyik műszakban (mikor),
- milyen konfigurációt kell felhelyezni (aktívvá tenni), és azzal
- milyen típusú terméket vagy termékeket kell gyártani.

Egy lehetséges termelési ütemterv megvalósításához – az előkészítési folyamatok összetettsége miatt – szükség van egy részletesen kidolgozott konfiguráció-előkészítési ütemtervre. Ennek meg kell válaszolnia azt a kérdést, hogy a termelési ütemterv szerint használni kívánt konfigurációkat mikor kell előkészíteni (összeszerelni a formahordozót és a formákat egy egységgé), figyelembe véve a konfigurációk pályára felhelyezésének tervezett időpontjait és az összeállításához szükséges komponensek legkorábbi rendelkezésre állási időpontjait. Ezt a beépülő (alárendelt) ütemezési részfeladatot önmagában egy speciális ütemezési feladattípusként definiálom, melynek megoldására egy új kiterjesztett modellt használok. Ennek a modellnek a felépítése a klasszikus ütemezési szemléletet tükrözi. Az ütemezés alapegysége a munka (*job*), amely itt egy dedikált konfiguráció előkészítését jelenti. Minden munkához csupán egyetlen művelet (*operation*) tartozik, amely az összeszerelési tevékenységet reprezentálja. Az aktuális termelési ütemterv (a befoglaló teljes feladat egy megoldása) dinamikusan generálja a beépülő részfeladat ütemezendő munkáit és a kapcsolódó

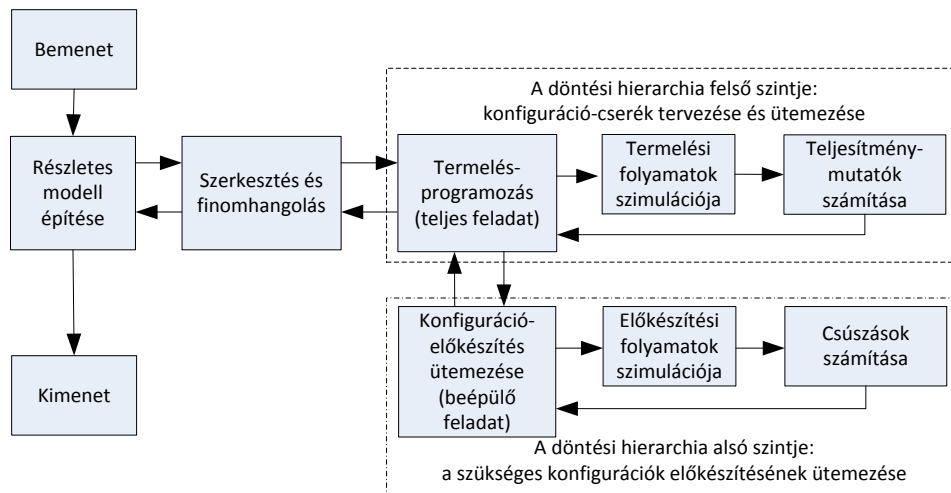
legkorábbi indítási valamint legkésőbbi befejezési időpontokra vonatkozó korlátozásokat. (Adott konfigurációk használatban vannak a pályákon valameddig, majd azok egy részét a pályáról történő levétel után át kell szerelni megadott határidőig.) Ezeket a munkákat szakképzett szerelő szakmunkások végzik el. A személyzet műszakbeosztása adott, és az elvégezhető munkák megengedett legnagyobb számát előíró korlát műszakról-műszakra változhat. Az ütemező rendszer ezeket a bemenő adatokat nem bírálhatja felül, ezért ezt a speciális korlátozást időben változó kapacitáskorláttal modellezem. Az így definiált beépülő részfeladat esetében az ütemezés célja a munkánként külön-külön előírt határidők betartása. Amennyiben ez lehetséges (a legnagyobb csúszás sem nagyobb nullánál), akkor a vizsgált befoglaló termelési ütemterv megvalósítható, máskülönben nem.

A termelési ütemterv és a konfiguráció-előkészítési ütemterv együttesét nevezem teljes termelési finomprogramnak, amely a teljes IPSIC probléma megoldását jelenti. Ennek az új integrált termelésprogramozási koncepciónak az egyszerűsített elvi vázlata látható a 25. ábrán.

A „Részletes modell építése” funkció szimbolikusan azt fejezi ki, hogy – az előrevezető irányban – létrejönnek az objektumok és az alapvető attribútumaik értéket kapnak. Majd egy automatizált részletes vizsgálat következik, amelynek során feltárulnak az összerendelhetőségi szabályok (pl. melyik gyártósor milyen terméktípusok gyártására képes, adott terméktípus milyen formákkal mely pozíciókban gyártható, melyik forma milyen formahordozóra szerelhető stb.). A visszafelé vezető irányban fordított szerepet játszik ez a modul, az elkészített megoldást alakítja át tárolható és továbbítható formátumra (ütemtervek, célfüggvény-értékek stb.). A „Szerkesztés és finomhangolás” funkció szintén kettős szerepet játszik. Egyrészt az előre vezető irányban grafikus felhasználói felületen keresztül lehetővé teszi a felhasználó számára a betöltött bemenő adatokból felépített kiindulási állapot konzisztens módosítását. Ez vonatkozhat pl. az erőforrások állapotára, a kiindulási megoldás részleteire, a célfüggvények prioritásaira stb. Ez teszi lehetővé például azt is, hogy a felhasználó zárolja vagy definiálja a finomprogram egyes részeit (pl. korlátozás formájában előírhatja, hogy egy vagy több kijelölt gyártósor kiválasztott pozícióiban adott konfigurációt vagy konfigurációkat meghatározott időintervallumokban kötelezően használni kell tesztelési céllal). Másrészt a visszafelé vezető irányban szintén ez a funkció teszi lehetővé a felhasználó számára az elkészített megoldás részleteinek vizsgálatát és konzisztens szerkesztését (pl. választhat az ajánlott különböző stratégiájú konfiguráció-előkészítési ütemtervek közül, szerkesztési eszközök használatával egyszerűen módosíthatja a javasolt termelési finomprogram részleteit stb.). Ezekről a funkciókról további részletek olvashatók a [s2] publikációban.

A bemutatott IPSIC probléma nagyon összetett, ezért a két egymáshoz kapcsolódó ütemezési feladatból egy kétszintű döntési hierarchiát alkotok, amelyben minden szint egy saját speciális optimalizálási modellt használ:

- A felső szinten a befoglaló feladat (konfiguráció-cserék ütemezése) nyilvánvalóan NP-hard, ezért azt meta-heurisztikus kereső algoritmussal kezelem.
- Az alsó szinten a beépülő részfeladat (konfiguráció-előkészítések ütemezése) viszont egy speciális modell-transzformációs eljárás közbeiktatásával polinomiális futási idejű egzakt algoritmussal megoldható.



25. ábra: A megoldási koncepció egyszerűsített elvi vázlata.

A felső ütemezési szinten elkészített minden esélyes megoldás bemenő adatokat, korlátozásokat és optimalizálási kritériumot (célfüggvényt) szállít az alsó szintre. Az átadott korlátozások halmazát a hierarchia alsó szintje kiegészíti a beépülő feladat saját korlátozásaival.

A befoglaló teljes ütemezési feladatot egy továbbfejlesztett, többcélú és többoperátoros tabukereső algoritlussal oldom meg. A beépülő részfeladat megoldására a felső szinten futó keresés minden egyes iterációjában szükség van, annak ellenőrzése céljából, hogy az aktuális teljes termelési finomprogram jelölt az előkészítés szempontjából megvalósítható-e. Mivel ezt minden jelölt esetében ellenőrizni kell, ezért rendkívül fontos, hogy a részprobléma feladata elfogadhatóan gyorsan (lehetőleg polinomiális futási idő alatt) megoldható legyen. Ha a beépülő részfeladat megoldása nem eredményez határidő túllépést (csúszást) az előkészítésben, akkor és csakis akkor lesz a teljes megoldást jelentő termelési finomprogram megvalósítható, egyébként nem.

A következő alfejezetekben a körvonalazott integrált termelésprogramozási koncepciót megvalósító új modellek és megoldási módszerek legfontosabb részleteit mutatom be. Először definiálom a teljes probléma formális optimalizálási modelljét. Ezt követően ismertetem a beépülő részprobléma (a konfiguráció-előkészítések ütemezésének) időben változó erőforrás-korlátos modelljét és megoldási algoritmusait. Végül bemutatom a termelésütemezési (konfiguráció-csere ütemezési) feladat megoldási módszerének részleteit.

4.2.3 Az IPSIC feladat formális optimalizálási modellje járműipari üléselemek gyártásának finomprogramozására

A vizsgált IPSIC termelésprogramozási feladat megoldása során a termékek gyártására alkalmas gyártósorok (pályák), formahordozók és formák egymáshoz rendelését és időbeli használatát kell úgy megtervezni, hogy a szigorú korlátozások betartása mellett a kitűzött célok teljesüljenek. Az IPSIC feladat megoldására kifejlesztett modellek és módszerek együttesen alapozzák meg a szoftveres megvalósítást biztosító funkciók implementálását.

A feladat formális modelljének leírásában használt alapelemeket, jelöléseket és összefüggéseket a következő csoportosításban mutatom be:

- Bemenő alapadatok.

- Alapadatokból származtatott segédváltozók.
- Elsődleges döntési változók.
- Elsődleges döntési változókból számított értékek.
- Másodlagos döntési változók.
- Korlátozások.
- Célfüggvények.

Bemenő alapadatok

- A gyártható terméktípusok: P_p ($p=0, 1, 2, \dots, N_P$), ahol a $p=0$ sorszámú terméktípus egy fiktív P_0 „üres” terméktípust jelent.
 - Az induló készlet nagysága darabszámban a P_p terméktípus esetén: I_p ($p=1, 2, \dots, N_P$).
 - Az előírt legkisebb zárókészlet (célkészlet) nagysága darabszámban a P_p terméktípus esetén: q_p ($p=1, 2, \dots, N_P$).
 - Az előírt legnagyobb zárókészlet (célkészlet) nagysága darabszámban a P_p terméktípus esetén: Q_p ($p=1, 2, \dots, N_P$).
- A megrendelések (terméklehívások): Θ_π ($\pi=1, 2, \dots, N_\Theta$).
 - A Θ_π megrendelés által igényelt terméktípus: Y_π ($\pi=1, 2, \dots, N_\Theta$), ahol $Y_\pi \in \{P_p \mid p=1, 2, \dots, N_P\}$. A P_0 típusú termék nem rendelhető.
 - A Θ_π megrendelés által igényelt darabszám: V_π ($\pi=1, 2, \dots, N_\Theta$).
 - A Θ_π megrendelés prioritása: W_π ($\pi=1, 2, \dots, N_\Theta$).
 - A Θ_π megrendelés teljesítésének határideje: D_π ($\pi=1, 2, \dots, N_\Theta$).
- Az ütemezési időhorizont engedélyezett műszakjai (a 26. ábrán látható rögzített hosszúságú időintervallumok globális sorszámmal azonosítva): s ($s=0, 1, 2, \dots, s_{max}$), ahol $s=0$ az új ütemterv szerinti gyártást közvetlenül megelőző időintervallumot jelenti.

Az ütemezési időhorizont engedélyezett műszakjai

8 órás időintervallumok (műszakok)	2017.04.03. Vasárnap			2017.04.04. Hétfő			2017.04.05. Kedd			2017.04.06. Szerda			...
	6:00	14:00	22:00	6:00	14:00	22:00	6:00	14:00	22:00	6:00	14:00	22:00	...
	1. Gyártósor: M_1 saját műszaksorszám (c)			aktív		aktív		aktív	aktív	aktív		aktív	aktív
			1		2		3	4	5		6	7	
2. Gyártósor: M_2 saját műszaksorszám (c)				aktív	aktív		aktív	aktív	aktív	aktív	aktív	aktív	
				1	2		3	4	5	6	7	8	
3. Gyártósor: M_3 saját műszaksorszám (c)				aktív	aktív		aktív	aktív		aktív	aktív		
				1	2		3	4		5	6		
Az engedélyezett műszakok globális sorszáma: s			1	2	3		4	5	6	7	8	9	...

Az M_m gyártósor saját c műszaksorszámainak és a műhely globális s műszaksorszámainak kapcsolata:

$s_{m,c} = s$ például: $m=3$ és $c=4$ bemenő értékek esetén az eredmény $s_{3,4} = 5$

$c(s,m) = c$ például: $s=5$ és $m=3$ bemenő értékek esetén az eredmény $c(5,3) = 4$

26. ábra: Illusztratív példa a gyártósorok engedélyezett műszakjaira.

- A gyártósorok (pályák): M_m ($m=1, 2, \dots, N_M$).
 - Az M_m gyártósor pozíciói: $M_{m,j}$ ($m=1, 2, \dots, N_M$); ($j=1, 2, \dots, N_{m,j}$).
 - Az M_m gyártósoron a P_p terméktípus gyártására alkalmas pozíciók halmaza:
 $M_{m,j}(p) \subseteq \{M_{m,j} / (m=1, 2, \dots, N_M); (j=1, 2, \dots, N_{m,j})\}$.
 - Az M_m gyártósoron a P_p terméktípussal együtt gyártható terméktípusok halmaza:
 $P(m, p) \subseteq \{P_p / p=0, 1, 2, \dots, N_P\}$.
 - Az M_m gyártósor engedélyezett saját aktív műszakjai (a 26. ábrán látható rögzített rendelkezésre állási időintervallumai): $s_{m,c}$ ($c=0, 1, 2, \dots, N_{m,c}$), ahol a $c=0$ sorszámú műszak az $s_{m,0}=0$ időintervallumot jelenti (26. ábra).
 - Az M_m gyártósoron a saját c sorszámú aktív műszakjában (az $s_{m,c}$ időintervallumban) elvégezhető konfiguráció-cserék megengedett legnagyobb száma:
 $\Delta_{m,c}$ ($m=1, 2, \dots, N_M$); ($c=0, 1, 2, \dots, N_{m,c}$).
 - Az M_m gyártósor által egy műszakban teljesített teljes körök (ciklusok) száma:
 Λ_m ($m=1, 2, \dots, N_M$);
- A formahordozók: H_h ($h=0, 1, 2, \dots, N_H$), ahol a H_0 egy fiktív „üres” formahordozót jelent.
 - A H_h formahordozó mérete szabványos termelési egységben (Standard Production Unit, SPU) mérve:
 B_h ($h=0, 1, 2, \dots, N_H$).
Az SPU mértékegység alapegysége megegyezik egy normál méretű termék egyetlen formájának helyigényével.
 - Az M_m gyártósoron használható formahordozók halmaza:
 - $H(m) \subseteq \{H_h / h=0, 1, 2, \dots, N_H\}$.
 - Az M_m gyártósoron a P_p terméktípus gyártásához használható formahordozók halmaza: $H(m, p) \subseteq H(m) \subseteq \{H_h / h=0, 1, 2, \dots, N_H\}$.
 - A formahordozók baloldalán gyártható terméktípusok halmaza:
 $P(L) \subseteq \{P_p / p=0, 1, 2, \dots, N_P\}$.
 - A formahordozók jobboldalán gyártható terméktípusok halmaza:
 $P(R) \subseteq \{P_p / p=0, 1, 2, \dots, N_P\}$.

- A formák (szerszámok): A_a ($a=0, 1, 2, \dots, N_A$), ahol az A_0 egy fiktív „üres” formát jelent.
 - Az A_a forma helyfoglalása a formahordozón (SPU mérete): b_a ($a=0, 1, 2, \dots, N_A$).
 - Az A_a formával egy kör (ciklus) alatt elkészíthető termék darabszáma: v_a ($a=0, 1, 2, \dots, N_A$)
 - A P_p terméktípus gyártásához használható formák halmaza: $A(p) \subseteq \{A_a / a=0, 1, 2, \dots, N_A\}$.
- Egy konfiguráció: ω , amely öt elemből álló vektor a következők szerint: $\omega := [\omega_1, \omega_2, \omega_3, \omega_4, \omega_5]$, ahol:
 - a konfiguráció keretét jelentő formahordozó: $\omega_1 \in \{H_h / h=0, 1, 2, \dots, N_H\}$
 - a baloldali formával gyártandó terméktípus: $\omega_2 \in \{P_p / p=0, 1, 2, \dots, N_P\}$
 - a formahordozó bal oldalára felszerelt forma: $\omega_3 \in \{A_a / a=0, 1, 2, \dots, N_A\}$
 - a jobboldali formával gyártandó terméktípus: $\omega_4 \in \{P_p / p=0, 1, 2, \dots, N_P\}$
 - a formahordozó jobb oldalára felszerelt forma: $\omega_5 \in \{A_a / a=0, 1, 2, \dots, N_A\}$
 A konfigurációk halmaza: Ω .
- Az induló (az indítást közvetlenül megelőző) rendszerállapot:
 - Az M_m gyártósor $M_{m,j}$ pozíciójában található aktív konfiguráció az $s_{m,0} = 0$ időintervallumban ($c = 0$ sorszámú műszakban): $x_{m,j,0} \in \Omega$ ($m=1, 2, \dots, N_M$); ($j=1, 2, \dots, N_{m,j}$).
 - Az indítást közvetlenül megelőző $s=0$ időintervallumban összeszerelten rendelkezésre álló (bevethető), de konkrét pozícióhoz nem kapcsolódó (nem aktív) konfigurációk halmaza: $\Phi_s \subseteq \Omega$ ($s=0$).
- Az s időintervallumban előkészíthető (összeszerelhető) konfigurációk megengedett legnagyobb darabszáma: Ψ_s ($s=0, 1, 2, \dots, s_{max}$).

Alapadatokból származtatott segédváltozók

- Egy adott konkrét időpont: t .
- Adott t időpontot magába foglaló s időintervallum: $s(t)$.
- Az s időintervallum befejezési időpontja: $\rho(s)$.
Például a 26. ábrán: $\rho(5) = „2017.04.05. 22:00”$.
- Az s időintervallumban rendelkezésre álló gyártósorok halmaza: $\mu_s := \{M_m / s \in \{s_{m,c} / m=1, 2, \dots, N_M; c=0, 1, 2, \dots, N_{m,c};\}\}$ (6)
Például a 26. ábrán az $s=2$ időintervallumban a 2. és a 3. gyártósor áll rendelkezésre: $\mu_2 := \{M_2, M_3\}$.
- Adott s időintervallummal egybeeső ($s=s_{m,c}$) aktív műszak c sorszáma az M_m gyártósor esetében: $c(s, m)$.
Például a 26. ábrán: $s=5$ és $m=3$ bemenő értékek esetén az eredmény $c(5,3) = 4$.
- Adott ϑ elem (formahordozó vagy forma) adott $\omega \in \Omega$ konfigurációhoz tartozását jelöli az alábbi függvény: $\varepsilon(\vartheta, \omega) := \begin{cases} 1, & \text{ha } \vartheta \in \omega \\ 0, & \text{egyébként} \end{cases}$ (7)

Az $\varepsilon(\vartheta, \omega) = 1$ érték azt jelzi, hogy a paraméterként megadott ϑ elem szerepel a vizsgált ω konfigurációban.

Elsődleges döntési változók

Az elsődleges döntési változók a termelési ütemtervet határozzák meg.

- A gyártósorok pozícióiban az engedélyezett műszakokban aktívan használandó konfigurációk:

$$x_{m,j,c} \in \Omega (m=1, 2, \dots, N_M); (j=1, 2, \dots, N_{m,j}); (c=1, 2, \dots, N_{m,c});$$

A döntési változók együtt egy állapot-idő táblázatot alkotnak, amelyet egy három dimenziós X tömbbel reprezentálok. Az X tömb három dimenziójának m, j, c indexei rendre felvehetik a gyártósorok, az azokhoz tartozó pozíciók és az azok számára engedélyezett műszakok sorszámait (azonosítóit). A három index együtt egyértelműen kijelöli az X tömb egy $x_{m,j,c}$ elemét (rekeszét), amely egy konkrét konfigurációt tartalmaz ($x_{m,j,c} \in \Omega$). Az X tömb valójában egy rejtett negyedik dimenzióval is rendelkezik, mert az $x_{m,j,c}$ konfiguráció maga is egy $\omega = [\omega_1, \omega_2, \omega_3, \omega_4, \omega_5]$ szerkezetű tömb (vektor). Ha figyelembe vesszük a konfigurációt alkotó elemeket is, akkor a következő felbontás adódik: $x_{m,j,c} = [x_{m,j,c,1}, x_{m,j,c,2}, x_{m,j,c,3}, x_{m,j,c,4}, x_{m,j,c,5}]$.

Elsődleges döntési változókból számított értékek

Determinisztikus működést feltételezve, a bemenő alapadatok és az elsődleges döntési változók beállított aktuális értékei egyértelműen meghatározzák a termelési rendszer viselkedését és ezáltal ismertté válnak a következő jellemzők:

- A Θ_π megrendelés teljesítésének tényleges időpontja: $Z_\pi (\pi=1, 2, \dots, N_\Theta)$.
- A Θ_π megrendelés teljesítésének határidő-túllépését jelöli a következő függvény:

$$U_\pi := \begin{cases} 1, & \text{ha } Z_\pi > D_\pi \\ 0, & \text{egyébként} \end{cases} \quad (8)$$

Az $U_\pi = 1$ érték jelzi, hogy Θ_π megrendelés teljesítésének Z_π tényleges időpontja meghaladja az előírt D_π határidőt.

- A P_p terméktípusra vonatkozó valamely rendelés teljesítésének csúszását jelző függvény:

$$\zeta_p := \begin{cases} 1, & \text{ha } \sum_{Y_\pi == P_p} U_\pi > 0 \\ 0, & \text{egyébként} \end{cases} \quad (9)$$

A $\zeta_p = 1$ érték jelzi, hogy a P_p terméktípusra vonatkozó megrendelések nem mindegyike teljesül határidőre.

- A P_p terméktípus aktuális készlet szintje darabszámban mérve a t időpontban:

$$\sigma(p, t) := \Gamma_p + \sum_{m=1}^{N_M} \left(\Lambda_m \cdot \sum_{j=1}^{N_{m,j}} \sum_{s=1}^{s(t)-1} \sum_{A_a \in A(p) \cap x_{m,j,c}(s,m)} v_a \right) - \sum_{Y_\pi == P_p \wedge Z_\pi \leq t} V_\pi \quad (10)$$

A $\sigma(p, t)$ érték kiszámításának magyarázata az, hogy a P_p terméktípus induló készlethez hozzá kell adni a t időpontig legyártott P_p típusú termékek darabszámát és le kell vonni a P_p terméktípusra vonatkozó aktivált rendelések teljesítésére kiszállított termékek darabszámait.

Az ütemezési időszak végén – a $\rho(s_{max})$ időpontban – a „határidőt túllépő és még nem teljesített” rendelések által igényelt termékmennyiségek is levonásra kerülnek az aktuális készletből, így negatív értékű zárókészlet (aktivált igény) is „kialakulhat”.

- A P_p terméktípus magas zárókészletét jelző függvény:

$$\xi_p := \begin{cases} 1, & \text{ha } \sigma(p, \rho(s_{max})) > Q_p \\ 0, & \text{egyébként} \end{cases} \quad (11)$$

A $\xi_p = 1$ érték jelzi, hogy a P_p terméktípus esetén a $\sigma(p, \rho(s_{max}))$ várható zárókészlet meghaladja a Q_p előírt legnagyobb zárókészlet (célkészlet) nagyságát.

- Konfiguráció-cserét jelző függvény:

$$\delta(x_{m,j,c}, x_{m^*,j^*,c^*}) := \begin{cases} 0, & \text{ha } x_{m,j,c} == x_{m^*,j^*,c^*} \\ 1, & \text{egyébként} \end{cases} \quad (12)$$

A δ függvény kifejezi, hogy az aktuális paraméterekkel megadott két elsődleges döntési változó értéke megegyezik-e vagy sem. Ennek megfelelő paraméterezésével könnyen lekérdezhető például, hogy adott M_m gyártósor adott $M_{m,j}$ pozícióijában adott c sorszámú engedélyezett műszakban az $x_{m,j,c}$ aktívan használandó konfiguráció megegyezik-e a következő ($c+1$ sorszámú) engedélyezett műszakban használandó $x_{m,j,c+1}$ konfigurációval. A $\delta = 1$ visszatérési érték jelzi, hogy a kérdéses műszak elején konfigurációt kell cserélni az $M_{m,j}$ pozícióban.

- A $\tau(\mathcal{G}, s'')$ függvény visszatérési értéke fejezi ki azt a legkorábbi időintervallumot, amelytől kezdve a \mathcal{G} elem folyamatosan szabad (nem aktív) egy adott s'' időintervallumig:

$$\tau(\mathcal{G}, s'') := \min_{s'=0}^{s''} \left\{ s' \mid \mathcal{G} \notin x_{m,j,c(\kappa,m)} \left(\begin{array}{l} \kappa = s', s' + 1, \dots, s''; \\ m = 1, 2, \dots, N_M; \\ j = 1, 2, \dots, N_{m,j}; \end{array} \right) \right\} \quad (13)$$

Az s' és a κ futóindexek a vizsgált s időintervallumok dupla ciklussal történő bejárását szimbolizálják. Az s' index a külső, a κ index a belső ciklust vezérli.

- Az $s=0$ kiindulási időintervallumnál későbbi adott s időintervallumban összeszerelten rendelkezésre álló (bevethető), de konkrét pozícióhoz nem kapcsolódó (nem aktív) konfigurációk időben dinamikusan változó halmazát fejezi ki az alábbi jelölés:

$$\Phi_s \subseteq \Omega(s=1, 2, \dots, s_{max}).$$

- Az elsődleges döntési változók értékeinek változása (konfiguráció-csere) valós konfiguráció-előkészítési munkát generál, ha az igényelt konfiguráció nem áll rendelkezésre:

$$\begin{aligned} & ((\delta(x_{m,j,c}, x_{m,j,c-1}) == 1) \wedge (x_{m,j,c} \notin \Phi_{s_{m,c-1}}) \wedge (x_{m,j,c,1} \neq H_0)) \Rightarrow \\ & J_i := x_{m,j,c}; d_i := s_{m,c} - 1; r_i := \max_{\mathcal{G} \in J_i} (\tau(\mathcal{G}, d_i)); N_j := |J_i| \\ & (m=1, 2, \dots, N_M); (j=1, 2, \dots, N_{m,j}); (c=1, 2, \dots, N_{m,c}); \end{aligned} \quad (14)$$

A formális leírás jelentése a következő:

Ha az $x_{m,j,c} \in \Omega$ elsődleges döntési változó aktuális értékei megváltoznak adott pozíció esetében, akkor ez a változás előírja az M_m gyártósor $M_{m,j}$ pozíciójában addig aktívan használt konfiguráció levételét a c sorszámú műszak elején. Ha az igényelt felhelyezendő konfiguráció valódi (nem marad üresen a pozíció) és az alapból a megadott időben nem fog rendelkezésre állni, akkor egy új konfiguráció-előkészítési munka (összeszerelési igény) jelenik meg a rendszerben. Ezt a generált munkát csak akkor lehet elkezdni, ha minden szükséges elem kiszerezhető a korábbi befoglaló konfigurációjából. A munkát be kell fejezni a konfiguráció pozícióba felhelyezését megelőző műszak végéig.

- A konfiguráció-előkészítési munkák (összeszerelések):

$$J_i \in \Omega (i=1, 2, \dots, N_J).$$

- A J_i munka elvégzésének legkorábbi megengedett időintervalluma:

$$r_i \in \{s \mid s=0, 1, 2, \dots, s_{max}\} (i=1, 2, \dots, N_J).$$

- A J_i munka elvégzésének legkésőbbi megengedett időintervalluma:

$$d_i \in \{s \mid s \geq r_i\} (i=1, 2, \dots, N_J).$$

- A J_i munka műveleti ideje intervallumok számában mérve:

$$p_i := 1 (i=1, 2, \dots, N_J).$$

Az egységnyi műveleti idő azt fejezi ki, hogy minden egyes munka valamely műszak elején kezdődik és ugyanabban a műszakban be is fejeződik. Ez a megközelítés teremti meg az összhangot a bemenő adatok között szereplő Ψ_s ($s=0, 1, 2, \dots, s_{max}$) értékekkel, amelyek az s időintervallumban összeszerelhető konfigurációk megengedett legnagyobb darabszámát írják elő.

Másodlagos döntési változók

A másodlagos döntési változók a konfiguráció-előkészítési ütemtervet határozzák meg.

- A szükséges $\{J_i \mid i=1, 2, \dots, N_J\}$ konfiguráció-előkészítési munkák végrehajtására kijelölt időintervallumok:

$$C_i \in \{s \mid s=0, 1, 2, \dots, s_{max}\} (i=1, 2, \dots, N_J)$$

Mivel a műveleti idő definíció szerint $p_i := 1 (i=1, 2, \dots, N_J)$, így a C_i változó által felvett valamely s műszaksorszám egyértelműen meghatározza, hogy a J_i konfiguráció-előkészítési munkát melyik időintervallumban kell elvégezni.

Ezek alapján, adott J_i konfiguráció-előkészítési munka és adott s időintervallum egymáshoz rendelése lekérdezhető a C_i másodlagos döntési változó értéke alapján az alábbi függvénnyel:

$$\lambda(J_i, s) := \begin{cases} 1, & \text{ha } C_i == s \\ 0, & \text{egyébként} \end{cases} \quad (15)$$

Korlátozások

Az aktív konfigurációkra vonatkozó korlátozások:

A termelési ütemtervben szereplő minden egyes $x_{m,j,c}$ ($m=1, 2, \dots, N_M$); ($j=1, 2, \dots, N_{m,j}$); ($c=0, 1, 2, \dots, N_{m,c}$); konfiguráció esetében be kell tartani a kiválasztott alkotóelemekre és a kijelölt pozícióra vonatkozó korlátozásokat.

Felhasználva a konfiguráció korábban bevezetett $\omega := [\omega_1, \omega_2, \omega_3, \omega_4, \omega_5]$ alakú definícióját, most jelölje az éppen vizsgált aktív konfiguráció összetételét a következő formalizmus:

$$x_{m,j,c} = [x_{m,j,c,1}, x_{m,j,c,2}, x_{m,j,c,3}, x_{m,j,c,4}, x_{m,j,c,5}] = [H_{h^*}, P_{l^*}, A_{u^*}, P_{r^*}, A_{v^*}] \quad (16)$$

Így a vizsgált konfigurációra vonatkozó korlátozások típusai a következőképpen formalizálhatók:

1. A formahordozó legyen kompatibilis a gyártóssal és a terméktípusokkal:

$$H_{h^*} \in H(m, l^*) \cap H(m, r^*) \quad (17)$$

2. A baloldali terméktípus legyen kompatibilis a gyártóssal és a jobboldali terméktípussal:

$$P_{l^*} \in P(m, r^*) \cap P(L) \quad (18)$$

3. A jobboldali terméktípus legyen kompatibilis a gyártóssal és a baloldali terméktípussal:

$$P_{r^*} \in P(m, l^*) \cap P(R) \quad (19)$$

4. A baloldalra szerelt forma és a gyártandó terméktípus legyen kompatibilis egymással:

$$A_{u^*} \in A(l^*) \quad (20)$$

5. A jobboldalra szerelt forma és a gyártandó terméktípus legyen kompatibilis egymással:

$$A_{v^*} \in A(r^*) \quad (21)$$

6. A baloldali és a jobboldali forma nem lehet ugyanaz a forma:

$$A_{u^*} \neq A_{v^*} \quad (22)$$

7. A kijelölt pozíció legyen alkalmas a baloldali és a jobboldali terméktípus gyártására:

$$M_{m,j} \in M_{m,j}(l^*) \cap M_{m,j}(r^*) \quad (23)$$

8. A formák férjenek el a formahordozón:

$$b_{u^*} + b_{v^*} \leq B_{h^*} \quad (24)$$

Az 1.-8. pontokban felsorolt korlátozások minden egyes aktív konfigurációra vonatkoznak: $x_{m,j,c}$ ($m=1, 2, \dots, N_M$); ($j=1, 2, \dots, N_{m,j}$); ($c=0, 1, 2, \dots, N_{m,c}$).

A konfiguráció-cserék darabszámára vonatkozó korlátozások:

9. A tervezett konfiguráció-cserék darabszáma nem haladhatja meg a megengedett legnagyobb értéket gyártósonkénti és azon belül műszakonkénti bontásban:

$$\sum_{j=1}^{N_{m,j}} \delta(x_{m,j,c}, x_{m,j,c-1}) \leq \Delta_{m,c} \quad (m=1, 2, \dots, N_M); \quad (c=1, 2, \dots, N_{m,c}); \quad (25)$$

A konfiguráció-elemek használatára vonatkozó korlátozások:

10. Egy forma egyszerre csak egyetlen konfigurációhoz tartozhat:

$$\sum_{M_m \in \mu_s} \sum_{j=1}^{N_{m,j}} \varepsilon(A_a, x_{m,j,c(s,m)}) + \sum_{\omega \in \Phi_s} \varepsilon(A_a, \omega) \leq 1$$

$$(a=1, 2, \dots, N_A); (s=0, 1, 2, \dots, s_{max}); \quad (26)$$

11. Egy formahordozó egyszerre csak egyetlen konfigurációhoz tartozhat:

$$\sum_{M_m \in \mu_s} \sum_{j=1}^{N_{m,j}} \varepsilon(H_h, x_{m,j,c(s,m)}) + \sum_{\omega \in \Phi_s} \varepsilon(H_h, \omega) \leq 1$$

$$(h=1, 2, \dots, N_H); (s=0, 1, 2, \dots, s_{max}); \quad (27)$$

A konfiguráció-előkészítési munkák ütemezésére vonatkozó korlátozások:

12. Az s aktív időintervallumhoz hozzárendelt munkák darabszáma nem haladhatja meg az s időintervallumra vonatkozó kapacitáskorlátot:

$$\sum_{i=1}^{N_J} \lambda(J_i, s) \leq \Psi_s \quad (s=0, 1, 2, \dots, s_{max}); \quad (28)$$

(A kapacitáskorlát alatt az elvégezhető munkák darabszáma értendő.)

13. Egy előkészítő munka (job) csak egyetlen időintervallumhoz lehet hozzárendelve:

$$\sum_{s=0}^{s_{max}} \lambda(J_i, s) = 1 \quad (i=1, 2, \dots, N_J); \quad (29)$$

14. A J_i munka nem ütemezhető korábban, mint amikor a szükséges elemek mindegyike már rendelkezésre áll:

$$r_i \leq C_i \quad (i=1, 2, \dots, N_J); \quad (30)$$

15. A J_i munkát be kell fejezni (a konfigurációt készre kell szerelni) az előírt határidőre:

$$C_i \leq d_i \quad (i=1, 2, \dots, N_J); \quad (31)$$

Célfüggvények

Az optimalizálási modellt úgy definiálom, hogy egyszerre több célfüggvény figyelembe vételével keresem a legjobb megoldást. A célfüggvények minimalizálandó alakban megadva a következők:

1. A rendelések teljesítésének időpontjában fellépő legnagyobb termékhiány [db]:

$$f_1 := \max_{\pi=1}^{N_{\theta}} \left(\max \left(0, V_{\pi} - \sigma(Y_{\pi}, \rho(s(D_{\pi}) - 1)) \right) \right) \rightarrow \min. \quad (32)$$

2. A rendelések teljesítésének időpontjában fellépő termékhiányok összege [db]:

$$f_2 := \sum_{\pi=1}^{N_{\theta}} \left(\max \left(0, V_{\pi} - \sigma(Y_{\pi}, \rho(s(D_{\pi}) - 1)) \right) \right) \rightarrow \min. \quad (33)$$

3. A csúszó (határidőt túllépő) rendelés-teljesítések száma:

$$f_3 := \sum_{\pi=1}^{N_{\theta}} U_{\pi} \rightarrow \min. \quad (34)$$

4. A konfiguráció-cserék száma:

$$f_4 := \sum_{m=1}^{N_M} \sum_{j=1}^{N_{m,j}} \sum_{c=1}^{N_{m,c}} \delta(x_{m,j,c}, x_{m,j,c-1}) \rightarrow \min. \quad (35)$$

5. A konfiguráció-cserék legnagyobb száma egy műszakban:

$$f_5 := \max_{m=1}^{N_M} \left(\max_{c=1}^{N_{m,c}} \left(\sum_{j=1}^{N_{m,j}} \delta(x_{m,j,c}, x_{m,j,c-1}) \right) \right) \rightarrow \min. \quad (36)$$

6. A többlettel záró terméktípusok száma:

$$f_6 := \sum_{p=1}^{N_P} \xi_p \rightarrow \min. \quad (37)$$

7. A terméktöbbletek összege az ütemezési időszak végén [db]:

$$f_7 := \sum_{p=1}^{N_P} \left(\max(0, \sigma(p, \rho(s_{max})) - Q_p) \right) \rightarrow \min. \quad (38)$$

8. A legnagyobb termékhiány az ütemezési időszak végén [db]:

$$f_8 := \max_{p=1}^{N_P} \left(\max(0, q_p - \sigma(p, \rho(s_{max})) \right) \rightarrow \min. \quad (39)$$

9. A termékhiány összege az ütemezési időszak végén [db]:

$$f_9 := \sum_{p=1}^{N_P} \left(\max(0, q_p - \sigma(p, \rho(s_{max})) \right) \rightarrow \min. \quad (40)$$

10. A határidőre nem teljesített rendelések prioritásainak összege:

$$f_{10} := \sum_{\pi=1}^{N_{\theta}} W_{\pi} \cdot U_{\pi} \rightarrow \min. \quad (41)$$

11. A határidőre nem teljesített rendelések legnagyobb prioritása:

$$f_{11} := \max_{\pi=1}^{N_{\theta}} (W_{\pi} \cdot U_{\pi}) \rightarrow \min. \quad (42)$$

12. A csúszó rendelés-teljesítések által érintett terméktípusok száma:

$$f_{12} := \sum_{p=1}^{N_P} \zeta_p \rightarrow \min. \quad (43)$$

13. A legnagyobb termékhiány (nullához képest) az ütemezési időszak végén [db]:

$$f_{13} := \max_{p=1}^{N_P} \left(\max(0, 0 - \sigma(p, \rho(s_{max})) \right) \rightarrow \min. \quad (44)$$

14. A termékhiány összege (nullához képest) az ütemezési időszak végén [db]:

$$f_{14} := \sum_{p=1}^{N_P} \left(\max(0, 0 - \sigma(p, \rho(s_{max})) \right) \rightarrow \min. \quad (45)$$

15. A rendelés-teljesítések legnagyobb csúszása (határidő túllépése) [műszak]:

$$f_{15} := \max_{\pi=1}^{N_{\theta}} \left(\max(0, s(Z_{\pi}) - (s(D_{\pi}) - 1)) \right) \rightarrow \min. \quad (46)$$

16. A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak]:

$$f_{16} := \sum_{\pi=1}^{N_{\theta}} \left(\max(0, s(Z_{\pi}) - (s(D_{\pi}) - 1)) \right) \rightarrow \min. \quad (47)$$

17. A konfiguráció-előkészítési munkák (job-ok) száma:

$$f_{17} := N_j \rightarrow \min. \quad (48)$$

18. Nem használt pályakapacitás [SPU]:

$$f_{18} := \sum_{m=1}^{N_M} \left(N_{m,j} \cdot N_{m,c} \cdot \max_{H_h \in H(m)} (B_h) - \sum_{j=1}^{N_{m,j}} \sum_{c=1}^{N_{m,c}} \sum_{A_a \in x_{m,j,c}} b_a \right) \rightarrow \min. \quad (49)$$

Az $f_{18} \rightarrow \min$ a gyártósorok kihasználatlanságának minimalizálását jelenti, ami megegyezik a gyártósorok kihasználtságának maximalizálásával. Ez a formalizált célfüggvény azt a törekvést fogalmazza meg, hogy a műszakokban aktívan használt formák összegzett valós termékgyártó képessége a teljes ütemezési időhorizonton a lehető legnagyobb mértékben közelítse meg az elméletben számítható legnagyobb értéket. A termékgyártó képesség mérőszáma itt azt fejezi ki, hogy a műszakokban aktívan használt formák összterfoglata hányszorosa az alapegységnek (az SPU-nak). A $f_{18} \rightarrow \min$ célfüggvény tehát azt fejezi ki, hogy a gyártósorok a lehető legnagyobb mértékben fel legyenek szerszámozva aktív formákkal.

A felsorolt célfüggvény-definíciókkal a termelés-menedzsment elvárásainak összességét modelleztem matematikai formalizmussal. A célfüggvények tervezésekor négy cél-kategóriát különböztettem meg, ezek a következők:

- I. A rendelkezések teljesítése az előírt határidőkre:
 $f_1, f_2, f_3, f_{10}, f_{11}, f_{12}, f_{15}, f_{16}$
- II. A beavatkozások és átállások mértékének minimalizálása:
 f_4, f_5, f_{17}
- III. Az előírt készletszintek fenntartása:
 $f_6, f_7, f_8, f_9, f_{13}, f_{14}$
- IV. A gyártósorok kapacitáskihasználtságának maximalizálása (a kihasználatlanság minimalizálása):
 f_{18}

A célfüggvények sorszáma a rendszerbe beépítés sorrendjét fejezi ki. A valós feladatokon végzett tesztek futási eredményei alapján fokozatosan bővítettem a célfüggvény-rendszert. Ezek a célfüggvények szükségesek és egyben elégségesek is a termelés-menedzsment követelményeinek teljesítéséhez. A fenti kategorizálás az elsődleges szerep szerint történt. Vannak azonban olyan célfüggvények is, amelyek egyszerre több cél-kategóriát támogatnak. Ilyen például az f_{13} és az f_{14} . Ezek a célfüggvények alapvetően a záró készletszintek vizsgálata alapján próbálják az abszolút termékhiányt minimalizálni az ütemezési időhorizont végén. Ugyanakkor ez a törekvés maga után vonja azt is, hogy az összes rendelést teljesíteni kell legkésőbb az időszak végéig, ellenkező esetben ugyanis negatív értékű zárókészlet alakulna ki. Így az f_{13} és az f_{14} célfüggvény az I. és a III. cél-kategóriát egyszerre erősíti.

Az IPSIC feladat megoldása érdekében a bemeneti alapadatok, a döntési változók, a korlátozások és a célfüggvények formális leírásával egy új többcélú optimalizálási modellt definiáltam.

A következő alfejezetekben bemutatom a definiált többcélú optimalizálási feladat általam kidolgozott megoldási módszerét. A koncepció leírásakor fentebb vázolt kétszintű döntési hierarchiára alapozva az új modellek és módszerek bemutatását két részre bontom. Az első részben a konfiguráció-előkészítési tevékenységek ütemezési modelljét és algoritmusait fejtem ki részletesen. Ezeket felhasználva a második részben a termelés főfolyamatainak ütemezését mutatom be.

4.3 A konfiguráció-előkészítési tevékenységek ütemezése

4.3.1 A konfiguráció-előkészítések ütemezési problémája

Az IPSIC feladat elsődleges – a főfolyamatra vonatkozó – döntési változóinak értékét egy többoperátoros és többcélú kereső algoritmus állítja be. A kifejlesztett megoldó algoritmus iteratívan módosítja az aktuális termelési ütemtervet, ezáltal az iterációkban konzisztens változtatásokkal új megoldásokat készít. Ennek a keresési folyamatnak minden egyes közbenső lépése által előállított megoldásról meg kell állapítani, hogy az konfiguráció-előkészítés szempontjából megvalósítható-e vagy sem. Ez azt jelenti, hogy az aktuálisan vizsgált termelési ütemtervet alapul véve készíteni kell egy megvalósítható konfiguráció-előkészítési ütemtervet, amely a korlátozásoknak megfelel. Ha ez lehetséges, akkor a termelési ütemterv megvalósítható, ellenkező esetben nem.

Ezt az eldöntendő kérdést rövid idő alatt kell megválaszolni, mivel ez a keresési folyamatban nagyon sokszor (minden egyes új jelölt esetében) felmerül. Ugyanakkor rendkívül fontos, hogy a beépülő feladat modellje a gyakorlati alkalmazhatóság szempontjából kellően pontos és részletes legyen.

A termelésütemezési feladatba beépülő konfiguráció-előkészítési részproblémát önmagában egy speciális ütemezési feladat formájában definiálok. Ez a feladat a következőképpen foglalható össze:

- Ismert számú egymástól független munkát kell elvégezni. Minden egyes munkának egyedileg definiált legkorábbi engedélyezett indítási időpontja és határideje van.
- Minden egyes munkához egyetlen operáció tartozik, ezek műveleti ideje adott. A műveletek nem szakíthatók meg.
- Az erőforrások rendelkezésre állása nem folyamatos. Adott időintervallumokban (műszakokban) végezhetők el a műveletek. A műszakok rögzített hosszúságúak (időtartamúak) és diszjunktak (nem lapolódhatnak át).
- A munkákat szerelő szakmunkások végzik el. Az egyes műszakokban rendelkezésre álló dolgozó szerelők száma adott. A dolgozók műveletvégzési sebessége ismert.
- Az ütemezés célja az, hogy a korlátozások betartásával minden munka sikeresen befejeződjön a saját határidejére.

Ez a probléma ebben az általános formában nagyon nehezen kezelhető. Ezért kidolgoztam egy probléma-transzformációs eljárást, melynek segítségével a problémát átalakítom egy továbbfejlesztett, párhuzamosan működő virtuális gépeket tartalmazó ütemezési feladatra, melynek speciális tulajdonsága az, hogy a rendelkezésre álló gépek száma függ az időtől.

4.3.2 A probléma transzformációja és egy új kiterjesztett ütemezési modell definiálása

Az IPSIC feladatba beépülő konfiguráció-előkészítési ütemezési probléma transzformációjának lényege a következő:

- Az erőforrások rendelkezésre állási időintervallumait (a rögzített műszakokat) a műhelyhez rendelt globális rendszerben kezdési időpont szerint növekvő sorrendben haladva besorszámozom decimális egészekkel (a korábban bemutatott 26. ábra szerint). A továbbiakban ezeket a sorszámokat lépéseknek nevezem és s szimbólummal jelölöm. A legnagyobb kiosztott sorszám az s_{max} . A kiosztásból következően a lépések összefüggő sorozatot alkotnak, így a modellben ez helyettesíti a valós időtengelyt.
- A szerelő szakmunkásokból álló csoportokat párhuzamosan működő virtuális gépekkel (erőforrásokkal) modellezem. Egy virtuális gép egyszerre csak egy munkán dolgozhat, és minden egyes munkán egyszerre csak egy virtuális gép dolgozhat. A virtuális gépek száma az s lépések függvényében (műszakról-műszakra) változhat. Ezt a változó kapacitáskorlátot szimbolikusan a $P(s)$ jelöléssel fejezem ki. Az adott s lépésben rendelkezésre álló virtuális gépek konkrét számát az érintett műszakban elvégezhető munkák számára vonatkozó eredeti korlátozás állítja be:

$$P(s) := \Psi_s (s=0, 1, 2, \dots, s_{max}). \quad (50)$$

- Az előírt konfiguráció-előkészítési tevékenységeket N_j számú egymástól független J_i ($i=1, 2, \dots, N_j$) munkának tekintem.
- A J_i munkához tartozó valós időadatokat rendre átalakítom diszkrét s lépésekre (műszaksorszámokra):
 - A konfigurációk valós előkészítési idejét a virtuális gépek korábban részletezett műveletvégző képességéhez és darabszámához igazítom. Ennek következtében minden egyes munka műveleti ideje definíció szerint egy műszak hosszának megfelelő egységnyi értéket kap: $p_i := 1$ ($i=1, 2, \dots, N_j$). Egy munka mindig egy kijelölt műszak elején kezdődik és a végén fejeződik be egy adott virtuális gépen.
 - A J_i munka legkorábbi indítási időkorlátját egy megfelelően kiválasztott műszak diszkrét lépésszámával (r_i) definiálom. Az a legkorábbi műszak jelenti a korlátot, melynek már az elején minden indítási feltétel teljesül (az adott konfigurációt alkotó komponensek rendelkezésre állnak).
 - A J_i munka legkésőbbi befejezési időkorlátját szintén egy megfelelően kiválasztott műszak diszkrét lépésszámával (d_i) definiálom. Itt az a legkésőbbi műszak jelenti a korlátot, melynek az elején még nem kerül fel az adott konfiguráció a gyártósorra.
 - A J_i munka teljesítésére kijelölt időintervallumot a hozzá tartozó diszkrét lépésszámmal fejezem ki (C_i). Ez az ütemezési feladat döntési változója. A J_i munka végrehajtása a C_i döntési változó értéke által kijelölt műszak elején kezdődik és ugyanannak a műszaknak a végén fejeződik be valamelyik virtuális gépen.
 - A J_i munka késését $L_i = C_i - d_i$; valamint csúszását $T_i = \max(0, L_i)$; szintén egységnyi lépésben mérem. Két aktív műszak lépésben mért távolsága a hozzájuk tartozó sorszámok különbségeként adódik. Ez eltér a tényleges időpontok között eltelt időtől. (Például ha hétvégén nincs aktív műszak, akkor a péntek 14:00-22:00 közötti műszak és a hétfő 6:00-14:00 közötti műszak távolsága 1 lépés. Ha hétvégén van két aktív műszak, akkor az előbb említett pénteki és hétfői műszakok távolsága már 3 lépés.)

A transzformációval előállított új ütemezési feladatot – a fenti jelölésekre alapozva és felhasználva a szakirodalomban használt $\alpha \mid \beta \mid \gamma$ formalizmust – a következőképpen definiálom:

$$P(s) \mid p_i = 1; r_i = integer; d_i = integer \mid L_{max} \quad (51)$$

Ez az új feladattípus hasonló a Gharbi és Haouari által a [17] cikkben vizsgált feladathoz abban az értelemben, hogy mindkét modellben párhuzamos erőforrások szerepelnek és a munkáknak indítási és befejezési időkorlátai is vannak. Azonban az én modellemben minden egyes gépnek (erőforrásnak) saját rendelkezésre állási intervallum-listája van, míg a Gharbi-modell csupán egyetlen időablakot rendel minden géphez. A két modellben alkalmazott célfüggvények is eltérőek.

Brucker [10] könyvében korábban bemutatta a $P \mid p_i=1; r_i=integer \mid L_{max}$ feladatot. A Brucker-féle modell nagyon hasonló az általam definiált modellhez, mert abban is egységnyi műveleti idők, egész értékű indítási és befejezési időkorlátok szerepelnek. Azonban én még azt is figyelembe veszem, hogy az erőforrások (gépek) rendelkezésre állása időben változó lehet, így az új kiterjesztett $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ feladattípus speciális esetként

magába foglalja a Brucker-féle klasszikus $P \mid p_i=1; r_i=integer \mid L_{max}$ feladatot, mert abban a gépek száma kötött és minden gép folyamatosan rendelkezésre áll.

4.3.3 Időtartalék-orientált megoldási módszer

A $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ ütemezési feladat megoldására kifejlesztettem egy új ütemezési algoritmust (27. ábra).

Időtartalék-orientált algoritmus a $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ feladathoz

```

{ Rendezzük a  $J_i$  ( $i=1, 2, \dots, N_J$ ) munkákat  $r_i$  szerint nem csökkenő sorrendbe;
  item  $\leftarrow 1$ ;
  while (item  $\leq N_J$ )
  {  $s \leftarrow r_{item}$ ;
    while ( $P(s) < 1$ )
    {
       $s \leftarrow s + 1$ ;
      if ( $s > s_{max}$ ) Kilépés megvalósítható megoldás nélkül;
    }
     $R \leftarrow \{J_i \mid J_i \text{ nem ütemezett és } r_i \leq s\}$ ;
    mach  $\leftarrow 1$ ;
    while ( $R$  nem üres)
    { Válasszuk ki a legkisebb  $d_i$  határidővel rendelkező  $J_i$  munkát az  $R$ -ből;
       $R \leftarrow R \setminus \{J_i\}$ ;
      Ütemezzük a  $J_i$ -t a mach.-edik rendelkezésre álló gépre az  $s$  műszakban;
       $C_i \leftarrow s$ ;
       $L_i \leftarrow C_i - d_i$ ;
       $T_i \leftarrow \max(0, L_i)$ ;
      item  $\leftarrow item + 1$ ;
      if (mach + 1  $\leq P(s)$ ) mach  $\leftarrow mach + 1$ ;
      else
      { mach  $\leftarrow 1$ ;
         $s \leftarrow s + 1$ ;
        while ( $P(s) < 1$ )
        {
           $s \leftarrow s + 1$ ;
          if ( $s > s_{max}$ ) Kilépés megvalósítható megoldás nélkül;
        }
         $R \leftarrow R \cup \{J_i \mid J_i \text{ nem ütemezett és } r_i \leq s\}$ 
      }
    }
  }
  Visszatérés az elkészített optimális megoldással;
}

```

27. ábra: Időtartalék-orientált ütemezési algoritmus

A megoldó algoritmus alapelve a következő:

A virtuális időtengelyen a változó előkészítési kapacitás adott. Minden egyes s lépéshez tartozó $P(s)$ kapacitáskorlát értékét tárolom egy vektorban. Induláskor rendezem a munkákat az indíthatóság r_i értéke szerint növekvő sorrendbe előkészítve a mindenkor aktuális s műszakban indítható munkák halmazának lekérdezését. Majd a legkisebb r_i értéktől kezdve az s és $P(s)$ párok mentén haladva kiválasztom az aktuális beütemezendő munkát. A kiválasztás úgy történik, hogy az aktuális s értéket vezérlő paraméternek tekintve meghatározom az akkor már indítható munkák halmazát (R), majd ebből kiválasztom a legkisebb „időtartalékkal” rendelkező munkát. Mivel minden egyes munka azonos műveleti idejű, így a legkisebb időtartaléka a legkorábbi határidejű munkának van. Az így kiválasztott munkát (J_i) a lehető legkorábbi szabad időintervallumra ütemezem, vagyis az aktuális s -re ($C_i \leftarrow s$).

Az algoritmus azokat a szituációkat is helyesen kezeli, amikor bizonyos műszakokban egyetlen munkát sem lehet elvégezni, mert ott a $P(s)$ érték rendre nulla (nincs előkészítő szerelő kapacitás). Az ilyen műszakokat az algoritmus nem terheli, hanem átlépi. Az is előfordulhat kapacitáshiány miatt, hogy nem minden aktivált munka ütemezhető be! Ezt jelzi a „*Kilépés megvalósítható megoldás nélkül;*” felirat.

Ha minden munka beütemezhető (létezik megvalósítható megoldás), akkor a bemutatott algoritmus minimális késést eredményező optimális megoldást állít elő polinomiális futási idő alatt. Ha a munkák sorszámai elve az indíthatósági időpontok növekvő sorrendjében vannak kiosztva ($r_1 \leq r_2 \leq \dots \leq r_i \leq \dots \leq r_{N_j}$), akkor az algoritmus futási idejének ordója csupán $O(n \log n)$, ahol $n=N_j$ a munkák számát jelenti.

4.3.4 Az optimális megoldás elérésének bizonyítása

Állítás:

Az időtartalék-orientált algoritmus a $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ feladat optimális megoldását állítja elő.

Bizonyítás:

A bizonyítás során bemutatom, hogy létezik olyan optimális megoldás, amelynek célfüggvény-értéke azonos az időtartalék-orientált algoritmus által előállított megoldás célfüggvény értékével.

Legyenek a munkákhoz tartozó sorszámok (az i indexek) a megengedett legkorábbi indítási időpontok növekvő sorrendjében kiosztva:

$$J_i (i=1, 2, \dots, N_j), \text{ ahol } r_1 \leq r_2 \leq \dots \leq r_i \leq \dots \leq r_{N_j}.$$

Jelölje S_a az időtartalék-orientált algoritmus által adott megoldást.

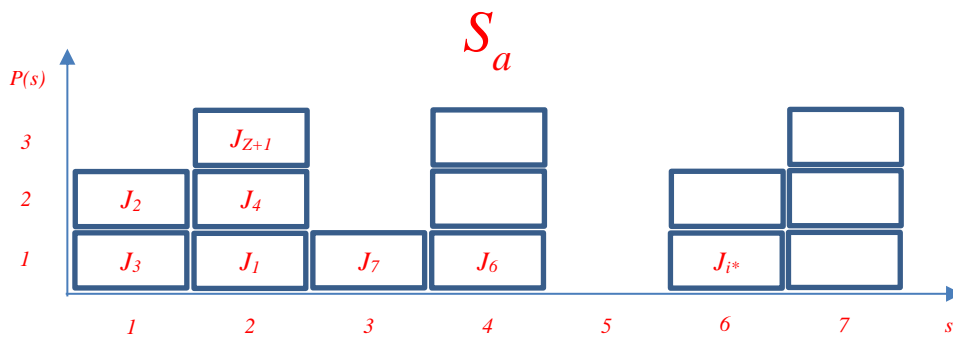
Legyen S_b egy optimális megoldás.

Az S_a és az S_b ismeretében meghatározható, hogy az i index szerinti első Z ($0 \leq Z \leq N_j$) számú munka rendre ugyanabba a műszakba van beütemezve az S_a és az S_b szerint. Így $Z+1$ jelöli azt a legkisebb i indexet, amelynél a J_i munka C_i befejezési időpontja különbözik az S_a és az S_b szerint (28. ábra).

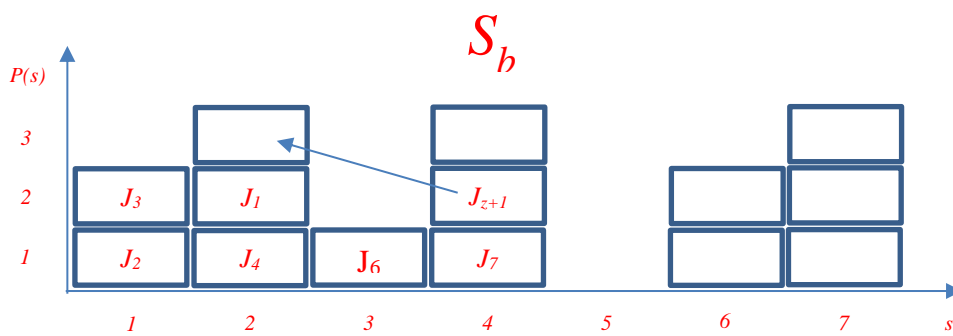
Ha $Z = N_J$, akkor minden egyes munka befejezési időpontja rendre azonos a két megoldás szerint, így a két megoldás célfüggvény-értéke azonos, továbbá mivel S_b optimális megoldás, így az S_a is optimális.

				$Z=4$	$Z+1=5$						
	i	1	2	3	4	5	6	...	i^*	...	N_J
S_a	C_i	2	1	1	2	2	4	...	6	...	
	i	1	2	3	4	5	6	...	i^*	...	N_J
S_b	C_i	2	1	1	2	4	3	...	2	...	

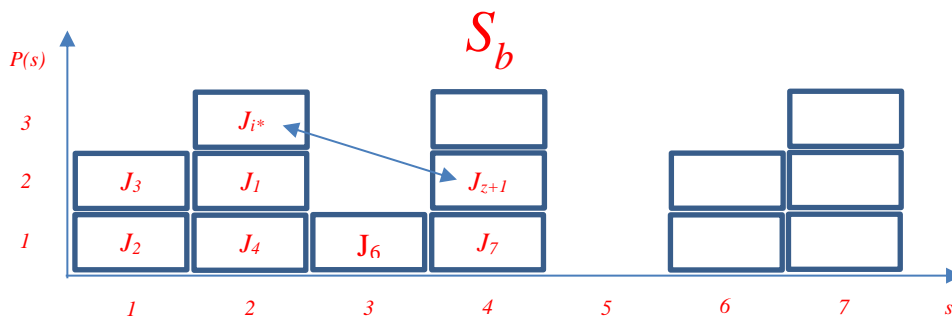
28. ábra: Két megoldás összehasonlítása a munkák befejezési időpontja alapján és a Z érték jelentése.



29. ábra: Illusztratív példa az időtartalék-orientált algoritmus eredményére.



30. ábra: Illusztratív példa az 1. esetre.



31. ábra: Illusztratív példa a 2. esetre.

Az összehasonlítás módjából és az időtartalék-orientált algoritmus működéséből következik, hogy ha a $Z < N_J$, akkor a J_{Z+1} munka a C_{Z+1} műszakba van beütemezve az S_a szerint (29. ábra), míg ugyanez a J_{Z+1} munka a C_{Z+1} műszaknál későbbi műszakba van beütemezve az S_b szerint. Ilyenkor az alábbi két eset különböztethető meg:

1. Van szabad virtuális gép a C_{Z+1} műszakban az S_b szerint (30. ábra).

Ebben az esetben a J_{Z+1} munka a szabad gépre áthelyezhető. Ez az S_b megoldáson elvégzett módosítás nem növeli meg az L_{max} célfüggvény értékét, mert a J_{Z+1} munka az eredetinel korábbi műszakba kerül, vagyis a késése nem nőhet. Tehát a módosított megoldás is optimális megoldás.

2. Nincs szabad virtuális gép a C_{Z+1} műszakban az S_b szerint (31. ábra).

Ebben az esetben létezik egy J_{i^*} munka, amelyik a C_{Z+1} műszakba van beütemezve az S_b szerint, de ez a J_{i^*} munka a C_{Z+1} műszaknál későbbi műszakba van beütemezve az S_a szerint.

Mivel az időtartalék-orientált algoritmus a J_{Z+1} munkát korábbi kezdéssel ütemezte be az S_a szerint mint a J_{i^*} munkát, így az algoritmus működéséből adódik, hogy a J_{i^*} munka d_{i^*} határideje nagyobb vagy egyenlő mint a J_{Z+1} munka d_{Z+1} határideje: $d_{i^*} \geq d_{Z+1}$. Bevezetve az $s_t \geq 0$ értéket, írható hogy:

$$d_{i^*} = d_{Z+1} + s_t. \quad (52)$$

Felhasználva a késés $L_i = C_i - d_i$ definícióját, az érintett munkák domináns késése az S_b megoldás szerint a következőképpen alakul:

$$\max(L_{i^*}, L_{Z+1}) = \max(C_{i^*} - d_{i^*}, C_{Z+1} - d_{Z+1}) = \max(C_{i^*} - d_{Z+1} - s_t, C_{Z+1} - d_{Z+1}). \quad (53)$$

Mivel $C_{Z+1} > C_{i^*}$, bevezetve az $s_T > 0$ értéket, írható hogy:

$$C_{Z+1} = C_{i^*} + s_T. \quad (54)$$

Ezt felhasználva adódik, hogy:

$$\max(L_{i^*}, L_{Z+1}) = \max(C_{i^*} - d_{Z+1} - s_t, C_{i^*} + s_T - d_{Z+1}). \quad (55)$$

Bevezetve az $L_{iZ} = C_{i^*} - d_{Z+1}$ jelölést adódik, hogy:

$$\max(L_{i^*}, L_{Z+1}) = \max(L_{iZ} - s_t, L_{iZ} + s_T) = L_{iZ} + s_T. \quad (56)$$

Az S_b megoldás J_{Z+1} és a J_{i^*} munkájának felcserélése után a két munka domináns késése a következőképpen alakul:

$$\max(L_{i^*}, L_{Z+1}) = \max(C_{Z+1} - d_{i^*}, C_{i^*} - d_{Z+1}). \quad (57)$$

Behelyettesítve a bevezetett jelöléseket adódik, hogy:

$$\begin{aligned} \max(L_{i^*}, L_{Z+1}) &= \max(C_{i^*} + s_T - d_{Z+1} - s_t, C_{i^*} - d_{Z+1}) = \max(L_{iZ} + s_T - s_t, L_{iZ}) \\ \max(L_{i^*}, L_{Z+1}) &= L_{iZ} + s_T - s_t. \end{aligned} \quad (58)$$

Mivel S_b optimális, így a csere után sem lehet kisebb a célfüggvény értéke mint az S_b esetében, ebből következik hogy:

$$L_{iZ} + s_T - s_t \geq L_{iZ} + s_T. \quad (59)$$

Mivel $s_t \geq 0$, így a J_{Z+1} és a J_{i^*} munka felcserélése az S_b megoldásban nem növeli meg a célfüggvény értékét.

Mindkét lehetséges esetben az eredeti optimális S_b megoldásból kiindulva a módosított S_b megoldás is optimális maradt, ugyanakkor most már az első $Z+1$ darab munka fejeződik be azonos műszakban az S_a és a módosított S_b szerint. A módosítás hatására a kiinduláshoz képes a Z érték növekedett eggyel.

A fenti logikát folytatva a módosítások sorozata elvezet oda, hogy a Z értéke addig nő, míg végül minden egyes J_i munka ($i=1, 2, \dots, N_J$) rendre ugyanabba a C_i műszakba lesz beütemezve a vizsgált S_a és egy optimális (többszörösen módosított S_b) megoldás szerint. Ebből az következik, hogy az S_a megoldás is optimális. Ez egyértelműen bizonyítja, hogy az időtartalék-orientált algoritmus a célfüggvény szerint optimális megoldást állítja elő.

4.3.5 A beépülő részfeladat és a teljes IPSIC feladat kapcsolata

Az IPSIC feladat elsődleges döntési változóinak értékváltozásai (a konfiguráció-cserék) generálják a valós konfiguráció-előkészítési munkákat. Ennek pontos definícióját magyarázattal együtt megadtam a 4.2.3. alfejezetben. Röviden összefoglalva:

$$\begin{aligned} & ((\delta(x_{m,j,c}, x_{m,j,c-1}) == 1) \wedge (x_{m,j,c} \notin \Phi_{s_{m,c-1}}) \wedge (x_{m,j,c,1} \neq H_0)) \Rightarrow \\ & J_i := x_{m,j,c}; d_i := s_{m,c} - 1; r_i := \max_{\vartheta \in J_i} (\tau(\vartheta, d_i)); N_J := |\{J_i\}| \\ & (m=1, 2, \dots, N_M); (j=1, 2, \dots, N_{m,j}); (c=1, 2, \dots, N_{m,c}); \end{aligned}$$

Ezek alapján kapnak aktuális értéket a beépülő feladat munkái (az előkészítő job-jai).

A $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ ütemezési feladattípus döntési változói megegyeznek az IPSIC feladat másodlagos döntési változóival:

$$C_i \in \{s \mid s=0, 1, 2, \dots, s_{max}\} (i=1, 2, \dots, N_J)$$

Az időtartalék-orientált algoritmus megoldja ezt a beépülő ütemezési feladatot azáltal, hogy beállítja minden egyes C_i döntési változó aktuális értékét. Ez mutatja, hogy a J_i munka a C_i sorszámú műszakba van beütemezve.

Az IPSIC feladat optimalizálási modelljében a konfiguráció-előkészítésekre vonatkozó korlátozások a 12.-15. sorszámmal szerepelnek. Ezek a következők:

$$\begin{aligned} \sum_{i=1}^{N_J} \lambda(J_i, s) &\leq \Psi_s (s=0, 1, 2, \dots, s_{max}); \\ \sum_{s=0}^{s_{max}} \lambda(J_i, s) &= 1 (i=1, 2, \dots, N_J); \\ r_i &\leq C_i (i=1, 2, \dots, N_J); \\ C_i &\leq d_i (i=1, 2, \dots, N_J); \end{aligned}$$

Ebből a négy korlátozás-típusból hármat meghagytam eredeti alakjában az alapján, hogy az s műszakban rendelkezésre álló gépek $P(s)$ számát az elvégezhető Ψ_s munkák számának megfelelően definiáltam:

$$P(s) := \Psi_s (s=0, 1, 2, \dots, s_{max}). \quad (60)$$

Az utolsó korlátozás-típust azonban átalakítottam célfüggvénnyé az alábbi módon:

$$L_{max} := \max_{i=1}^{N_J} (\max(0, C_i - d_i)) \rightarrow \min. \quad (61)$$

Így a $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ ütemezési feladat konkrét példányának optimális megoldására alapozva egyértelműen eldönthető az, hogy az a termelési ütemterv, ami generálta ezt a beépülő ütemezési feladatot, a konfiguráció-előkészítésekre vonatkozó korlátozások szempontjából megvalósítható-e vagy sem.

A $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ feladat definíciója garantálja, hogy ha minden egyes munka beütemezésre kerül, akkor az első három feltétel-típus már biztosan teljesül a célfüggvény értékétől függetlenül. Ha az L_{max} célfüggvény értéke legfeljebb nulla ($L_{max} \leq 0$), akkor a negyedik feltétel-típus is teljesül: $C_i \leq d_i$ ($i=1, 2, \dots, N_J$). Ebben az esetben az IPSIC feladat aktuálisan vizsgált teljes termelési ütemterve megvalósítható, tehát optimalitás szempontjából értékelhető. Ellenkező esetben az aktuális teljes ütemterv nem felel meg a korlátozásoknak, tehát nem megvalósítható.

4.3.6 Egy JIT-orientált megoldási módszer

A gyakorlatban a modern termelés-menedzsment számára a valós idejű termelésirányítás feladata úgy is megfogalmazható, hogy a gyártásirányítás „éppen időben” (Just-In-Time, JIT) paradigma szerint tudja biztosítani a kibocsátott termelési finomprogram végrehajtási feltételeit. Ez a szemlélet tovább finomíthatja a termelési folyamattal kapcsolatos követelményeket. Például a termeléshez szükséges aktuális szerszám-konfigurációkat nem szerencsés túl korán előkészíteni, mert ha valamilyen váratlan esemény következik be az előkészítés tényleges befejezési időpontja és az azt igénylő gyártási feladat vagy művelet tényleges indítási időpontja között, akkor adott esetben a termelési finomprogramot újra kell tervezni, és így a változások miatt az előkészítési műveletek egy része feleslegesen lett elvégezve.

Ebből következik, hogy célszerű az előkészítési tevékenységek végrehajtását az előírt határidőhöz (a kapcsolódó gyártási műveletek indítási időpontjához) a lehető legközelebbi befejezéssel ütemezni figyelembe véve a végrehajtási időket.

Az időtartalék-orientált algoritmus a konfiguráció-előkészítési munkák legnagyobb késését minimalizálja. Ez kiválóan alkalmazható a teljes IPSIC feladat legjobb termelési ütemtervét kereső algoritmus kiszolgálására azért, hogy a ciklikusan generált beépülő feladatokat megoldja. A keresési folyamat végén a végleges termelési ütemterv elfogadása után az ahhoz tartozó beépülő feladat megoldásának célfüggvény-értéke nem nagyobb nullánál. Ezt a konfiguráció-előkészítési ütemtervet tovább finomítom a JIT-elv alapján, miközben az elfogadott termelési ütemtervet változatlanul hagyom.

Ennek érdekében, kifejlesztettem egy második új algoritmust, amely képes csökkenteni a munkák legnagyobb sietésének mértékét úgy, hogy a legnagyobb késés ne lépje túl a nullát.

Adott J_i munka sietését (Earliness) – a késéshez hasonlóan – a virtuális időtengelyen az engedélyezett műszakok számában (lépésekben) mérem (s) a következőképpen:

$$E_i = \max(0, d_i - C_i),$$

ahol d_i a munka határideje (a legkésőbbi befejezési időkorlátját jelentő műszak sorszáma) és C_i a munka tényleges befejezési időpontja (a munka elvégzésére kijelölt műszak sorszáma).

Az új módszert JIT-orientált algoritmusnak nevezem. Ennek kényege a következő:

1. Inicializálás:

Készítsünk egy S_a kiinduló megoldást az időtartalék-orientált algoritmussal.

Készítsünk egy LDD listát úgy, hogy a munkák a határidők szerint nem növekvő sorrendben szerepeljenek a listán (Latest Due Date, LDD).

2. Válasszuk ki az LDD lista első elemét.

Jelölje a kiválasztott munkát J_{i^*} , a hozzá tartozó határidőt d_{i^*} , és C_{i^*} jelölje annak a műszaknak a sorszámát, amelyikbe az S_a szerint eredetileg be van ütemezve.

3. Vizsgáljuk meg a műszakok kihasználtságát:

A J_{i^*} munka d_{i^*} határidejének megfelelő műszaktól induljunk visszafelé az időben műszakról műszakra lépegetve. Ha találunk egy szabad gépet vagy elérjük a J_{i^*} munkához eredetileg hozzárendelt C_{i^*} műszakot, akkor álljunk meg.

4. Az ütemterv módosítása:

Ha találunk szabad gépet, akkor tegyük át a J_{i^*} munkát a szabad gépre.

Ha nincs szabad gép a vizsgált tartományban, akkor a J_{i^*} munka az eredeti C_{i^*} műszakban az eredeti helyén marad.

5. Töröljük a J_{i^*} munkát az LDD lista elejéről.

6. Ha az LDD lista nem üres, akkor ismételjük meg a lépéseket a 2. ponttól kezdve, máskülönben készen vagyunk.

Az időtartalék-orientált (első) algoritmus a teljes termelésprogramozási feladat keresési algoritmusának minden egyes jelölt megoldása esetében lefut, ezzel szemben a JIT-orientált algoritmus csak a teljes termelésprogramozási feladat kereső algoritmusának befejezése után egyetlen egyszer fut le. Ebből következik, hogy a finomítani kívánt S_a megoldás szerint minden munka határidőre befejeződik.

A JIT-orientált algoritmus az S_a ütemterv átalakításával egy új S_{JIT} ütemtervet hoz létre úgy, hogy minden egyes J_i munka továbbra is befejeződik legkésőbb a saját határidejére ($C_i \leq d_i \forall i=1, 2, \dots, N_j$), ráadásul a sietése (E_i) a lehető legkisebb értékre csökken. Ebből következik, hogy a legnagyobb csúszás értéke nulla ($T_{max}=0$), így az S_{JIT} ütemterv is megfelel az IPSIC feladat által támasztott szigorú követelményeknek és ezen túlmenően még a JIT paradigma gyakorlati alkalmazását is biztosítja.

A JIT-orientált algoritmus helyességének igazolása

Könnyen belátható, hogy ha az algoritmus áthelyezi a kiválasztott J_{i^*} munkát az eredetileg kijelölt műszakból egy későbbi műszakba, akkor a munka sietése csökken.

Definíció szerint a sietés:

$$E_i := \max(0, d_i - C_i) \quad (62)$$

Jelölje a J_{i^*} munka S_a szerinti eredeti befejezési időpontját $C_{i^*}(S_a)$, és az S_{JIT} szerinti új befejezési időpontját $C_{i^*}(S_{JIT})$.

Az algoritmus a J_{i^*} munka ütemtervből való kiemelését követően a szabad gép keresését a következőképpen végzi:

$$C_{i^*}(S_{JIT}) := \max_{s=C_{i^*}(S_a)}^{d_{i^*}} \left(s \mid P(s) > \sum_{i=1}^{N_J} \lambda(J_i, s) \right). \quad (63)$$

A vizsgált intervallum $C_{i^*}(S_a)$ alsó határa biztosítja, hogy a munka az eredeti műszaknál csak későbbre kerülhet vagy helyben marad. A felső határként beállított d_{i^*} garantálja, hogy a munka nem lépi túl a saját határidejét. Az algoritmus csak szabad géppel rendelkező műszakot választ ki, vagyis minden egyes vizsgált s műszak esetében a rendelkezésre álló virtuális gépek számára és a beütemezett munkák számára teljesül, hogy $P(s) > \sum_{i=1}^{N_J} \lambda(J_i, s)$.

Mivel a mindenkori kiválasztott munka biztosan csak későbbi műszakba helyezhető át, ezért $C_{i^*}(S_a) < C_{i^*}(S_{JIT})$, melyből következik, hogy $d_i - C_{i^*}(S_a) > d_i - C_{i^*}(S_{JIT})$, vagyis az áthelyezett munka sietése biztosan csökkent. Ha a vizsgált tartományban nincs szabad virtuális gép, akkor a munka nem helyezhető át, és így a sietése is és a késése is változatlan marad.

Az áthelyezéseket az algoritmus a munkát határidő szerinti nem növekvő sorrendjében végzi el, így a munkákat azokba a műszakokba is át tudja helyezni, amelyeket a futás során korábban áthelyezett munkák kiemelésével szabadított fel. Következésképpen a munkákat a lehető legnagyobb mértékben „jobbra húzza” az időben. Ha nincs mód egyetlen munka áthelyezésére sem a szűkös kapacitáskorlátok és az időtartalékok miatt, akkor a legkedvezőtlenebb esetben minden munka az eredeti helyén marad. A vizsgált rendszerben a gyakorlati alkalmazás eredményei azt mutatják, hogy a tervezési időhorizont elején rendszerint kicsi a munkák időtartaléka, de egy viszonylag rövid kezdeti szakasz után a nagyobb időtartalékoknak köszönhetően már érezhető a „jobbra húzás” módszerének hatása.

4.3.7 A konfiguráció-előkészítési ütemterv rugalmasságának növelése egy továbbfejlesztett JIT-orientált algoritmussal

Az IPSIC feladatba beépülő konfiguráció-előkészítések ütemezési problémáját a $P(s) \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ formalizmussal modelleztem. Ennek a feladatnak az optimális megoldását adja az időtartalék-orientált algoritmus. Ez a legnagyobb késés minimalizálására kiválóan alkalmas, de minden munkát a lehető legkorábbra ütemez. A külső forrásból érkező váratlan események (pl. rendelésmódosítás) káros hatásának csökkentése érdekében a JIT-orientált algoritmus a legnagyobb csúszást legfeljebb nullára emelve csökkenti a munkák túl korai befejezését. Ez viszont nagyon „éles” ütemtervet eredményez a belső előkészítési folyamatokban esetleg fellépő váratlan események miatt (pl. szerszám-meghibásodás).

E két véglet közötti rugalmas és hibatűrő megoldás elérése érdekében a JIT-orientált algoritmust kibővítettem munka-függő vezérlő paraméterekkel (32. ábra), amelyek definiálják minden egyes munka számára a biztonsági időtartalék nagyságát. Ez az érték azt írja elő, hogy ütemezéskor mennyi műszakkal korábbra célszerű tervezni az adott J_i munka C_i tényleges befejezési idejét a d_i határidejénél. Másképpen megfogalmazva, a J_i munkához kapcsolódó $E_{i,scp}$ biztonsági vezérlő paraméter (Safety Control Parameter) egy javasolt sietésnek tekinthető:

$$E_i \geq E_{i,scp} \quad (i=1, 2, \dots, N_J). \quad (64)$$

Ezeket a munka-függő $E_{i,scp}$ ($i=1, 2, \dots, N_J$) vezérlő paramétereket felhasználva, a JIT-orientált algoritmus szabad gépet kereső ciklusa a J_i munka esetében nem a munka d_i határidejétől indul visszafelé, hanem a d_i határidőből levonja a J_i munka saját $E_{i,scp}$ értékét, és onnan indul visszafelé. Ebből következik, hogy a munkák áthelyezése csak akkor történik meg,

ha az ilyenformán szűkített időtartományban talál szabad gépet. Belátható, hogy mivel csak a vizsgált intervallum felső határát csökkenti a biztonsági vezérlő paraméter, így az új eredmény továbbra is minden feltételnek megfelel. Ha a munka S_a ütemterv szerinti E_i sietése eleve kisebb mint az ajánlott minimális $E_{i,scp}$, akkor a munka az eredeti helyén marad.

A munka-függő biztonsági vezérlőparaméterek aktuális értékei az előkészítést ütemező algoritmus bemenő adatai közé tartoznak. A felhasználó közvetlenül megadhatja ezeket az értékeket, vagy jól definiált szabályokkal közvetetten is előírhatja. Például a konfigurációkat alkotó alapelemekre állíthat be különböző értékeket és az aktuális feladatban szereplő konfigurációk összetételéből automatikusan következik a definiált szabályok alapján a konkrét munkákra vonatkozó javasolt érték. Ezeket az értékeket múltbeli tapasztalatok alapján célszerű folyamatosan frissíteni (pl. adott formahordozó esetében korábban gyakran felmerült váratlan események miatt beállított nagyobb biztonsági érték csökkenthető ha a közelmúltban csökkent a kapcsolódó problémák száma).

Továbbfejlesztett JIT-orientált algoritmus

```

{ Készítsünk egy kiinduló  $S_a$  ütemtervet az időtartalék-orientált algoritmussal;
List  $\leftarrow$  Rendezzük a  $J_i$  ( $i=1, 2, \dots, N_J$ ) munkákat  $d_i$  szerint nem növekvő sorrendbe;
item  $\leftarrow 1$ ;
while (item  $\leq N_J$ )
{
   $J_{index} \leftarrow List[item]$ ;
   $s \leftarrow d_{item} - E_{item,scp}$ ;
  found  $\leftarrow false$ ;
  while ( $s > C_{index+1}$ )
  {
    if ( $P(s) > \sum_{i=1}^{N_J} \lambda(J_i, s)$ )
    {
      found  $\leftarrow true$ ;
      break;
    }
     $s \leftarrow s - 1$ ;
  }
  if (found == true)
  {
    Tegyük át a  $J_{index}$  munkát az  $s$  műszakba;
     $C_{index} \leftarrow s$ ;
     $L_{index} \leftarrow C_{index} - d_{index}$ ;
     $E_{index} \leftarrow \max(0, -L_{index})$ ;
  }
  item  $\leftarrow item + 1$ ;
}
Visszatérés az elkészített megoldással;
}

```

32. ábra: Továbbfejlesztett JIT-orientált ütemezési algoritmus

A bemutatott ütemezési módszer kifejezetten alkalmas a gyártásirányítás kifinomult elvárásainak teljesítésére azért, hogy hatékony, rugalmas és hibatűrő megoldásokat állít elő a mindenkori aktuális szituációnak megfelelően.

4.3.8 Következtetések és alkalmazási lehetőségek

Az IPSIC termelésprogramozási feladatba beépülő konfiguráció-előkészítési tevékenységek ütemezésére kidolgoztam egy új modellt. Ez egy időben változó kapacitáskorlátos erőforrás-környezetet reprezentál, amelyben egymástól független egyetlen operációt igénylő munkákat kell végrehajtani. A munkákhoz szigorú indítási időkorlátok, befejezési határidők és egységnyi műveleti idők kapcsolódnak. Az ütemezés elsődleges célja a határidők betartása.

A legnagyobb késés minimalizálására kidolgoztam egy új időtartalék-orientált ütemezési algoritmust, amely polinomiális futási idő alatt optimális megoldást állít elő. Az optimális megoldás ismeretében objektíven megválaszolható az a kérdés, hogy az IPSIC feladat teljes megoldása konfiguráció-előkészítés szempontjából megvalósítható-e vagy sem. Ha ugyanis a késés szempontjából optimális megoldás esetében is van határidő túllépés, akkor a generált beépülő feladat biztosan nem oldható meg határidő túllépés nélkül. Ráadásul mivel az algoritmus futási ideje polinomiális, így a teljes IPSIC feladat megoldását végző kereső algoritmusban a nagyszámú iteratív ismételt meghívás ellenére is kiválóan alkalmazható ez a módszer a feltételek teljesülésének ellenőrzésére nagyméretű feladatok esetében is.

Az időtartalék-orientált (első) algoritmusból kiindulva kidolgoztam egy JIT-orientált (második) megoldó algoritmust, amely a Just-In-Time paradigma gyakorlati alkalmazását támogatja a főfolyamatokat kiszolgáló segédfolyamatok ütemezésében. Ez a változat a végeredményt úgy módosítja, hogy a határidők megsértése nélkül a munkák „sietését” (túl korai elkészülését) csökkenti. A gyártásirányítási igények kiszolgálása érdekében biztonsági vezérlő paraméterek bevezetésével továbbfejlesztettem a módszert, ezáltal a konfiguráció-előkészítési tevékenységek ütemezésének rugalmasságát és hibatűrő képességét sikerült jelentős mértékben megnövelni.

A konkrét probléma sikeres megoldásán túlmenően általánosítható eredmények születtek. A kidolgozott konkrét modell és a kapcsolódó megoldási módszerek hatékonyan alkalmazhatók különböző ütemezési feladatok megoldására.

Az egyik lehetséges eset az, amikor az ütemezendő folyamat közvetlenül kezelhető a modellel. A gyártási főfolyamatok és segédfolyamatok sokféleségéből következik, hogy számos tevékenység egységnyi műveleti idővel modellezhető (pl. rutinszerű mérési/ellenőrzési folyamat, csomagolás, szerszám-előkészítés, belső logisztikai folyamat-elemek stb.), így ezekben az esetekben a modell közvetlenül alkalmazható, akár van változó kapacitáskorlát, akár rögzített kapacitáskorlátot kell figyelembe venni. A felsorolt példák alapvetően diszkrét gyártási folyamatokra vonatkoztak, ugyanakkor a problémát általánosabban szemlélve számos szolgáltatással kapcsolatos ütemezési feladat is ide sorolható. Ilyen feladat lehet például a nagy értékű, korlátozottan rendelkezésre álló orvosi műszerek és diagnosztizáló személyzet időkapacitásának figyelembe vétele határidős, sorban álló beavatkozások és/vagy vizsgálatok ütemezése során. További alkalmazási terület lehet például a nagyméretű logisztikai rendszerek

kiszolgálásakor jelentkező, dinamikusan generált beépülő feladatok (pl. repülőtéri üzemanyag-feltöltési tevékenységek, kikötői konténer rakodási műveletek stb.) ütemezése.

Az eredmények hasznosításának egy másik lehetséges módja az, amikor a konkrét modell és annak algoritmusai nem közvetlenül kerülnek bevetésre, hanem közvetett módon kerülnek alkalmazásba. Például, ha valamely folyamatban a műveleti időket nem célszerű vagy nem is lehet egységnyi értéknek tekinteni, a bemutatott koncepció akkor is alkalmazható, de már nem garantálható az optimális megoldás elérése. Ilyenkor a bemutatott módszerek gyors heurisztikus felépítő algoritmusoknak tekinthetők. Az NP-hard feladatok esetében ezeknek nagyon fontos szerepe van. Ezt figyelembe véve, a korlátozottan rendelkezésre álló párhuzamos kapacitásokból és az egyoperációs határidős munkákból álló ütemezési feladatok halmaza tovább szélesedik.

A kombinált időtartalék-orientált és JIT-orientált szemlélet használható több operációból álló munkák esetében is. Ilyenkor az adott munka indítási időpont-korlátja az elsőként indítható operációra vonatkozik, míg a határidő az utolsó operáció befejezési idejének szab felső határt. Ez alapján minden egyes operáció esetében számítható egy saját időtartalék úgy, hogy a kapcsolódó munka határidejéből ki kell vonni ennek a munkának a még el nem végzett operációinak műveleti időösszegét, továbbá ki kell vonni a döntés pillanatához tartozó mindenkor aktuális időpontot is. Ezt felhasználva az időtartalék-orientált algoritmus úgy működik, hogy a változó kapacitáskorlát mentén haladva ki kell választani a vizsgált időpontban indítható operációkat a teljes operációhalmazból, majd a legkisebb időtartalékkal rendelkező operációt kell beütemezni a lehető legkorábbi időpontra. Ha a kapacitás-korlátot eléri a terhelés, akkor tovább kell lépni az időben a következő szabad kapacitás rendelkezésre állásáig. Az ütemterv elkészítése után, hasonló megfontolásokkal lehet a JIT-orientált algoritmust is adaptálni a többoperációs esetre. Határidő szerint nem növekvő sorrendben haladva az egyes operációkat kell időben jobbra mozgatva közelíteni a saját határidejükhöz figyelembe véve a szabad kapacitásokat és a biztonsági vezérlő paraméterek által javasolt sietéseket.

Az így általánosított elvek önállóan vagy más heurisztikus módszerekkel kombinálva hatékonyan alkalmazhatók erőforrás-korlátos és határidős összetett ütemezési feladatok megoldására. Például az ismert klasszikus erőforrás-környezetek (pl. P, Q, R, F, J, O, X, G, stb. [10]) a bemutatott időben változó kapacitás-korlátokkal kiegészítve további új kiterjesztett modelleket alkotnak.

4.4 A termelés főfolyamatainak ütemezése

4.4.1 A konfiguráció-cserék ütemezési problémája

A termelési főfolyamat finomprogramozásakor a termelésütemezés és a készletgazdálkodás egy kombinált új feladatát kell modellezni és megoldani. Ez az IPSIC probléma, amelynek megoldására – annak összetettsége miatt – egy kétszintű döntési hierarchiát használunk. Minkét szintnek saját optimalizálási modellje van. Az elsődleges döntési változók a hierarchia felső szintjéhez tartoznak és meghatározzák a főfolyamatokat – a vizsgált járműipari üléselemek esetében – a konfiguráció-cserék ütemezésén keresztül. A másodlagos döntési változók az alsó

szinten kapnak értékeket a felső szinten generált beépülő feladat megoldásaként. Ebben az alfejezetben bemutatom az IPSIC feladat elsődleges döntési változóinak értékét beállító – a főfolyamat termékváltásait ütemező – megoldó algoritmus legfontosabb lépéseit.

Az IPSIC probléma befoglaló termelésprogramozási feladatának megoldására kifejlesztettem egy olyan algoritmust, amely a lokális szomszédságon alapuló keresési meta-heurisztika alapelvét kiterjeszti többféle szomszédsági (módosító) operátor egyidejű használatával és több célfüggvény egyidejű figyelembe vételével.

Az IPSIC optimalizálási feladatból kiemelem a másodlagos döntési változókat és a konfiguráció-előkészítési munkák ütemezésére vonatkozó korlátozásokat. Ezekkel a döntési hierarchia felső szintjén futó keresési folyamat közvetlenül nem foglalkozik. A kiemelt részt egyetlen új korlátozással helyettesítem. Felhasználva a 4.3. alfejezetben bemutatott ütemezési modellt és a hozzá tartozó megoldó algoritmusokat, az új feltétel akkor teljesül, ha a konfiguráció-előkészítési tevékenységek határidőre elvégezhetők ($L_{max} \leq 0$). Ez egy eldöntendő kérdés, melyre a választ megadja minden iterációban a hierarchia alsó szintjén beépülő ütemezési feladat egzakt megoldása.

Felhasználva ezt az új korlátozást és hivatkozva a 4.2.3. alfejezetben bevezetett IPSIC modellre, a felső szint optimalizálási feladata a következőképpen foglalható össze:

Elsődleges döntési változók

A gyártósorok pozícióiban az engedélyezett műszakokban aktívan használandó konfigurációk: $x_{m,j,c} \in \Omega (m=1, 2, \dots, N_M); (j=1, 2, \dots, N_{m,j}); (c=1, 2, \dots, N_{m,c})$.

Korlátozások

Az aktív konfigurációkra vonatkozó korlátozás-típusok: 1.-8.

A konfiguráció-cserék darabszámára vonatkozó korlátozás-típus: 9.

A konfiguráció-elemek használatára vonatkozó korlátozás-típusok: 10.-12.

A konfiguráció-előkészítési munkák ütemezésére vonatkozó korlátozás: $L_{max} \leq 0$.

Célfüggvények

$f_k \rightarrow \min. (k=1, 2, \dots, 18)$.

A továbbiakban ezt az optimalizálási feladatot nevezem termelésprogramozási feladatnak.

4.4.2 Továbbfejlesztett többcélú és többoperátoros keresési algoritmus

A termelésprogramozási feladat megoldása során az $x_{m,j,c} (m=1, 2, \dots, N_M; j=1, 2, \dots, N_{m,j}; c=1, 2, \dots, N_{m,c})$ döntési változók aktuális értékeinek beállításával a termékek gyártására alkalmas gyártósorok, formahordozók és formák lekötését (terhelését) kell úgy megtervezni, hogy a szigorú korlátozások megsértése nélkül a megfogalmazott célok elérhetőek legyenek.

A [23] értekezésben és a [24] cikkben korábban publikált legalkalmasabb kiindulási eredményeknek tartott többoperátoros és többcélú keresési algoritmusok jellemzőire támaszkodva, és kibővítve a vizsgált IPSIC feladat sajátosságainak megfelelő elemekkel egy

továbbfejlesztett módszert hoztam létre. A megoldást előállító kereső algoritmus egyszerűsített elvi vázlata a 33. ábrán látható.

```

EasyForce algoritmus
{
   $SCH_0 \leftarrow$  Kezdeti megoldás készítése a betöltött induló állapot érvényesítésével;
   $SCH^* \leftarrow SCH_0$ ;
   $Taboo\_List \leftarrow NULL$ ;
  while ( Leállási feltétel nem teljesül )
  { while ( Szomszédság kiterjesztésének feltétele teljesül )
    {
       $NEOP_z \leftarrow$  Az aktuális szomszédsági operátor kiválasztása (Priority_List);
       $SCH \leftarrow$  Szomszédos megoldás készítése ( $SCH_0, NEOP_z$ );
       $L_{max} \leftarrow$  Konfiguráció-előkészítések ütemezése ( $SCH$ );
      if (  $L_{max} \leq 0$  )
      {
        if ( A Taboo_List nem tartalmazza ( $SCH$ ) )
        {
          A Taboo_List bővítése új elemmel ( $SCH$ );
          if ( Taboo elemek száma > megengedett érték )
            A Taboo_List legkorábban felvett elemének törlése;
          if ( A szomszédság kiterjesztésének első eleme ( $SCH$ ) )  $SCH_k \leftarrow SCH$ ;
          else if (  $SCH < SCH_k$  )  $SCH_k \leftarrow SCH$ ;
        }
      }
       $SCH_0 \leftarrow SCH_k$ ;
      if (  $SCH_k < SCH^*$  )  $SCH^* \leftarrow SCH_k$ ;
    }
  }
  return  $SCH^*$ ;
}

```

33. ábra: Továbbfejlesztett többoperátoros, többcélú kereső algoritmus.

A 33. ábrán alkalmazott jelölések a következők:

- SCH_0 – a szomszédság képzésének bázis megoldása;
- SCH – az aktuális szomszédos megoldás;
- SCH_k – az adott kiterjesztés legjobb ütemterve;
- SCH^* – a keresés során megtalált legjobb megoldás;
- $NEOP_z$ – az aktuálisan kiválasztott szomszédsági (módosító) operátor;
- Taboo_List* – a megjegyzett megoldásokat tároló tabulista;
- Priority_List* – a szomszédsági operátorok kiválasztását szabályozó prioritáslista;

Az *EasyForce* algoritmus működésének alapelve hasonló a 3.3.3. alfejezetben bemutatott *MOMOTS* kereső algoritmus működési elvéhez. A lényegi különbség a kezdeti megoldás előállításában, a megoldások struktúrájában, a tabuelemek felépítésében és az alkalmazott szomszédsági operátorok működésében van.

Az integrált termelésütemezési és készletgazdálkodási problématípus (IPSIC) igényeit az új szomszédsági (módosító) operátorok, az új célfüggvények és a termelési rendszer működését szimuláló algoritmusok együttesen elégítik ki. Ezek a kiterjesztések tetszőleges keresési meta-heurisztikába (pl. genetikus algoritmus-vázba) beilleszthetők.

Az *EasyForce* keresési folyamat során egy SCH_0 kezdeti termelési ütemtervből kiindulva a $NEOP_z$ szomszédsági operátorok által elvégzett konzisztens módosítások ismételt végrehajtásával, az előállított ütemtervek összehasonlító értékelésével alakul ki az SCH^* végső ütemterv.

Induláskor a kezdeti SCH_0 ütemterv úgy készül, hogy a gyártórendszer aktuális állapota változtatás nélkül kerül beállításra minden engedélyezett műszakba:

$SCH_0 \leftarrow$ Kezdeti megoldás készítése a betöltött induló állapot érvényesítésével;

Ennek lényege az, hogy az elsődleges döntési változók értékét az induló konfigurációk adják: $x_{m,j,c} = x_{m,j,0}$ ($m=1, 2, \dots, N_M; j=1, 2, \dots, N_{m,j}; c=1, 2, \dots, N_{m,c}$). Ennek hatására a gyártósorok minden egyes pozíciójához hozzárendeli a kiindulási állapotának megfelelő konfigurációt változtatás (csere) nélkül az ütemezési időhorizont minden aktív műszakjában.

A kereső algoritmus egy külső és egy belső ciklust tartalmaz. A külső ciklus vezérli a keresési lépéseket, míg a belső ciklus az adott lépéshez tartozó bázismegoldás szomszédos megoldásait állítja elő. A teljes szomszédság halmaza helyett, annak egy részhalmazát (kiterjesztés) vizsgálja meg a belső ciklus.

Az iteratív javítás egy közbenső lépése során, a belső ciklus az aktuális SCH_0 bázismegoldásból kiindulva paraméter által megadott számú kiterjesztett SCH ütemtervet készít. Egy ilyen szomszédos megoldás úgy készül, hogy a $NEOP_z$ ($z=1, 2, \dots, 12$) szomszédsági operátorok halmazából kiválasztásra kerül egy operátor, majd ez az operátor készít egy másolatot a bázismegoldásról és végül az $x_{m,j,c}$ döntési változók értékeit módosítja. A módosító operátorok kiválasztását prioritáslista (*Priority_List*) és kvázi-véletlenszám generátor együttműködése határozza meg. A prioritáslista az algoritmus paramétereként az operátorok kiválasztási valószínűségét definiálja.

A rendszer megoldja az új jelölt SCH ütemterv által igényelt konfiguráció-előkészítési tevékenységek ütemezési feladatát az „időtartalék-orientált” algoritmussal:

$L_{max} \leftarrow$ Konfiguráció-előkészítések ütemezése (SCH);

Ha a legnagyobb késésre kapott eredmény nullánál nem nagyobb ($L_{max} \leq 0$), akkor a vizsgált SCH ütemterv érvényes és bekerül a szomszédságba, máskülönben elvetésre kerül.

A lokális optimumból való kilépést és a kellő mértékű eltávolodást egy rövid távú memóriaként működő tabulista (*Taboo_List*) segíti. Ezen a listán olyan megoldások vannak egyszerűsített formában nyilvántartva, melyeket az algoritmus a „közelmúltban” megvizsgált. Minden egyes tabuelem egy megoldást reprezentál úgy, hogy a megoldáshoz tartozó konfiguráció-cseréket tárolja egy listán időrendi sorrendben, azon belül gyártósor és pozíció szerint rendezve. E hármasszoros rendezésnek köszönhetően egy tabuelem és egy megoldás gyorsan és egyszerűen összehasonlítható, mert csak a két megoldásra jellemző konfiguráció-cseréket

kell összevetni a listaelemeken végighaladva. Az első eltérés azonnal jelzi a két megoldás különbözőségét. A két lista egyezősége esetén a két megoldás azonos.

Ha egy kiterjesztett SCH ütemterv már szerepel a tabulistán, akkor az algoritmus azt nem értékeli ki újra, hanem elveti, ellenkező esetben felkerül a tabulistára, és ha a megengedett tabuelemek száma elérte a maximális értéket, akkor a legkorábban felvett listaelem törlődik. Így a rövidtávú memória „felejt”.

A belső ciklusban a mindenkor SCH ütemterv szimulációs kiértékelését követően elérhetővé válnak a célfüggvények aktuális értékei (f_1, f_2, \dots, f_{18}). Két megoldás több célfüggvény szerinti összehasonlítására a 3.3.3. alfejezetben bemutatott módszert használom: Értelemszerűen az eredeti definícióban szereplő s_x és s_y helyett itt most SCH_x és SCH_y szerepel.

Ha a célfüggvény-értékek alapján a kiterjesztett ütemterv jobb, mint az adott kiterjesztés legjobb ütemterve ($SCH < SCH_k$), akkor az felülírja a kiterjesztés legjobb megoldását. A külső ciklus által vezérelt keresési lépések úgy kapcsolódnak össze, hogy a belső ciklusban vizsgált kiterjesztés legjobb ütemterve lesz a következő lépés kiindulási bázisa ($SCH_0 \leftarrow SCH_k$), és ha ez a megoldás jobb, mint a keresés során megtalált legjobb megoldás ($SCH_k < SCH^*$), akkor ez felülírja a legjobb megoldást ($SCH^* \leftarrow SCH_k$).

A hatalmas keresési térben található nagyon nagyszámú lehetséges ütemterv generálását és vizsgálatát az *EasyForce* algoritmus csupán négy darab (SCH_0, SCH, SCH_k, SCH^*) ütemtervet reprezentáló változóval oldja meg. Ezek az algoritmus elején jönnek létre és a végén szűnnek meg annak köszönhetően, hogy a belső memóriamodelljük teljesen azonos és a döntési változók értékei egymásba átmásolhatók, így az időigényes memóriefoglalás és felszabadítás iteratív ismétlésére nincs szükség. Ez futás közben jelentős sebességnövekedést eredményez.

4.4.3 A szomszédsági operátorok működése és szerepe

A $NEOP_z$ ($z=1, 2, \dots, 12$) szomszédsági operátorok a gyártósorok pozícióiban az adott időhorizonton aktívan használt konfigurációkra vonatkozó $x_{m,j,c}$ döntési változókat módosítják.

Minden egyes terméktípushoz hozzá van rendelve egy alapértelmezett gyártósor. Ezt a technológia szempontjából elsődleges gyártósort mindig a felhasználó által megadott bemenő adatok definiálják. Adott terméktípus gyártása szempontjából az összes többi alkalmas gyártósort alternatív gyártósornak tekinti a rendszer. Ezekre akkor ütemezi a termék gyártását, ha az alapértelmezett gyártósor valamilyen okból túlterhelt a tárgyidőszakban.

Az alkalmazott szomszédsági (módosító) operátorok a következők:

- $NEOP_1$: Egy véletlenszerűen kiválasztott gyártósor véletlenszerűen választott pozíciójáról egy véletlenszerűen választott műszakban leveszi az aktuálisan használt konfigurációt.
- $NEOP_2$: Egy véletlenszerűen választott rendeléshez tartozó termék gyártását ütemezi be az aktuális termék alapértelmezett gyártósorán. Ha ez nem lehetséges, akkor egy alternatív gyártósort választ.
- $NEOP_3$: Egy véletlenszerűen választott csúszó rendeléshez tartozó terméket választ és azt a $NEOP_2$ működése szerint ütemezi be.
- $NEOP_4$: Egy véletlenszerűen választott csúszó rendeléshez tartozó terméket választ és annak gyártását egy véletlenszerűen választott alkalmas gyártósorra ütemezi be.

- *NEOP₅*: A legnagyobb prioritású csúszó rendeléshez tartozó terméket választja ki és annak gyártását egy véletlenszerűen választott alkalmas gyártósorra ütemezi be.
- *NEOP₆*: Ez valójában három operátort foglal magába, melyek közül véletlenszerűen kerül kiválasztásra az egyik minden meghíváskor:
 - A termékhiányos (csúszó) rendelések közül kiválasztja a legkevesebb formával rendelkezőt, és a hozzá tartozó terméket beilleszti az ütemtervbe.
 - A termékhiányos (csúszó) rendelések közül kiválasztja a legkorábbi határidejűt, és a hozzá tartozó terméket beilleszti az ütemtervbe.
 - A termékhiányos (csúszó) rendelések közül kiválasztja a legkevesebb gyártósoron gyárthatót, és a hozzá tartozó terméket beilleszti az ütemtervbe.
 Mindhárom esetben az operátor az alkalmas gyártósorok közül a legkevesebb átállítással terhelt gyártósorra ütemezi a kiválasztott termék gyártását.
- *NEOP₇*: Az alábbi két operátor közül véletlenszerűen kiválasztja az egyiket és azt aktiválja:
 - A legnagyobb prioritású csúszó rendeléshez tartozó terméket választja ki. Ha a termék minden formája foglalt, akkor véletlenszerűen felszabadít egyet és azt felhasználva az alkalmas gyártósorok közül a legkevesebb átállítással terhelt gyártósorra ütemezi a kiválasztott termék gyártását.
 - Egy véletlenszerűen választott csúszó rendeléshez tartozó terméket választ ki. Ha a termék minden formája foglalt, akkor véletlenszerűen felszabadít egyet és azt felhasználva az alkalmas gyártósorok közül véletlenszerűen választott gyártósorra ütemezi a kiválasztott termék gyártását.
- *NEOP₈*: Egy véletlenszerűen választott konfiguráció-cserét töröl az ütemtervből.
- *NEOP₉*: Kiválasztja a legnagyobb csúszással jellemezhető rendelést és a hozzá tartozó termék gyártását egy véletlenszerűen választott alkalmas gyártósorra ütemezi be.
- *NEOP₁₀*: Kiválasztja a legnagyobb csúszással jellemezhető rendelést és a hozzá tartozó terméket jelöli ki. Ha a termék minden formája foglalt, akkor véletlenszerűen felszabadít egyet és azt felhasználva a kijelölt termék alapértelmezett gyártósorára ütemezi be. Ha ez nem lehetséges, akkor egy alternatív gyártósort választ.
- *NEOP₁₁*: Az ütemezési időszak végét alapul véve, az előírt minimális készlethez viszonyítva a legnagyobb hiánnyal rendelkező terméket választja ki, és azt a hozzá tartozó alapértelmezett gyártósorra ütemezi be. Ha ez nem lehetséges, akkor egy alternatív gyártósort választ.
- *NEOP₁₂*: Az ütemezési időszak végét alapul véve, az előírt minimális készlethez viszonyítva hiánnyal rendelkező termékek közül véletlenszerűen kiválaszt egyet, és azt a hozzá tartozó alapértelmezett gyártósorra ütemezi be. Ha ez nem lehetséges, akkor egy alternatív gyártósort választ.

A felsorolt szomszédsági operátorok – a *NEOP₁* és a *NEOP₈* kivételével – teljes konfiguráció-cserén alapuló módosítást végeznek el a bázismegoldáson. Ezek alapvetően abban különböznek egymástól, hogy eltérő módszerrel választják ki a beütemezendő terméktípust és annak gyártását megvalósító gyártósort. A beütemezés módszere azonban mindegyik operátor

esetében azonos. A kiválasztott terméktípus és gyártósor ismeretében a legfontosabb részfeladatok a következők:

- Pozíció kiválasztása.
- Kezdő műszak kiválasztása.
- Befejező műszak kiválasztása.
- Szabad formák (szerszámok) lekérdezése.
- Szabad formahordozók lekérdezése.
- Bevethető előszerelt konfigurációk lekérdezése.
- Konfiguráció kiválasztása/összeállítása.
- A változtatások aktiválása.

Egy kiválasztott M_m gyártósoron egy kiválasztott P_p terméktípus gyártásának beütemezése azt jelenti, hogy kiválasztásra kerül egy alkalmas $M_{m,j}$ pozíció és egy olyan $s_{m,c}$ műszak, amelyben az előírt $\Delta_{m,c}$ csereszám-korlát megsértése nélkül egy új csere végrehajtható. A csere során a kiválasztott $M_{m,j}$ pozícióból leveszik az addig aktív $x_{m,j,c-1}$ konfigurációt, és a helyére egy másik $x_{m,j,c} = \omega \in \Phi_s$ már előkészített konfigurációt csatlakoztatnak. Ez a konfiguráció a következő tervezett cseréig lesz aktív az adott pozícióban. Ha nincs későbbi csere betervezve, akkor az ütemezési időszak végéig aktív marad.

A szomszédsági operátor egy alkalmas formahordozóból és a rászertelt formából vagy formákból „állítja össze” a kiválasztott ω konfigurációt a következő műveletekkel:

- A forma kiválasztása előtt lekérdezi a kijelölt időszakban szabad (nem aktív) formák listáját.
- Megvizsgálja, hogy van-e olyan előkészített konfiguráció, ami bevethető (tartalmaz megfelelő formát).
- Ha igen, akkor azokból választ egyet és azt használja. Ha nem áll rendelkezésre alkalmas konfiguráció abban az időszakban, akkor a szabad formahordozók közül választ ki egyet, továbbá a szabad formák közül is kiválaszt egyet.
- Amennyiben a formahordozóra több forma is elhelyezhető, akkor több formát is kiválaszt a rendelkezésre állók közül. Ha lehetséges, akkor a kijelölt P_p terméktípus több formájából állítja össze a konfigurációt. Ha nincs rá mód, akkor ebben a döntési helyzetben azt is figyelembe veszi, hogy mely terméktípusok gyárthatók együtt.

Ennek a „szituációtól függő intelligens viselkedésnek” köszönhetően a módosító operátorok kulcsszerepet játszanak a keresési folyamatban, mert mindig csak megvalósítható termelési ütemtervet állítanak elő betartva az IPSIC feladat következő korlátfeltételeit:

- az aktív konfigurációkra vonatkozó korlátozás-típusokat (1.-8.),
- a konfiguráció-cserék darabszámára vonatkozó korlátozás-típust (9.),
- a konfiguráció-elemek használatára vonatkozó korlátozás-típusokat (10.-12.).

Az első csoportba tartozó korlátfeltételek érvényesítését úgy valósítja meg a rendszer, hogy az alapadatokban megadott értékekből kiindulva részletes vizsgálatot végez még a kereső algoritmus futása előtt. Az összerendelésekkel kapcsolatos halmazokat listákra gyűjti. Ezeket a listákat a kapcsolódó objektumokban tárolja adattagként. Így bármely objektumból kiindulva

gyorsan elérhető a vele kompatibilis objektumok. A keresési folyamat valamely közbenső iterációjában az aktuális konfiguráció összetételének meghatározásakor az operátorok mindig egy kiválasztott terméktípust vesznek alapul. Ebből kiindulva keresnek egy alkalmas pozíciót és szabad kompatibilis formahordozót az előzetesen elkészített listák alapján. Hasonlóan előkészített lista segíti a kiválasztott terméktípussal együttgyártható terméktípusok halmazának lekérdezését is. A konfigurációt alkotó formák kiválasztásánál a helyfoglalásra vonatkozó méretkorlátot is az operátorok veszik figyelembe. A listákról a konkrét objektumok kiválasztásának módja alapvetően kvázi egyenletes valószínűséggel generált random számokra van leképezve.

A második csoportba tartozó korlátozásokot a rendszer úgy veszi figyelembe, hogy minden egyes gyártósorhoz hozzárendel egy vektort, melynek elemei számlálóként működnek és minden egyes c sorszámú műszakban jelzik az addig betervezett cserék aktuális számát. Ez a nyilvántartás teszi lehetővé, hogy a módosító operátorok gyorsan áttekinthessék az új cseréket befogadni képes műszakok halmazát, és így eleve csak olyan műszakkal próbálkoznak, amelyiknél a kapcsolódó számláló még nem érte el a csereszám-korlátot.

A harmadik csoportba tartozó korlátozások betartását az garantálja, hogy minden konfigurációelem számára a rendszer létrehoz egy foglaltság-nyilvántartó vektort. A vektor indexelt rekeszei mutatják az s műszakokban az adott konfiguráció-elem foglaltságát. A módosító operátorok a működésük közben folyamatosan aktualizálják ezeket a foglaltsági vektorokat. Ezt felhasználva adott időpillanatban adott elem biztosan csak egyetlen konfigurációhoz lehet hozzárendelve.

A keresési folyamatban az adott lépéshez tartozó szomszédságot több iterációban többféle módosító operátor együtt állítja elő. A szomszédság képzésében résztvevő operátorok kiválasztási valószínűségét egy prioritáslista írja le, melynek értékeit a felhasználó állíthatja be. A bemutatott többoperátoros működési mód változó szomszédsági-struktúrára alapuló keresési algoritmust valósít meg.

4.4.4 A szimuláció szerepe

Az *EasyForce* kereső algoritmus által vizsgált termelési ütemtervek minősítésének alapját az optimalizálási célfüggvények aktuális értékei adják. Ezek kiszámításához szükség van a teljes termelési folyamat gyors szimulációjára. Egy termelési ütemterv (megoldás jelölt) végrehajtását jellemző időpontok és készletszint-értékek számítását egy numerikus eljárás végzi el. A szimuláció szoftveres implementálását a Miskolci Egyetem Alkalmazott Informatikai Intézeti Tanszékén Kulcsár Gyula végezte el. Az algoritmus figyelembe veszi az általam definiált IPSIC feladat ismertett bemenő alapadatait, a döntési változók aktuális ütemterv szerinti értékeit, a korlátozásokat és a termelési rendszer működési jellemzőit. A szimuláció determinisztikus működési modellt használ. Adott ütemterv többszöri szimulációja pontosan ugyanarra az eredményre vezet.

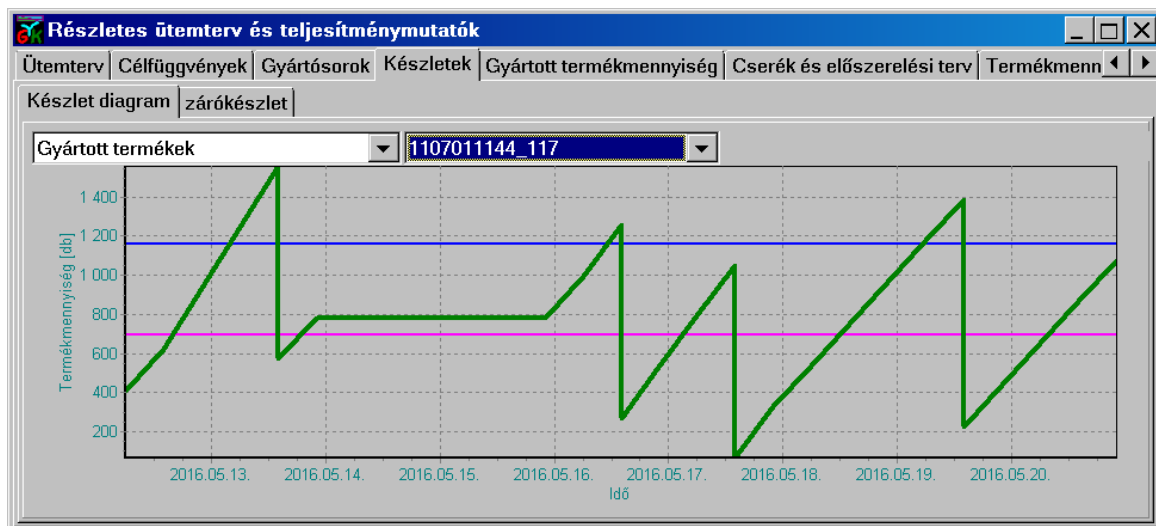
A szimuláció legfontosabb része az idő múlásának modellezése. A bemenő adatok egyértelműen meghatározzák a gyártósorok engedélyezett rendelkezésre állási időintervallumait (műszakjait). Mivel ezeket az ütemező szoftver nem módosíthatja, így a kereső algoritmus indítása előtt előzetesen felépül egy virtuális idővonal.

Az idővonal alapját a teljes gyártórendszer műszakjainak uniójához tartozó s sorszámok ($s=0, 1, 2, \dots, s_{max}$) adják. Az idővonal koncepció biztosítja bármelyik engedélyezett műszakhoz tartozó s intervallum-sorszám, m gyártósor-azonosító, és c gyártósor szerinti műszak-sorszám gyors elérését.

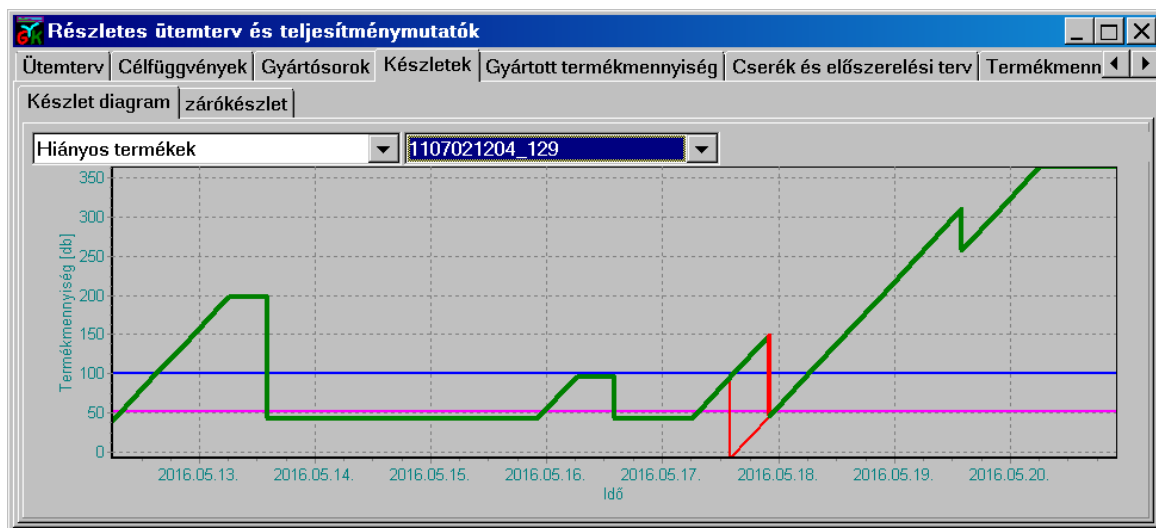
Az idővonal koncepció a gyártási rendeléseket is az s sorszámok szerint rendezett formába transzformálja. Ez azt jelenti, hogy minden egyes megrendelést hozzárendel ahhoz a legnagyobb s sorszámú időkerethez, amelynek befejezési időpontja még nem haladja meg az adott rendelés saját határidejét. Ilyenformán előáll egy rendelés mátrix.

Az idővonalhoz kapcsolódó előzetesen elkészített objektumok adják az alapját egy olyan végrehajtás-vezérelt szimuláció megvalósításának, amely a termelési folyamatot időkeretekre osztja, majd az időkereteket járja végig időugrásokkal. Egy adott s sorszámú időkeretben több konkrét gyártósor is aktív lehet. Ilyenkor az algoritmus végigjárja az aktív gyártósorokat és azok minden egyes pozícióját. A döntési változók által előírt konfigurációkban szereplő aktív formák kapacitásából valamint a gyártósorok műszakonként elvégezhető ciklusainak számából kiszámítja az adott műszakban az előírt terméktípusokból gyártott darabszámokat. Minden egyes műszakot homogénnek tekint és az azon belül gyártott termékek mennyiségét az intervallum kezdetétől eltelt idővel egyenes arányban növeli. Az azonos s sorszámhoz tartozó műszakokban gyártott termékek darabszámát terméktípusonként külön-külön összegzi és az idővonal szerkezetének megfelelően tárolja. Így előáll egy termék-időkeret mátrix. Ezzel befejeződik a szimuláció első fázisa.

Az IPSIC feladat megoldásának támogatására a gyors szimuláció egyik legfontosabb feladata a terméktípusonkénti készlet szintek időbeli alakulásának számítása. A szimuláció második fázisát megvalósító algoritmus az induló készlet szinteket alapul véve a termék-időkeret mátrix celláiban található mennyiségeket és a rendelés mátrix celláiban tárolt mennyiségeket az s időkereteken végighaladva konzisztensen összevonja, ezáltal előállítja az aktuális készlet szint-idő diagramot minden egyes terméktípus esetében (34. ábra). Az aktuális értékek felhasználásával az adott referencia alsó és felső határokhoz viszonyított hiány vagy többlet egyszerűen számítható. A rendelések teljesítésének csúszása szintén egyszerűen számítható a készlet-idő diagram alapján. Az előírt határidő pillanatában az aktuális készlet szint és a rendelt mennyiség különbségeként adódó előjeles érték mutatja a hiányt (negatív készlet) vagy a többletet (pozitív készlet). A rendelés teljesítésének csúszása azt az időtartamot jelenti, amely az előírt határidő és a tényleges teljesítés között eltelik (35. ábra).



34. ábra: Készlet-idő diagram.



35. ábra: A rendelés teljesítésének csúszása készlethiány (negatív készlet) miatt.

A készletszintek változásának és a rendelések teljesítésének számításán túlmenően a gyors szimuláció a termelési folyamatban használt osztott hozzáférésű erőforrások használatát is nyomon követi. A konfiguráció-cserék által okozott állapotváltozásokat az algoritmus nemcsak a gyártósok esetében, hanem a formák, a formahordozók és az előszerelt konfigurációk esetében is nyilvántartja. A nyilvántartás alapját itt is az idővonal jelenti. Az elsődleges döntési változók értékeinek változását az s időkereteken végighaladva vizsgálja. Adott pozícióhoz rendelt aktív konfigurációk értékében felismerhető változás jelzi a konfiguráció-cserét. A módosító operátorok működésének leírásakor vázolt foglaltság-nyilvántartó vektorok a formák és formahordozók aktív/passzív állapotát egyértelműen mutatják minden időkeretben. Ezen túlmenően a rendszer időkeretenkénti bontásban nyilvántartja az előszerelt konfigurációk listáit is. A korábbi szemléletnek megfelelően ezeket is egy konfiguráció-idő mátrix tárolja. Egy előszerelt konfiguráció egyidejűleg vagy valamelyik gyártósor aktív pozíciójához, vagy a konfiguráció-idő mátrixhoz van hozzárendelve. A szimuláció ezeket az állapotnyilvántartásokat is konzisztensen frissíti azáltal, hogy a kezdeti állapotból kiindulva elvégzi a tervezett konfiguráció-cseréket időrendi sorrendben. Minden egyes konfiguráció-csere

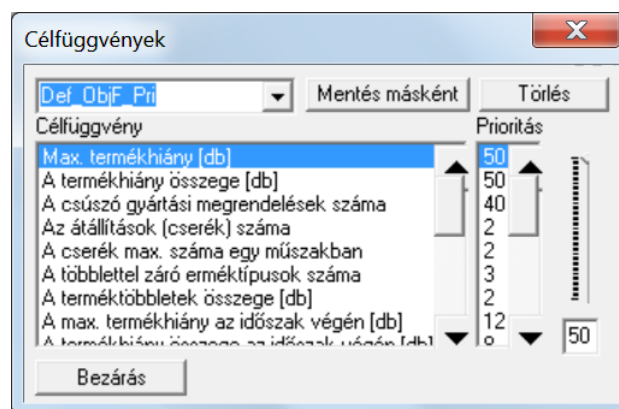
esetében a kijelölt időponttól az ütemezési időszak végéig elvégzi a nyilvántartási segédváltozók aktualizálását. A levett konfigurációt átteszi a konfiguráció-idő mátrix megfelelő listáira, valamint a konfigurációt alkotó elemek vektoraiban jelzi azok szabad állapotát. A felhelyezett konfiguráció-elemait foglalt állapotba teszi, és ha azok korábban más konfigurációkhoz tartoztak, akkor azokat a konfigurációkat leveszi a konfiguráció-idő mátrix listáiról. Így bármikor könnyedén lekérdezhető az osztott hozzáférésű erőforrások időbeli rendelkezésre állása (állapota).

A vázolt végrehajtás-vezérelt szimuláció és az ütemtervhez kapcsolódó metódusok együttesen szolgáltatják az IPSIC feladat optimalizálási modelljében definiált segédváltozók konkrét értékeit és a bevezetett függvények visszatérési értékeit. Összefoglalva ezek a következők:

- A Θ_π megrendelés teljesítésének tényleges időpontja: Z_π ($\pi=1, 2, \dots, N_\Theta$), és az ezek alapján számítható U_π és ζ_p függvények.
- A P_p terméktípus t időpontbeli készlet szintjét lekérdező $\sigma(p, t)$ függvény és a magas zárókészletet jelző ξ_p függvény.
- A konfiguráció-cserét jelző $\delta(x_{m,j,c}, x_{m^*,j^*,c^*})$ függvény.
- Adott konfiguráció-elem valamely időkeret előtti utolsó felszabadulását jelző $\pi(\mathcal{G}, s)$ függvény.
- Az s időkeretben összeszerelten rendelkezésre álló inaktív konfigurációk Φ_s halmazát lekérdező függvény.
- Az ütemterv (az elsődleges döntési változók aktuális értékei) által igényelt J_i ($i=1, 2, \dots, N_J$) konfiguráció-előkészítési munkákat definiáló metódus.

4.4.5 Igény szerint változtatható optimalizálási célfüggvény-rendszer

A termelés-menedzsment által megfogalmazott ütemezési célok fontossága időben változhat. Ennek rugalmas követése érdekében az optimalizálási célfüggvények aktuális fontosságát prioritásértékek megadásával fejezheti ki a felhasználó (36. ábra).



36. ábra: A célfüggvények prioritásainak konfigurálását támogató felület.

A prioritásokkal kiegészített célfüggvény-rendszer kezelésére – a 3.3.3. alfejezetben bemutatott esethez hasonlóan – a szakirodalomban „*relatív minősítésre alapozott többcélú eredmény-értékelő matematikai modell*” néven a [24] cikkben publikált megoldás koncepcióját használom. A kiválasztott módszer lényege az, hogy két vizsgált megoldás összehasonlításakor az egyik megoldás másik megoldáshoz viszonyított jóságának számértéke alapján dől el, hogy melyik a jobb megoldás. A két megoldás minden egyes célfüggvény szempontjából összehasonlításra kerül és számszerűvé válik az adott szempont szerinti százalékos változás mértéke. A célfüggvények prioritásaival súlyozott relatív változások összege határozza meg a két vizsgált megoldás viszonyát. Erre a – Kulcsár Gyula értekezésében [23] leírt – javaslatra támaszkodva készült az IPSIC feladathoz a többcélú (multi-objective) szemléletet is magába foglaló megoldási módszer.

A felhasználó a prioritásértékek beállításával könnyedén tudja szabályozni az ütemezési célok fontosságának mértékét. Az IPSIC feladatban a prioritásértékek együtt egy sémát alkotnak. A sémák elmenthetők és visszatölthetők, valamint meglévők finoman hangolhatók vagy új néven elmenthetők. A felhasználó igényei a különböző sémákkal egyszerűen és hatékonyan menedzselhetők.

A vizsgált valós ipari környezetben elvégzett tesztek alapján a termelés-menedzsment agilis gyártásra orientált stratégiájának legjobban megfelelő prioritásértékeket a 6. táblázat foglalja össze.

6. táblázat: A célfüggvények prioritásértékei egy konkrét feladatban (Agile Control).

Célfüggvény	Prioritás-érték
f_1 : A rendelések teljesítésének időpontjában fellépő legnagyobb termékhiány [db]	42
f_2 : A rendelések teljesítésének időpontjában fellépő termékhiányok összege [db]	44
f_3 : A csúszó (határidőt túllépő) rendelés-teljesítések száma	44
f_4 : A konfiguráció-cserék száma	2
f_5 : A konfiguráció-cserék legnagyobb száma egy műszakban	2
f_6 : A többlettel záró terméktípusok száma	3
f_7 : A terméktöbbletek összege az időszak végén [db]	2
f_8 : A legnagyobb termékhiány az ütemezési időszak végén [db]	12
f_9 : A termékhiány összege az ütemezési időszak végén [db]	8
f_{10} : A határidőre nem teljesített rendelések prioritásainak összege	1
f_{11} : A határidőre nem teljesített rendelések legnagyobb prioritása	1
f_{12} : A csúszó rendelés-teljesítések által érintett terméktípusok száma	1
f_{13} : A legnagyobb termékhiány (nullához képest) az ütemezési időszak végén [db]	1
f_{14} : A termékhiány összege (nullához képest) az ütemezési időszak végén [db]	1
f_{15} : A rendelés-teljesítések legnagyobb csúszása (határidő túllépése) [műszak]	50
f_{16} : A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak]	50
f_{17} : A konfiguráció-előkészítési munkák száma	1
f_{18} : Nem használt pályakapacitás [SPU]	40

Ezzel a kiemelt példával modellezett ütemezési stratégiában azok a célfüggvények kapják a legnagyobb prioritást, amelyek a rendelések határidőre történő teljesítését szolgálják. Az is megfigyelhető, hogy a beállított értékek azokban az esetekben, amikor túlterhelés miatt biztosan nem teljesíthető minden rendelés határidőre, azokat az ütemterveket részesítik előnyben, amelyek inkább több megrendelés esetében okoznak kisebb csúszásokat, mintsem kevesebb rendelés esetében legyen nagyobb mértékű csúszás. A kapacitáskihasználás növelését szolgáló célfüggvény szintén viszonylag magas prioritást kap. Ennek a háttérben egyrészt az az igény húzódik meg, hogy a konfigurációk összeállítása olyan legyen, hogy az aktív formahordozók és ezáltal a gyártósorok kihasználtsága minél nagyobb legyen, másrészt inkább legyen kisebb mértékű készlet-felhalmozás az időszak végén, minthogy előírt készlet szintek alakuljanak ki, de alacsony kihasználtsággal működjenek a gyártósorok. A készlet szintek alakulását szabályozó célfüggvények és az átállításokat mérséklő célfüggvények alárendelt szerepet játszanak a határidők betartását segítő célfüggvényekhez képest. Az alacsony prioritású célfüggvényeket sem szabad azonban elhanyagolni, mert ha a feladat megoldási folyamatában a magasabb prioritású célfüggvények értékei bizonyos szint alá már nem vihetők, akkor a rendszer az egyre kisebb prioritású célfüggvények szempontjából még javíthat a megoldás minőségén.

A prioritások megfelelő kiosztásával nagyon sokféle ütemezési stratégiai kialakítható. Az ütemező rendszer egyszerűen finoman hangolható, ezáltal a piaci környezetből származó változó igények könnyen és gyorsan követhetők.

4.4.6 Egy valós alkalmazási példa és néhány jellemző futási eredmény

A bemutatott ütemezési modellek és algoritmusok C++ programozási nyelven kerültek implementálásra és egy már létező keretrendszerbe lettek beillesztve. Az új funkciókkal rendelkező termelésprogramozó szoftver a MO2PS fantázianevet kapta (Multi-Objective and Multi-Operator Production Scheduler). Az implementációs és kódolási feladatokat a keretrendszer eredeti fejlesztője végezte az általam kidolgozott modellek és algoritmusok alapján.

Az IPSIC modell sajátosságai miatt a szakirodalomban összehasonlításra alkalmas tesztadatokat nem találtam. A fejlesztés és a tesztelés során a Fehrer Hungaria Járműipari Kft. által megadott valós ipari adatokat használtam. A MO2PS szoftver fejlesztésének anyagi fedezetét szintén ez a cég biztosította.

Illusztratív példaként bemutatok egy valós ipari feladat-instanciát, melynek legfontosabb jellemző méretei – az üzleti titoknak minősülő konkrét adatok nélkül – tájékoztató jelleggel a 7. táblázatban láthatók.

A keresési algoritmus paraméterei a következők:

- A lépések maximális száma: 3000.
- Adott lépésben generált kiterjesztett megoldások maximális száma: 150.
- A tabulista maximális elemszáma: 150.
- A POOL-ban tárolt legjobb megoldások száma (ismételt futások száma): 1.

7. táblázat: Az IPSIC feladat egy valós ipari példája (Fehrer_work_2015_1_25_ext_PO.sif).

Ütemezési időhorizont	2 hét
A műszakok időtartama	8 óra
Gyártósorok száma	4
Pozíciók száma	88
Formahordozók száma	144
Formák száma	418
Terméktípusok száma	182
Termékcsoportok száma	129
Megrendelések száma	856

Az alkalmazott PC-1 futási tesztkörnyezet fontosabb jellemezői a következők:

- Intel Core i7 M620 2,67 GHz CPU,
- 4 GB RAM,
- Microsoft Windows 7 Home (64 bit) OS.

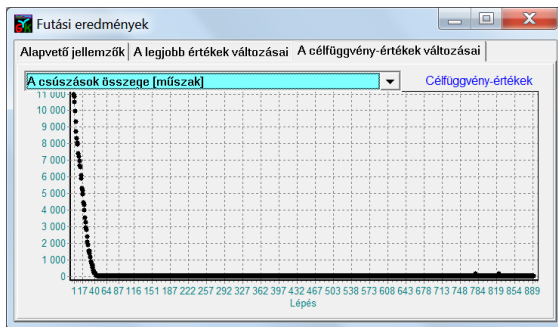
A továbbiakban egy jellemző futtatási esetet mutatok be részletesen. A kiválasztott esetben az algoritmus futási ideje 1 perc 26 másodperc volt. A keresési folyamat induló és befejező állapotához tartozó megoldások célfüggvény-értékeit a 8. táblázat foglalja össze.

8. táblázat: A keresési folyamat induló és befejező állapotához tartozó célfüggvény-értékek.

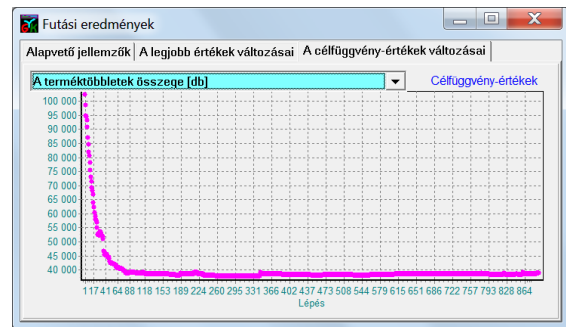
<i>Célfüggvény</i>	<i>Prioritás-érték</i>	<i>Induló állapot</i>	<i>A keresés eredménye</i>
f_1	42	5445	0
f_2	44	57457	0
f_3	44	307	0
f_4	2	0	103
f_5	2	0	3
f_6	3	74	71
f_7	2	105281	38316
f_8	12	5545	135
f_9	8	60170	652
f_{10}	1	307	0
f_{11}	1	1	0
f_{12}	1	55	0
f_{13}	1	5445	0
f_{14}	1	41813	0
f_{15}	50	42	0
f_{16}	50	11092	0
f_{17}	1	0	51
f_{18}	40	0	0

Az I. mellékletben található 61.-78. ábrák a 18 célfüggvény értékeinek változását mutatják az *EasyForce* keresés során megtett lépések függvényében. A diagramok az adott lépésen belül megvizsgált kiterjesztett ütemtervek részleteit nem szemléltetik, az adott lépéshez a kiterjesztés legjobb megoldásának értékeit rendeli hozzá.

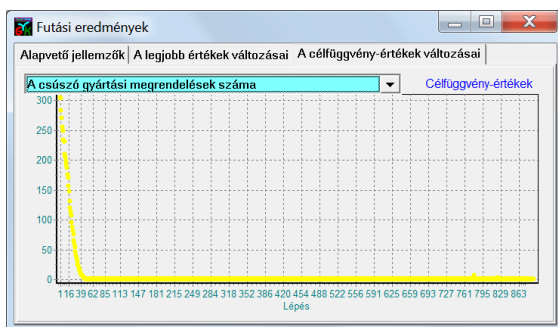
Terjedelmi okokból itt csak hat célfüggvény diagramját helyeztem el (37.- 42. ábra). A legjobb megoldást eredményező lépés sorszáma: 759, ezen belül a legjobb kiterjesztés sorszáma: 35.



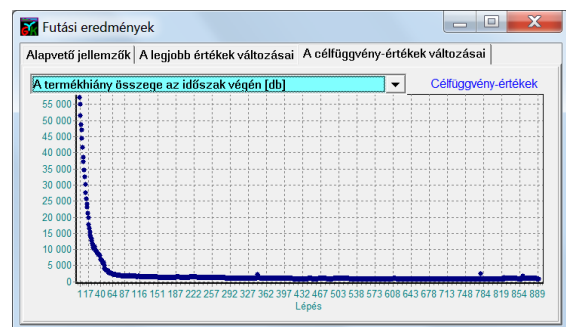
37. ábra: (f_{16}) A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak].



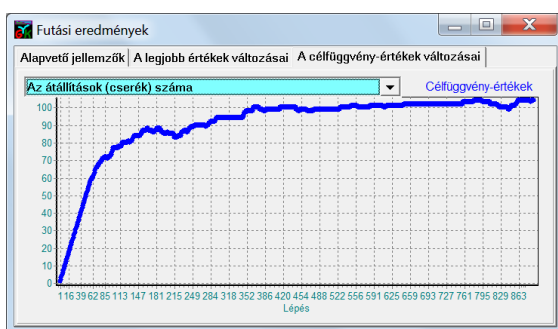
40. ábra: (f_7) A terméktöbbletek összege az időszak végén [db].



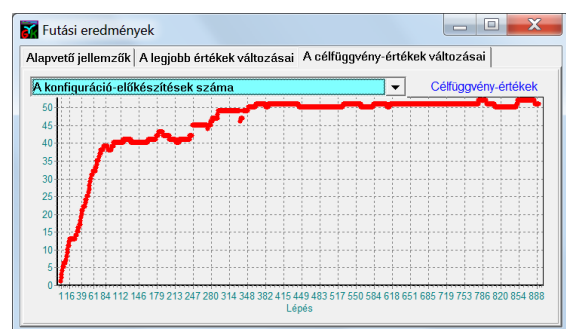
38. ábra: (f_3) A csúszó (határidőt túllépő) rendelés-teljesítések száma.



41. ábra: (f_9) A termékhiány összege az ütemezési időszak végén [db].



39. ábra: (f_4) A konfiguráció-cserék száma.



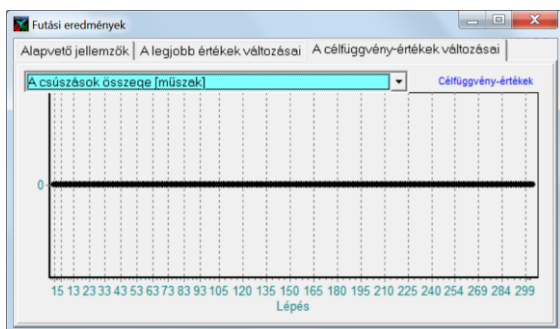
42. ábra: (f_{17}) A konfiguráció-előkészítési munkák száma.

Az *EasyForce* algoritmus a beállított célfüggvény-prioritásoknak megfelelően a keresési lépésekben olyan megoldásokat választ ki és fejleszt tovább, amelyek elsősorban a rendelés-teljesítések csúszásait, a csúszó megrendelések számát csökkentik drasztikusan (37.-38. ábra), ugyanakkor a készlet-szinteket az előírt sávba terelik azáltal, hogy a termék-többleteket és a termék-hiányokat csökkentik (40.-41. ábra). Ezek a célfüggvények egyidejűleg javulnak a megoldási folyamatban, viszont a konfiguráció-cserék (39. ábra) és a konfiguráció-előkészítési

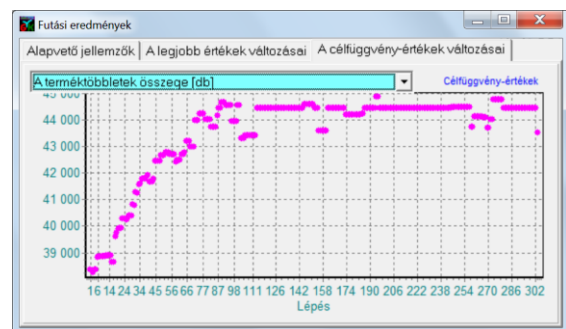
tevékenységek száma (42. ábra) kezdetben drasztikusan, majd egyre kisebb mértékben emelkedik. A végeredményben 103 konfiguráció-csere szerepel, melyből 51 igényel konfiguráció-előkészítést, 52 csere során már korábban használt konfiguráció újra felhelyezése valósul meg.

9. táblázat: Példák különböző prioritási sémákra (objf_pri.ofp).

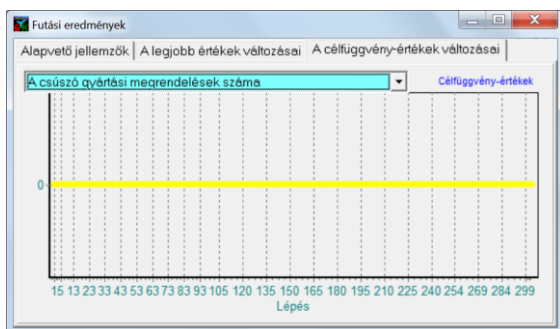
Séma neve	Prioritás-értékek
	$f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8 f_9 f_{10} f_{11} f_{12} f_{13} f_{14} f_{15} f_{16} f_{17} f_{18} $
Agile_Control	42 44 44 2 2 3 2 12 8 1 1 1 1 50 50 1 40
PO_Tardiness	0 0 0 1 1 0 0 0 0 0 0 0 0 50 50 0 0
Setup_for_PO	40 40 40 10 10 0 0 0 0 2 2 1 1 50 50 6 48
Stock	40 40 40 1 1 1 1 50 50 30 18 1 50 50 50 50 1 40
Num_of_Tardy_PO	0 0 50 2 2 0 0 0 0 50 0 0 0 0 0 0 0
Product_Shortage	50 50 0 2 2 0 0 0 0 0 0 0 0 0 0 0 0



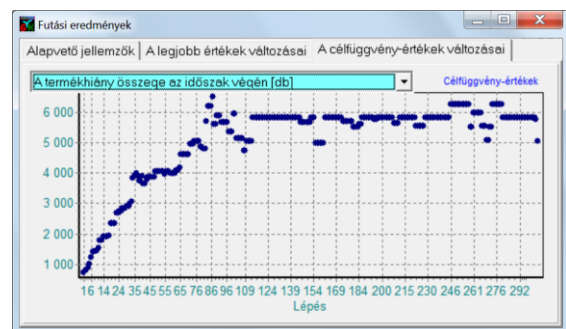
43. ábra: (f_{16}) A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak].



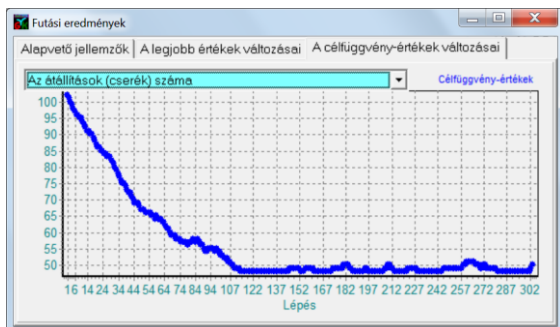
46. ábra: (f_7) A terméktöbbletek összege az időszak végén [db].



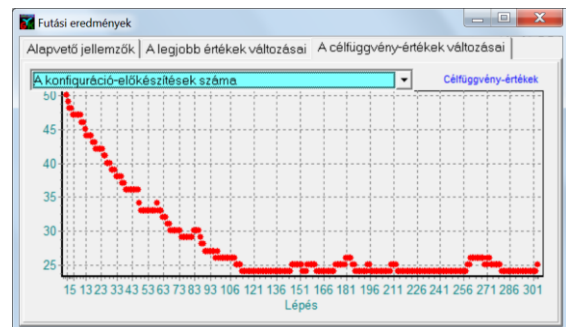
44. ábra: (f_3) A csúszó (határidőt túllépő) rendelés-teljesítések száma.



47. ábra: (f_9) A termékhány összege az ütemezési időszak végén [db].



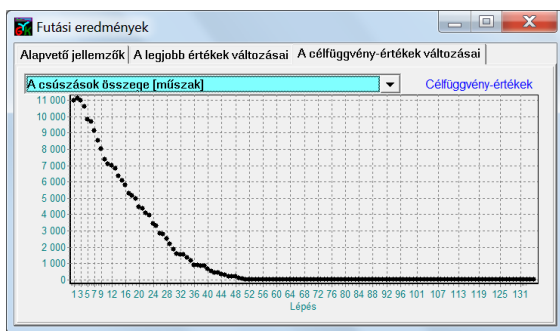
45. ábra: (f_4) A konfiguráció-cserék száma.



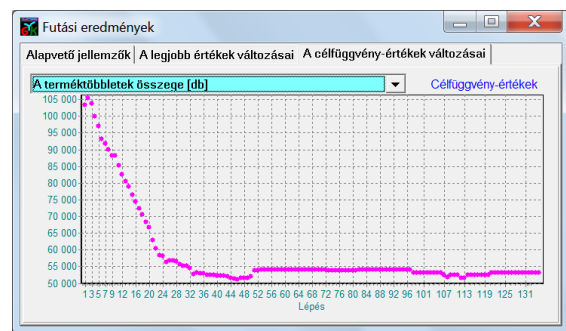
48. ábra: (f_{17}) A konfiguráció-előkészítési munkák száma.

Az *EasyForce* algoritmus rugalmasságának köszönhetően a megtalált legjobb megoldással inicializálva újraindítható a keresés, és olyan célfüggvény-prioritási séma is használható (9. táblázat: PO_Tardiness), melynek segítségével a rendelés-teljesítéseket szinten tartva (43.- 44. ábra) a konfiguráció-cserék száma csökkenthető (45. és 48. ábra) a készletszintek rovására (46.- 47. ábra). A rendelések teljesítéséhez elegendő 48 konfiguráció-csere, melyből 24 igényel konfiguráció-előkészítést.

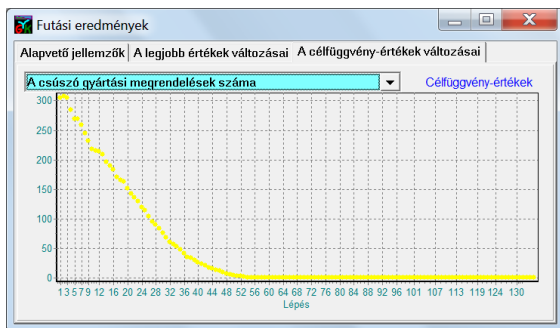
Ez a bemutatott több szakaszból álló ütemezési funkció, arra is alkalmassá teszi az ütemező rendszert, hogy ha olyan feladattal kerül szembe, amely az adott szigorú korlátozások betartása mellett nem oldható meg csúszások nélkül, akkor a konfiguráció-cserékre és előkészítésekre vonatkozó szigorú korlátozásokat fellazítva kereshető elfogadható megoldás, majd az elért megoldás csúszásokkal kapcsolatos célfüggvény-értékeit nem rontva kereshető tovább olyan módosított megoldás, amely a konfiguráció-cserék és előkészítések szempontjából kedvezőbb.



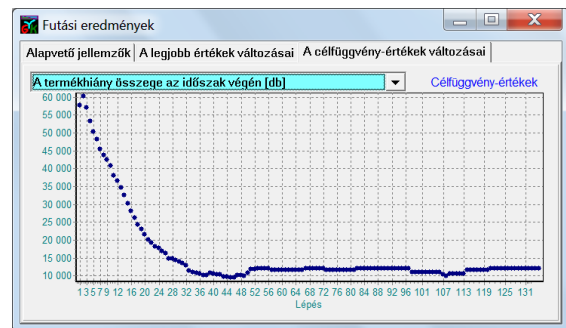
49. ábra: (f_{16}) A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak].



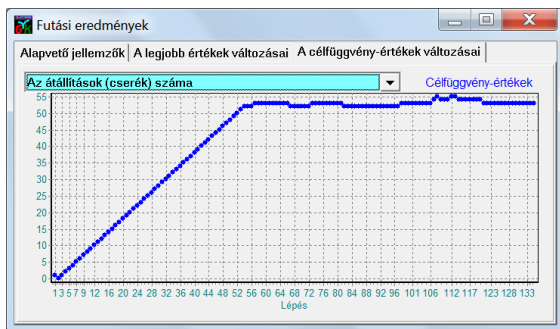
52. ábra: (f_7) A termék többletek összege az időszak végén [db].



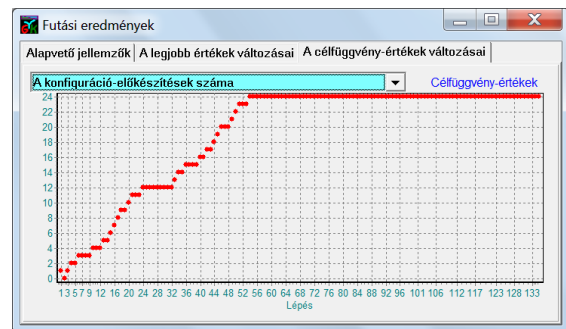
50. ábra: (f_3) A csúszó (határidőt túllépő) rendelés-teljesítések száma.



53. ábra: (f_9) A termékhiány összege az ütemezési időszak végén [db].

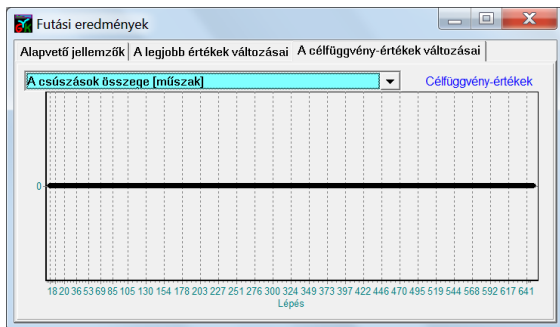


51. ábra: (f_4) A konfiguráció-cserék száma.

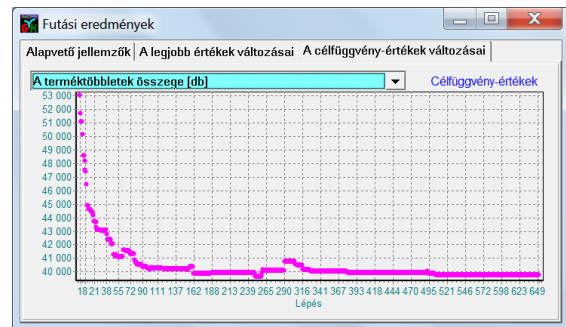


54. ábra: (f_{17}) A konfiguráció-előkészítési munkák száma.

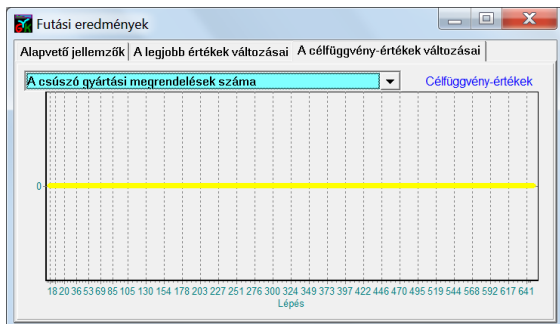
Arra is lehetőség van, hogy olyan prioritási sémával kezdje az *EasyForce* algoritmus a feladat megoldását, amely a konfiguráció-cserékhez és konfiguráció-előkészítésekhez, valamint a rendelések csúszásaihoz viszonylag magas prioritásokat társít, míg a készletszintekhez alacsony prioritást rendel (9. táblázat: Setup_for_PO). Ezáltal a keresés során csak addig tervez be új cseréket az ütemtervbe, amíg azok a megrendelések teljesítését szolgálják (49.-54. ábra). Ezzel a stratégiával is elérhető az a határ, amely fölött a cserék száma már csak készletezési célokat szolgál. Az így megtalált megoldásból kiindulva is folytatható más sémával (9. táblázat: Stock) a keresés, melynek során a csúszások szinten tartása mellett a sávosan előírt készletszintek elérésére törekszik az algoritmus (55.- 60. ábra).



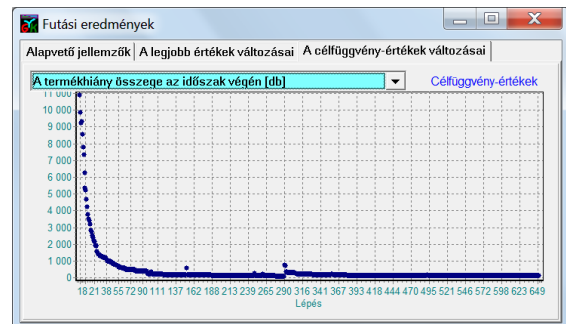
55. ábra: (f_{16}) A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak].



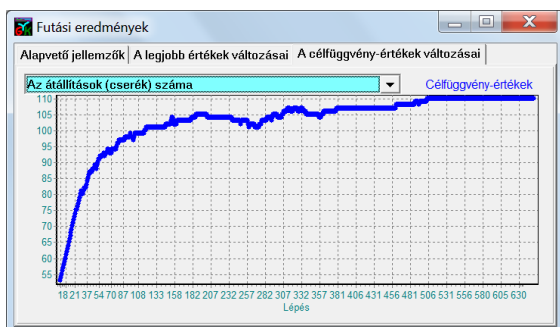
58. ábra: (f_7) A terméktöbbletek összege az időszak végén [db].



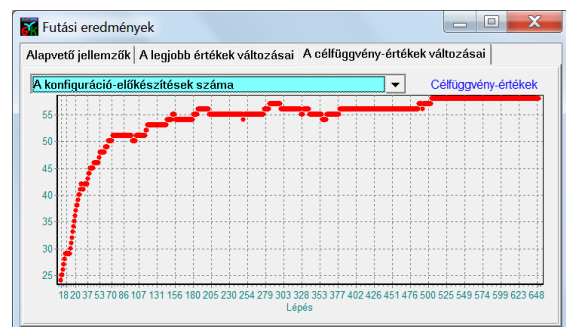
56. ábra: (f_3) A csúszó (határidőt túllépő) rendelés-teljesítések száma.



59. ábra: (f_9) A termékhiány összege az ütemezési időszak végén [db].



57. ábra: (f_4) A konfiguráció-cserék száma.



60. ábra: (f_{17}) A konfiguráció-előkészítési munkák száma.

A mindenkori célfüggvény-rendszerben rejlő, bizonyos határokon túl már csak egymás rovására javítható (trade-off) elemek kezelését a prioritások megfelelő kiosztásával lehet

megvalósítani. A fenti példában bemutatott *előre-vissza* szakaszból álló, valamint *előre-előre* szakaszból álló megoldási folyamat általános esetben tetszőleges számú szakaszból is összeállítható eltérő prioritási sémákat használva. A rendszer lehetővé teszi bármely lépésben a keresési folyamat paramétereinek konfigurálását, a célfüggvények prioritásainak módosítását, az eredmények folyamatos nyomkövetését, mentését és visszatöltését, továbbá a folyamat újraindítását bármely mentett állapotból. Ezek a funkciók a valós ipari környezetre ráhangolás eszközeként használhatók. Az elmentett sémák később a mindennapos feladatmegoldásokban betölthetők és alkalmazhatók.

A választott célfüggvények halmazán belül vannak olyan részhalmazok, amelyek elemei – szituációtól függő esetekben – azonos szempont szerint együtt hatékonyabban tudják a megoldást javítani, mint külön-külön. Máskor viszont egymással is konfliktusba kerülhetnek. Ilyen például a legnagyobb csúszás, a csúszó megrendelések száma és a rendelés teljesítésekor hiányzó termékmennyiség (II. Melléklet). Ha a feladat megoldható úgy, hogy minden egyes rendelés határidőre teljesül, akkor együtt használva ezeket a célfüggvényeket könnyebb a feladatot megoldani. Ha viszont a feladat nem oldható meg határidő-túllépés nélkül, akkor viszont a három említett célfüggvény más-más irányba próbálja terelni a keresési folyamatot. Ilyenkor a kiosztott prioritások döntenek el, hogy melyik szempont érvényesül. Kevesebb számú rendelés esetében legyenek viszonylag nagyobb csúszások, vagy több rendelés esetében legyenek kisebb mértékű csúszások.

Ehhez hasonló belső viszonyokkal jellemezhető részhalmazt alkotnak a készletszintekhez kapcsolódó célfüggvények is. A hatékony cserék kiválasztását együtt hatékonyabban támogatják, mint külön-külön. Azonban a rendelések teljesítését követően, ha a fennmaradó konfiguráció-cserélési és előkészítési lehetőségekkel nem biztosítható minden egyes terméktípus esetében az, hogy a zárókészlet a saját sávjába kerüljön, akkor a sáv feletti többletek és a sáv alatti hiányok, valamint az érintett terméktípusok száma szintén „trade-off” viszonyba kerül. Ilyenkor a felhasználó elvárásai szintén leképezhetők a prioritások megfelelő kiosztásával.

A célfüggvény-rendszert alkotó halmazok egymással is ilyen kettős viszonyban állnak. Például a keresési folyamat kezdeti szakaszában a jó cserét az jellemzi, hogy egyrészt a lehető legnagyobb mértékben javítja a rendelések teljesítését azáltal, hogy olyan konfigurációt tesz fel a gyártósorra, amelyik a rendelt terméktípusokat és a hiánnyal küzdő terméktípusokat fogja gyártani, ráadásul a megfelelő intervallumokban teszi ezt, másrészt egyúttal olyan konfigurációt vesz le, amelyik egyébként csak többletet gyártana az adott időszakban. A keresési folyamat előrehaladása során az ilyen *minden szempontból előnyös* cserék lehetősége folyamatosan csökken. Így a rendelés-teljesítésre orientált csere és a készletre orientált csere kompromisszumos döntésekké alakul át.

A különböző összetételű és nehézségi fokú tesztfeladatok vizsgálata során azt tapasztaltam, hogy a gyakorlatban fellépő nagyméretű és változó összetételű feladatok megoldását az *EasyForce* kereső algoritmus néhány perc alatt képes előállítani. A vizsgált legbonyolultabb feladatok és a leghosszabb időintervallumok esetében sem haladta meg a futási idő a 20 percet.

A saját tesztjeim mellett a cég éles ipari feladatokon valós környezetben tesztelte a fejlesztés különböző fázisaiban az ütemező rendszer aktuális változatait. A fejlesztés során a gyakorlati visszajelzéseket figyelembe véve folyamatosan bővítettem és finomítottam a rendszert. A befejező éles tesztek eredményeiből és tapasztalataiból azt a következtetést fogalmaztam meg,

hogy a bemutatott modellek és megoldási módszerek nagyméretű és bonyolult feladatok esetében is hatékonyan alkalmazhatók, rugalmasan alkalmazkodnak a felhasználó változó elvárásaihoz és rövid időn belül szolgáltatják az eredményeket. A kutatás-fejlesztés kitűzött céljai teljes mértékben teljesültek.

4.5 Összefoglalás és következtetések

Az értekezés 4. fejezete a termelésütemezés és a készletgazdálkodás kombinált új termelésprogramozási (IPSIC) feladatának – és példaként egy járműipari alkatrészgyártás ilyen típusú termelésprogramozási feladatának – modellezését és konkrét megoldását foglalja össze. A feladat speciálisan az igény szerinti rugalmas gyártás proaktív termelésütemezésére és készletgazdálkodására vonatkozik, amely magába foglalja az elvégzendő tevékenységek definiálását, a végrehajtás ütemezését és az osztott hozzáférésű erőforrások menedzselését, valamint a terméktípusok készletszintjének irányított kezelését. A vizsgált erőforrás-korlátos gyártórendszer legfontosabb jellemzői közé tartoznak az osztott hozzáférésű szerszámok és készülékek, a korlátozottan rendelkezésre álló gyártósorok és kiszolgáló személyzet, az eltérő gyártási intenzitások, a szerszám-előkészítési tevékenységek valamint a szigorú határidős rendelések.

A konkrét ipari problémák megoldása érdekében két irányból közelítettem meg a feladatot. Egyrészt a valós folyamatok elemzését követően a konkrét ütemezési feladatokat modelleztem, másrészt a szakirodalomban elérhető modellek vizsgálatát követően, kiindulási alapmodelleket választottam és a konkrét ipari problémák sajátosságainak megfelelően továbbfejlesztettem. A kiterjesztett modellek és a valós problémák összekapcsolására a problémater transzformációjára alapozott szemantikai megfeleltetés módszerét alkalmaztam.

Az integrált termelésprogramozási problémát keresési feladatként fogalmaztam meg, kidolgoztam a probléma optimalizálási modelljét, melyben definiáltam a bemenő alapadatokat, a döntési változókat, az előírt korlátozásokat és a megfogalmazott célokat kifejező célfüggvényeket.

A részletes termelési finomprogramok készítésére kifejlesztettem egy új kombinált megoldási módszert, amely egy kétszintű döntési hierarchiára épül. A felső szint az IPSIC feladat igényeinek megfelelően továbbfejlesztett többoperátoros és többcélú keresési algoritmust használ, amelyhez az alsó szinten egy egzakt felépítő algoritmus illeszkedik. Az új koncepciót megvalósító *EasyForce* módszer rugalmasságának köszönhetően képes alkalmazkodni a változó feltételekhez és célfüggvény-rendszerekhez.

A bemutatott új termelésprogramozási modell fontos jellemzője, hogy képes integráltan kezelni a gyártási főfolyamatokat, valamint az ahhoz közvetlenül kapcsolódó előkészítési és készletgazdálkodási feladatokat. A gyakorlatban (valós körülmények között) elvégzett elemzések azt mutatták, hogy a kidolgozott elméleti modellek és módszerek alapján implementált termelésprogramozó szoftver képes automatikusan elkészíteni a feladat megoldását jelentő részletes termelési finomprogramot. Az elvégzett vizsgálatok eredményei alapján igazoltam, hogy a bemutatott megoldási módszer alkalmas IPSIC típusú valós ipari feladatok megoldására.

Az új integrált megoldási koncepció egyaránt támogatja a prediktív ütemezési és a reaktív újraütemezési stratégiák gyakorlati alkalmazását. Alapértelmezés szerint a termelésprogramozó rendszer bizonyos időközönként rendszeresen, adott időszakra előre az előző ütemezett időszakokkal részleges átfedésben (pl.: minden hét elején kéthetes időintervallumra) előállítja a tervezett finomprogramot. Közbenső reaktív újraütemezésre akkor van szükség, ha valamilyen előre nem tervezett esemény vagy események bekövetkezése (pl. erőforrások kiesése, rendelések változása stb.) miatt szükségessé válik a végrehajtás alatt álló finomprogram módosítása. Az újraütemezés során a megváltozott szituációból kiindulva készül el az új megoldás. Az aktuális állapot válik a kiindulási állapottá és az implementált eljárás növekményes rendszerben újragenerálja a termelési finomprogramot, amely magába foglalja a megváltozott helyzethez igazodó konfiguráció-cserék és konfiguráció-előkészítések ütemtervét.

Az általam kidolgozott megoldási koncepció fontos jellemzője, hogy proaktív (előretekintő) szemléletet is magába foglal. Ez azt jelenti, hogy a termelésprogramozó rendszer a vevői rendelések határidőre történő teljesítése mellett arra is figyel, hogy az ütemezett időszak végén a gyártórendszer olyan állapotot vegyen fel, amely már a következő időszakra való rákészülést is biztosítja. Ezért olyan konfiguráció-cseréket is betervez, amelyek hatására a készletszintek a terméktípusonként külön-külön előírt optimum közelébe (az előírt tartományokba) kerülnek, emellett arra is törekszik, hogy a szükséges konfiguráció-előkészítések száma és a konfiguráció-cserék száma minimális legyen. Ez a proaktív szemlélet a termelésstervezési és ütemezési stratégiákban előszeretettel használt klasszikus „gördülő” tervezési technika egyfajta továbbfejlesztett és kiterjesztett változatának is tekinthető. A konfiguráció-előkészítési tevékenységek időbeli tervezésére egy paraméterezzhető robusztus ütemezési stratégiát alkalmazok. Ennek lényege, hogy a felhasználó biztonsági tartalékidő megadásával paraméterezzheti az ütemezés „élességét”. Választhat tisztán tartalék-idő maximalizálási stratégiát és választhat „éppen időben” (Just-In-Time) szemléletű stratégiát. E két véglet közötti kompromisszumos megoldások is elérhetők a vezérlő paramétereknek köszönhetően.

A bemutatott IPSIC modell és a megoldási módszerei alkalmasak a konkrét járműipari alkatrészgyártáson túl másféle gyártástechnológiát alkalmazó műhelyek vagy üzemek termelési folyamatainak finomprogramozására is. A modell entitásai (pl. gyártósor, pozíció, szerszám, formahordozó, szakmunkás stb.) leképezhetők más típusú de hasonló funkciót ellátó objektumokra, és ezáltal például a gyártósor és annak pozíciói együtt egy általános (pl. fröccsöntő, forgácsoló vagy összeszerelő stb.) gépet/gépsort reprezentálhat, míg a formák és formahordozók megfeleltethetők tetszőleges (pl. fröccsöntő vagy forgácsoló) szerszámnak és szerszámbefogónak. Amennyiben nincs szükség az előkészítési folyamatok kapacitáskorlátainak érvényesítésére, akkor az kellően nagy fiktív szabadkapacitás beállításával kikapcsolható. A készletszintek és a rendelések kezelése pedig technológiától függetlenül alkalmazható tetszőleges termelési rendszerben.

5 Új tudományos eredmények

1. tézis: **Műszakbeosztások igény-vezérelt tervezési módszere termelési finomprogramokban.**

Kidolgoztam egy igény-vezérelt műszakbeosztási modellt az emberi erőforrások és gépi erőforrások szinkronizálására, amely az erőforrások rendelkezésre állási időintervallumait döntési változóként és/vagy korlátfeltételként képes kezelni a termelési finomprogramozási feladatokban. Új módszert dolgoztam ki a fenti finomprogram-tervezési feladat megoldására a diszkrét gyártó-szerelő rendszerek ütemezési és műszakbeosztási feladatainak integrált megoldásával.

Az integrált tervezési módszer fontosabb jellemzői:

- 1.1. A műszakbeosztásra vonatkozó modell-kiterjesztés tetszőleges típusú termelésütemezési feladat esetében alkalmazható.
- 1.2. A szinkronizált feladatmegoldás kétfázisú: Az első fázisban az erőforrásokhoz önállóan konstansként definiált időintervallumok rendelhetők (korlátfeltételek). A második fázisban az ütemezési feladat megoldása során kialakuló aktuális terhelések alapján véglegesíthető az aktív műszakok kiosztása (döntési változók).
- 1.3. A műszakok költségének minimalizálása érdekében kidolgoztam egy új „relatív műszakköltség” szemléletű célfüggvény koncepcióját.
- 1.4. A kidolgozott módszerrel az ütemezési és a műszakbeosztási feladat egyidejűleg megoldható egy bővített többoperátoros és többcélú keresési technikával (MOMOTS, Multi-Operator and Multi-Objective Tabu Search). A kiindulási (alap) ütemezési feladat választott kereső algoritmusának módosító operátora (vagy operátorai) kiegészítendő a műszakbeosztást módosító operátorokkal, továbbá az eredeti célfüggvények a műszakbeosztásra vonatkozó célfüggvényekkel.
- 1.5. Az integrált megoldási módszer – a fentiekén túlmenően – támogatja az alternatív technológiai útvonal változatok menedzselését, az anyagmozgatási idők és korlátozott méretű műveletközi tárolók figyelembe vételét, valamint a rendelési, gyártási és logisztikai sorozatnagyságok megkülönböztetését és kezelését.
- 1.6. Egy nagyméretű, valós ipari feladat megoldásán keresztül igazoltam, hogy az új koncepció alkalmas diszkrét alkatrészgyártási és szerelési folyamatok termelésütemezési és műszakbeosztási feladatainak integrált megoldására.

Az 1. tézisben összefoglalt eredményeket az értekezés 3. fejezetében részletesen ismerttettem, valamint az [s4], [s3], [s5], [s7], [s9], [s10], [s11], [s12], [s14], [s15], [s17], [s16] publikációkban is bemutatam.

2. tézis: Proaktív szemléletű modell és kombinált ütemezési módszer a „lehívásos” jellegű alkatrészgyártó rendszerek finomprogramozási feladatainak megoldására.

Új modellt és megoldási módszert dolgoztam ki a megrendelésre és raktárra gyártás együttes ütemezésére, amely alkalmas a beszállítói szerződésekből rögzített cikkenként eltérő minimális készletszintek fenntartására és a szigorú határidős direkt rendelések (lehívások) egyidejű kiszolgálására. A termelésütemezés és az igény-vezérelt készletgazdálkodás integrációján túlmenően a javasolt modellezési szemlélet lehetővé teszi a gyártási főfolyamatok ütemezésének és a kapcsolódó előkészítési tevékenységek ütemezésének együttes megoldását is.

A modell fontosabb jellemzői:

- 2.1 A kidolgozott IPSIC modell alapvetően az autóiipari üléselemek gyártási jellemzői és követelményei alapján készült (párhuzamosan működő körpályás gyártások modellje), de más gyártástechnológiai környezetre is gyorsan adaptálható.
- 2.2 A modell együtt (integráltan) kezeli a határidős gyártási rendelések kiszolgálását és a cikkszámoként előírt készletszint-optimalizálást.
- 2.3 A modell figyelembe veszi az erőforrások sajátosságait és korlátozott rendelkezésre állásait.
- 2.4 Megvalósítja a gyártósorok szinkronizált átállítási (set-up) feladatainak ütemezését, figyelembe véve a gyártáselőkészítési feladatokat is.
- 2.5 Támogatja a gyártási feladatok elvégzéséhez szükséges osztott hozzáférésű szerszámok és készülékek konfigurálásának igény-vezérelt menedzselését az aktuális technológiai előírások és alternatívák szerint.

A 2. tézisben összefoglalt eredményeket az értekezés 4.1., 4.2., és 4.4. alfejezeteiben részletesen ismertettem, valamint az [s1], [s2], [s6], [s13] publikációkban is bemutattam.

3. tézis: Új modell és megoldási módszer egyoperációs, időben változó számú erőforrásos ütemezési feladatok megoldására.

Az egyoperációs határidős munkák ütemezésére kidolgoztam a $P(s) | p_i=1; r_i=\text{integer}; d_i=\text{integer} | L_{max}$ formalizmussal definiált modellt és annak egzakt megoldó algoritmusát. A modell legfontosabb újonsága, hogy a párhuzamosan működő erőforrások száma függhet az időtől. Kidolgoztam egy továbbfejlesztett JIT-orientált algoritmust, amely biztonsági vezérlő paramétereket használva a határidők betartása mellett csökkenteni a munkák túl korai befejezését.

A modell és megoldási koncepció fontosabb jellemzői:

- 3.1 Egy valós ipari feladat által motivált igény alapján kidolgoztam egy új általánosított modellt, amely a szakirodalomban megtalálható $P \mid p_i=1; r_i=integer; d_i=integer \mid L_{max}$ modell kiterjesztésének és általánosításának tekinthető.
- 3.2 Az új modellel leírható feladatok megoldására polinomiális futási idejű egzakt megoldó algoritmust dolgoztam ki: Tartalékidő-orientált algoritmus (Slack-Oriented Algorithm).
- 3.3 A külső és belső forrásból érkező váratlan események káros hatásának csökkentése érdekében kidolgoztam egy továbbfejlesztett JIT-orientált algoritmust, amely biztonsági vezérlő paramétereket használva a határidők betartása mellett csökkenti a munkák túl korai befejezését.
- 3.4 A modell és a megoldási módszerek gyakorlati alkalmazása érdekében kidolgoztam egy speciális probléma-transzformációs eljárást, amely a formális modell és egy valós műhelyszintű ütemezési feladattípus egyértelmű megfeleltetését biztosítja.

A 3. tézisben összefoglalt eredményeket az értekezés 4.3. alfejezetében részletesen ismertettem, valamint az [s1], [s2], [s6], [s13] publikációkban is bemutattam.

6 Az eredmények hasznosítása

Napjainkban a termelési rendszerek és folyamatok fejlesztésében, az ipari informatikai infrastruktúrák tervezésében, létrehozásában és működtetésében egyre fontosabb szerepet játszanak a *kiber-fizikai gyártórendszerek (Cyber-Physical Production Systems)*, az *Ipar 4.0 (Industry 4.0)*, a *dolgok Internete (Internet of Things, IoT)*, a *felhőalapú számítás (Cloud Computing)*, és a *nagy adatmennyiség (Big Data)* paradigmák és technológiák. A rohamosan fejlődő informatikai és kommunikációs környezetben megnövekszik az igény az új technológiák által nyújtott szolgáltatások üzleti és automatizálási célú kiaknázására, többek között például integrált optimalizálási, innovatív döntéstámogató modellek és módszerek kidolgozására, valamint implementált szoftvereken keresztül ezek gyakorlati alkalmazására.

Az általam kidolgozott új modellek hozzájárulnak ehhez a fejlődéshez, mert a felsorolt technológiák lehetővé teszik a megfelelő bemenő adatok, a számítási kapacitás, az alkalmazási környezet biztosítását, ezáltal megvalósítható a modellek integrálása a vállalatok hagyományos és korszerű termelésinformatikai rendszerébe egyaránt.

Az értekezésben összefoglalt eredmények a diszkrét termelési rendszerekben folyó igény szerinti rugalmas gyártás és szerelés területein hasznosíthatók termelésstervezési, ütemezési, finomprogramozási és integrált döntéstámogató feladatok megoldására.

A tézisek érvényességét – a mesterségesen létrehozott adatokon végzett futtatási tesztek kedvező tapasztalatait követően – valós problémák ipari környezetben történő megoldásával igazoltam. Ehhez konkrét termelési rendszerek műszaki specifikációi és ipari adatok kerültek felhasználásra ipari partnerek bevonásával (Electrolux Lehel Hűtőgépgyár, Jászberény; Fehrer Hungaria Járműipari Kft., Mór).

A kapcsolódó K+F+I projektek tapasztalatait is felhasználva, a tudományos eredményeket az irodalomjegyzékben felsorolt publikációk formájában konferenciákon mutattam be és folyóiratokban is publikáltam.

Az új modelleket bemutató részek végén – a 3.5., a 4.3.8., és a 4.5. alfejezetekben – összefoglaltam a kapcsolódó eredmények alkalmazási lehetőségeit.

A kidolgozott új modellek és algoritmusok, valamint a kapcsolódó ütemező szoftverek bekerültek a Miskolci Egyetemen folyó mérnökinformatikus és gazdaságinformatikus képzés termelésinformatikai specializációjának oktatásába az Alkalmazott Informatikai Intézeti Tanszék által gondozott „A termelésinformatikai alapjai”, a „Diszkrét termelési folyamatok számítógépes tervezése és irányítása”, és a „Virtuális vállalat” c. tantárgyak tananyagán keresztül.

Az eredmények ipari alkalmazási példák formájában részét képezik a Miskolci Egyetem online kurzusait menedzselő „MeMOOC Studio” szabad hozzáférésű (open online) keretrendszerben elérhető „Ütemezési modellek és algoritmusok” c. magyar nyelvű és „Scheduling Models and Algorithms” c. angol nyelvű kurzusnak is.

Saját publikációk

- [s1] Kulcsárné Forrai, M., Kulcsár, Gy.: Modeling and Solving an Extended Parallel Resource Scheduling Problem in the Automotive Industry, *Acta Polytechnica Hungarica*, ISSN 1785-8860 (Paper Number: 3102, accepted).
- [s2] Kulcsárné Forrai, M., Kulcsár, Gy.: A New Scheduling Software for Supporting Automotive Component Manufacturing, *Lecture Notes in Mechanical Engineering*, ISSN 2195-4356, Vol. 1, pp. 257-274, 2017.
- [s3] Kulcsár, Gy., Kulcsárné Forrai, M.: Detailed Production Scheduling Based on Multi-Objective Search and Simulation, *Production Systems and Information Engineering*, ISSN 1785-1270, Vol. 6, pp. 41-56, 2013.
- [s4] Kulcsár, Gy., Kulcsárné Forrai, M.: Solving Multi-Objective Production Scheduling Problems Using a New Approach, *Production Systems and Information Engineering*, ISSN 1785-1270, Vol. 5, pp. 81-94, 2009.
- [s5] Bikfalvi, P., Erdélyi, F., Kulcsár, Gy., Tóth, T., Kulcsárné Forrai, M.: On Some Functions of the MES Applications Supporting Production Operations Management, In: Bognár, G., Tóth, T., (eds.) *Applied Information Science, Engineering and Technology: Selected Topics from the Field of Production Information Engineering and IT for Manufacturing: Theory and Practice*, (Topics in Intelligent Engineering and Informatics; 7.), ISBN:978-3-319-01918-5. Berlin: Springer-Verlag, pp. 103-129, 2014.
- [s6] Kulcsárné Forrai, M., Kulcsár, Gy., Új modell időben változó erőforrás-korlátos ütemezési feladatok megoldására, *Műszaki Tudományos Közlemények / Papers on Technical Science*, 3, ISSN 2393 – 1280, pp. 195-198, 2015.
- [s7] Kulcsár, Gy., Kulcsárné Forrai, M.: Kiterjesztett termelésprogramozási modell erőforrás-korlátos ütemezési feladatok megoldására, *Multidiszciplináris tudományok*, Vol. 4 (1), HU ISSN 2062-9737, pp. 19-30, 2014.

- [s8] Hadházy, P., Kulcsár, Gy., Kulcsárné Forrai, M., Magyar, D.: A FlexiForce Hungary Kft. termelésstervezési feladatának modellezése és heurisztikus megoldása, *Multidiszciplináris tudományok*, Vol. 3 (1-2), HU ISSN 2062-9737, pp. 305-316, 2013.
- [s9] Tóth, T., Kulcsár, Gy., Erdélyi, F., Kulcsárné Forrai, M., Bikfalvi, P.: Extended Models for Supporting CAPP-MES Integration in Discrete Production Systems, *Proceedings of The 16th International Conference on Machine Design and Production: UMTIK*, June 30 – July 03 2014, İzmir, Turkey, (ISBN:978-975-429-332-6 (1.c)), pp. 555-570, 2014.
- [s10] Tóth, T., Kulcsár, Gy., Erdélyi, F., Kulcsárné Forrai, M., Bikfalvi, P.: The Role of CAPP-MES Integration in the Cyber-Physical Production Systems Paradigm, *Proceedings of the Tenth International Symposium on Tools and Methods of Competitive Engineering: TMCE 2014*, May 19-23, Budapest, Hungary, (ISBN:9789461861764; 9789461861771), pp. 1381-1392, 2014.
- [s11] Kulcsárné Forrai, M., Erdélyi, F., Kulcsár, Gy.: A New Extended Model for Solving Flexible Job Shop Scheduling Problems, *Proceedings of the International Conference on Innovative Technologies, IN-TECH 2013*, (ISBN:978-953-6326-88-4), Budapest, Hungary, 2013.09.10-2013.09.12. pp. 325-328, 2013.
- [s12] Kulcsár, Gy., Kulcsárné Forrai, M.: Detailed Scheduling Based on Simulation, *Proceedings of the XXVI. microCAD International Scientific Conference*, Miskolc, Hungary, 29th – 30th March, 2012, p. 6. (CD), 2012.
- [s13] Kulcsár, Gy., Kulcsárné Forrai, M.: Járműipari alkatrészgyártás termelésprogramozási feladatainak modellezése és megoldása, *Műszaki Tudomány az Észak-kelet Magyarországi Régióban 2016 konferencia*, Miskolc, 2016, május 25. ISBN 978-963-7064-33-3, pp. 278-291. 2016.
- [s14] Kulcsárné Forrai, M., Kulcsár, Gy.: Termelésütemezési és műszakbeosztási feladatok integrált megoldása, *Műszaki Tudomány az Észak-kelet Magyarországi Régióban 2016 konferencia*, Miskolc, 2016, május 25. ISBN 978-963-7064-33-3, pp. 292-304, 2016.
- [s15] Kulcsár, Gy., Kulcsárné Forrai, M.: Kiterjesztett rugalmas Job Shop ütemezési feladatok megoldása többcélú keresési módszer alkalmazásával, *Műszaki Tudomány az Észak-kelet Magyarországi Régióban 2013 konferencia*, Debrecen, 2013, június 4. ISBN 978-963-7064-30-2, pp. 271-279, 2013.
- [s16] Kulcsárné Forrai, M.: Erőforrás-korlátos ütemezési feladatok szimulációval segített megoldása, In: Vadászné Dr Bognár Gabriella (szerk.) *DOKTORANDUSZOK FÓRUMA Gépészmérnöki és Informatikai Kar Szekciókiadványa*, Miskolc, p.6., 2013.
- [s17] Kulcsárné Forrai, M., Erdélyi, F.: Gyártásirányítási modellek és ütemezési célok, In: Vadászné Dr Bognár Gabriella (szerk.) *DOKTORANDUSZOK FÓRUMA Gépészmérnöki és Informatikai Kar Szekciókiadványa*, Miskolc, pp. 36-41, 2012.

Irodalomjegyzék

- [1] Allahverdi, A., Ng, C. T., Cheng, T. C. E., Kovalyov, M. Y.: A survey of scheduling problems with setup times or costs, *European Journal of Operational Research*, Vol. 187, pp. 985-1032, 2008.

- [2] Allaoui, H., Artiba, A.: Scheduling two-stage hybrid flow shop with availability constraints, *Computers and Operations Research*, Vol. 33, pp. 1399-1419, 2006.
- [3] Askin, R. G., Standridge, C. R.: *Modeling and Analysis of Manufacturing Systems*, J. Wiley Inc., New York, 1993.
- [4] Aytug, H., Lawley, M. A., Mckay, K., Mohan, S., Uzsoy, R.: Executing production schedules in the face of uncertainties: a review and some future directions, *European Journal of Operational Research*, Vol. 161, pp 86-110, 2005.
- [5] Barkmeyer E., Denno P., Feng S., Jones A., Wallace E.: NIST Response to MES Request for Information, NISTIR 6397, National Institute of Standards and Technology, pp. 1-124, 1999.
- [6] Bauer, A., Bowden, R., Browne, J. Duggan, J.: *Shop Floor Control Systems - from Design to Implementation*, Chapman & Hall, UK, 1993.
- [7] Baykasoğlu, A., Özbakir, L., Dereli, T.: Multiple dispatching rule based heuristic for multi-objective scheduling of job shops using tabu search, *Proceedings of the 5th International Conference on Managing Innovations in Manufacturing*, Milwaukee, USA, pp. 396-402, 2002.
- [8] Bikfalvi, P., Kulcsár, Gy., Erdélyi, F., Tóth, T.: Performance Analysis of some Manufacturing Systems based on Multi-Objective Approach, *Proceedings of the 15th International Conference on Machine Design and Production (UMTIK 2012)*, Pamukkale, Denizli, Turkey, 19th–22th June 2012, CD, Paper No. 55, 2012.
- [9] Buzacott, J. A., Shanthikumar, J. G.: *Stochastic Models of Manufacturing Systems*, Prentice Hall, Englewood Cliffs, NY, 1993.
- [10] Brucker, P.: *Scheduling Algorithms*, 5th ed., Springer-Verlag Berlin Heidelberg, 2007.
- [11] Chaudry, I. A., Khan, A. A.: A research survey: review of flexible job shop scheduling techniques, *International Transactions in Operational Research*, Volume 23, Issue 3, pp. 551–591, 2016.
- [12] Cheng, T. C. E., Sin, C. C. S.: A state-of-the-art review of parallel-machine scheduling research, *European Journal of Operational Research*, Vol. 47, pp. 271-292, 1990.
- [13] Csáji, B. Cs., Monostori, L.: Stochastic Reactive Production Scheduling by Multi-Agent Based Asynchronous Approximate Dynamic Programming, *Lecture Notes in Artificial Intelligence*, Springer, Vol. 3690, *Proceedings of the 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2005)*, September 15-17, 2005, Budapest, Hungary, pp 388-397, 2005.
- [14] Davari, M., Demeulemeester, E., Leus, R. Nobibon, F. T.: Exact Algorithms for Single-Machine Scheduling with Time Windows and Precedence Constraints, *Journal of Scheduling*, Vol. 19, Issue 3, pp. 309-334, 2016.
- [15] Defersha, F. M., Chen, M.: A Parallel Genetic Algorithm for a Flexible Job-Shop Scheduling Problem with Sequence Dependent Setups, *International Journal of Advanced Manufacturing Technology*, 49 (1), pp. 263-279, 2012.
- [16] Erdélyi, F., Tóth, T., Kulcsár, Gy., Bikfalvi, P.: Some New Considerations for Applying MES Models to Improve the Effectiveness of Production Operations Management in Discrete Manufacturing, *Proceedings of the 14th International Conference on Modern Information Technology in the Innovation Processes of the Industrial Enterprises - MITIP 2012*, Budapest, Hungary, pp. 391-400, 2012.

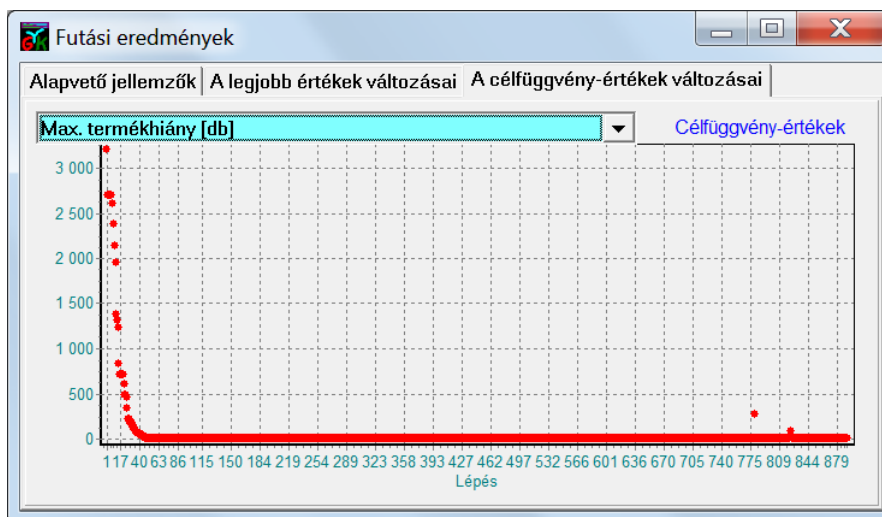
- [17] Gharbi, A., Haouari, M.: Optimal parallel machines scheduling with availability constraints, *Discrete Applied Mathematics*, Vol. 148, pp. 63-87, 2005.
- [18] Graham, R. L., Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G.: Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, *Annals of Discrete Mathematics*, 5, pp. 287–326, 1979.
- [19] Kaabi, J., Harrath, Y.: A survey of parallel machine scheduling under availability constraints, *International Journal of Computer and Information Technology*, Vol. 3, Issue 2, pp. 238-245, 2014.
- [20] Kádár, B., Pfeiffer, A., Monostori, L.: Stability-oriented Evaluation of Rescheduling Strategies by using Simulation, *Computers in Industry*, Volume 58, Issue 7, pp. 630-643, 2007.
- [21] Koulamas, C.: The total tardiness problem: review and extensions, *Operations Research*, Vol. 42, pp. 1025-1041, 1994.
- [22] Kulcsár, Gy.: A Practice-Oriented Approach for Solving Production Scheduling Problems, *Proceedings of the XXV microCAD International Scientific Conference*, Miskolc, Hungary, pp. 61-66, 2011.
- [23] Kulcsár, Gy.: Ütemezési modell és heurisztikus módszerek az igény szerinti tömeggyártás finomprogramozásának támogatására, PhD értekezés, Miskolci Egyetem, 2007.
- [24] Kulcsár, Gy., Erdélyi F.: A New Approach to Solve Multi-Objective Scheduling and Rescheduling Tasks, *International Journal of Computational Intelligence Research*, 3 (4), pp. 343-351, 2007.
- [25] Kulcsár, Gy., Erdélyi, F.: Modelling and solving of the extended flexible flow shop scheduling problem, *Production Systems and Information Engineering*, Vol. 3, pp. 121-139, 2006.
- [26] Lamothe, J., Marmier, F., Dupuy, M., Gaborit, P., Dupont, L.: Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints, *Computers & Operations Research*, Vol. 39, Issue 6, pp. 1236–1244, 2012.
- [27] Lazarev, A. A., Arkhipov, D. I., Werner, F.: Scheduling jobs with equal processing times on a single machine: minimizing maximum lateness and makespan, *Optimization Letters*, pp. 1-13, 2016.
- [28] Lee, C.-Y., Chen, Z.-L.: Machine Scheduling with Transportation Considerations, *Journal of Scheduling*, Vol. 4, Issue 1, pp. 3-24, 2001.
- [29] Lee, K., Leung, J. Y.-T., Pinedo, M. L.: Scheduling jobs with equal processing times subject to machine eligibility constraints, *Journal of Scheduling*, Vol. 14, Issue 1, pp. 27-38, 2011.
- [30] Lei, D.: Multi-objective production scheduling: a survey, *The International Journal of Advanced Manufacturing Technology*, Vol. 43, Issue 9-10, pp. 926-938, 2009.
- [31] Lengyel, A., Erdélyi, F.: Behaviour Based Combined Approaches to Uncertainty Management in Manufacturing Systems, *Proceedings of the 6th International Workshop on Emergent Synthesis, IWES'06*, 18-19 August, 2006, Kashiwa, Japan, pp. 77-82, 2006.
- [32] Linn, R., Zhang, W.: Hybrid flow shop scheduling: a survey, *Computers and Industrial Engineering*, Vol. 37, No. 1-2, pp. 57-61, 1999.
- [33] Lin, Y., Li, W.: Parallel machine scheduling of machine-dependent jobs with unit-length, *European Journal of Operational Research*, Vol. 156, Issue 1, pp. 261–266, 2004.

- [34] Loukil, T., Teghem, J., Tuyttens, D.: Solving multi-objective production scheduling problems using metaheuristics, *European Journal of Operational Research*, 161, pp. 42-61, 2005.
- [35] Ma, Y., Chu, C. B., Zuo, C. R.: A survey of scheduling with deterministic machine availability constraints, *Computers & Industrial Engineering*, Vol. 58, pp. 199-211, 2010.
- [36] Mokotoff, E.: Parallel machine scheduling problems: a survey, *Asia-Pacific Journal of Operational Research*, Vol. 18, pp. 193-242, 2001.
- [37] Petrovic, D., Duenas, A.: A Fuzzy Logic Based Production Scheduling/Rescheduling in the Presence of Uncertain Disruptions, *Fuzzy Sets and Systems*, Vol. 157, pp. 2273-2285, 2006.
- [38] Pinedo, M. L.: *Planning and Scheduling in Manufacturing and Service*, 2nd ed., Springer Verlag New York, 2009.
- [39] Pinedo, M. L.: *Scheduling Theory, Algorithms, and Systems*, 3rd ed., Springer Verlag New York, 2008.
- [40] Quadt, D., Kuhn, H.: A Taxonomy of Flexible Flow Line Scheduling Procedures, *European Journal of Operational Research*, 178, pp. 686-698, 2007.
- [41] Sbalzarini, L. F., Müller, S., Koumoutsos, P.: Multiobjective optimization using evolutionary algorithms, In *Center of Turbulence Research, Proceedings of the Summer Program 2000*, pp. 63-74, 2000.
- [42] Scholten, B.: *The road to integration. A guide to applying the ISA 95 standard in manufacturing*, ISA Instrumentation Systems and Automation Society. ISBN 0 9792 343-8-7, 2007.
- [43] Smith, K. I., Everson, R. M., Fieldsend, J. E.: Dominance Measures for Multi-Objective Simulated Annealing, *Proceedings of Congress on Evolutionary Computation*, Portland, Oregon, USA, 19-23 June 2004, pp. 23-30, 2004.
- [44] Tóth, T., Erdélyi, F.: New Consideration of Production Performance Management for Discrete Manufacturing, *Proceedings of the 8th International Conference on The Modern Information Technology in the Innovation Processes of the Industrial Enterprises*, September 11-12, 2006, Budapest, Hungary, pp. 435-444, 2006.
- [45] Tóth, T.: *Termelési rendszerek és folyamatok*, Miskolci Egyetemi Kiadó, Miskolc, 2004.
- [46] Tóth, T.: *Tervezési elvek, modellek és módszerek a számítógéppel integrált gyártásban*. Miskolci Egyetemi Kiadó, 2006.
- [47] Vieira, G., Hermann, J., Lin, E.: Predicting the Performance of Rescheduling Strategies for Paralell Machines Systems, *Journal of Manufacturing Systems*, Vol. 19, No. 4, pp. 256-266, 2000.
- [48] Vieira, G., Hermann, J., Lin, E.: Rescheduling Manufacturing Systems: A Framework of Strategies, Policies and Methods, *Journal of Scheduling*, Vol. 6, No. 1, pp. 35-58, 2003.
- [49] Wang, W.: Flexible flow shop scheduling: optimum, heuristics, and artificial intelligence solutions, *Expert Systems*, Vol. 22, No. 2, pp. 78-85, 2005.
- [50] Weng, M. X., Lu, J., Ren, H.: Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective, *International Journal of Production Economics*, Vol. 70, pp. 215-226, 2001.

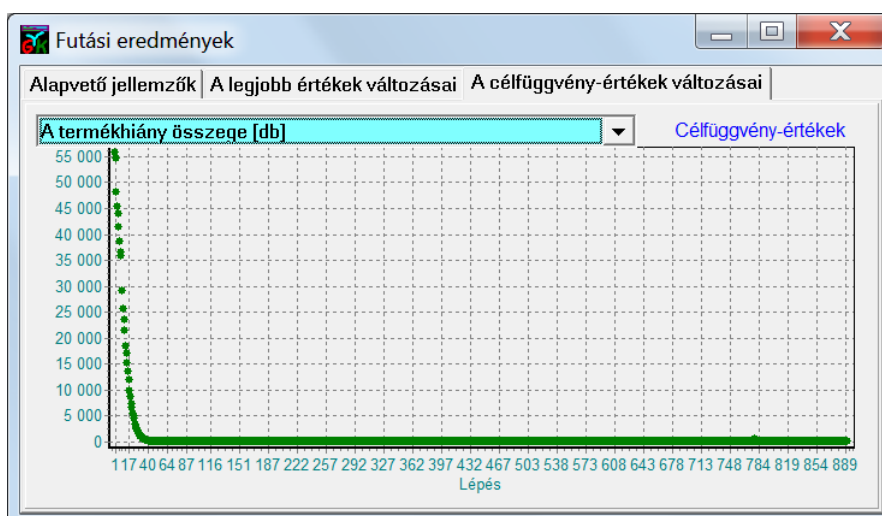
Melléletek

I. Melléklet: A célfüggvény-értékek változása az EasyForce kereső algoritmus működése során

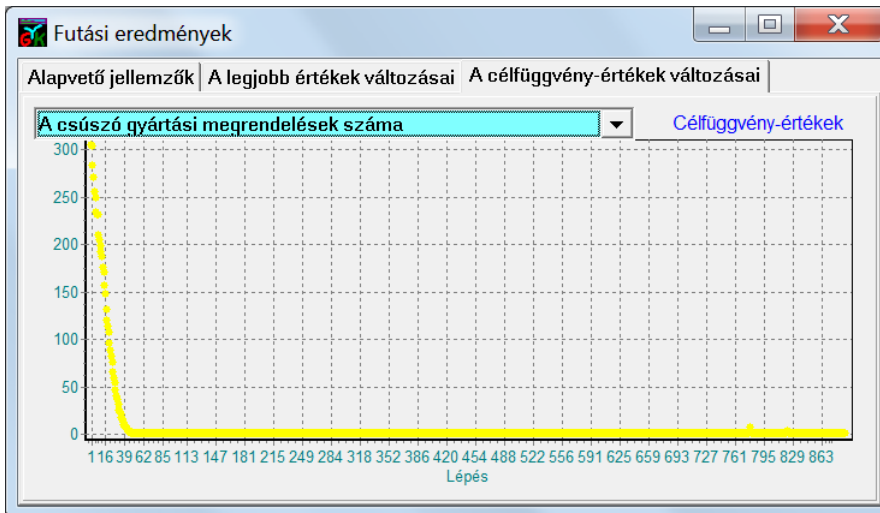
A 61.-78. ábrák az 4.4.6. alfejezetben bemutatott konkrét példa megoldását jellemző 18 célfüggvény értékeinek változását mutatják az *EasyForce* keresés során megtett lépések függvényében. A diagramok az adott lépésen belül megvizsgált kiterjesztett ütemtervek részleteit nem szemléltetik, az adott lépéshez a kiterjesztés legjobb megoldásának értékeit rendeli hozzá.



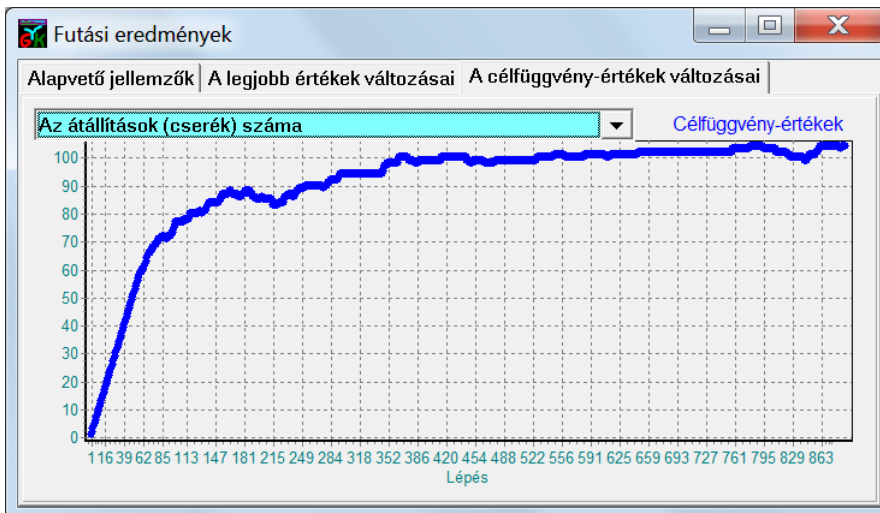
61. ábra: (f₁) A rendelések teljesítésének időpontjában fellépő legnagyobb termékhiány [db].



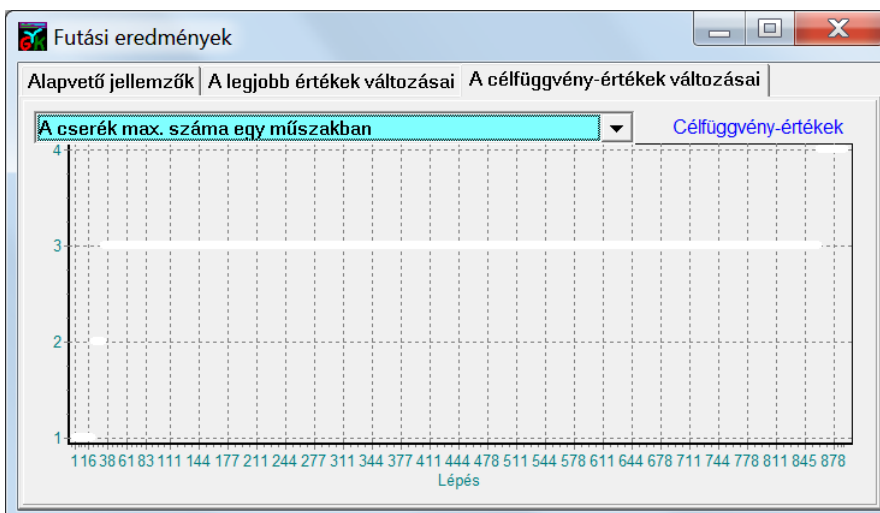
62. ábra: (f₂) A rendelések teljesítésének időpontjában fellépő termékhiányok összege [db].



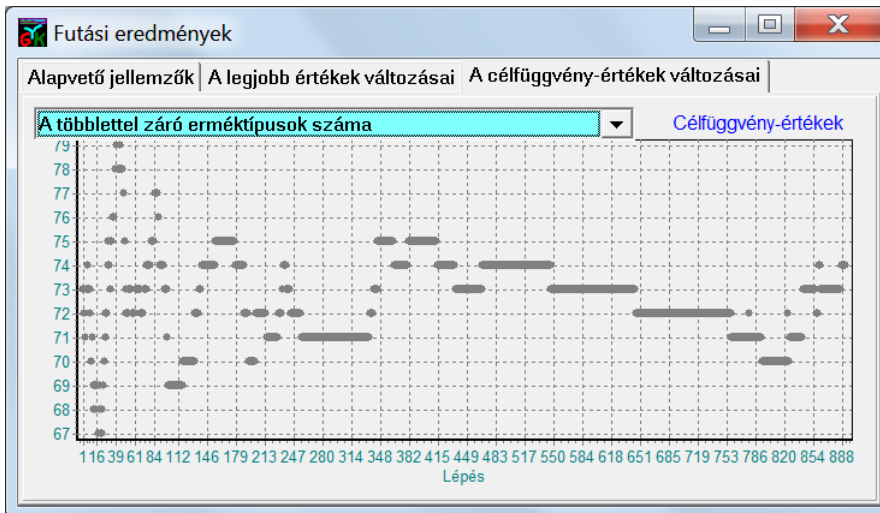
63. ábra: (f_3) A csúszó (határidőt túllépő) rendelés-teljesítések száma.



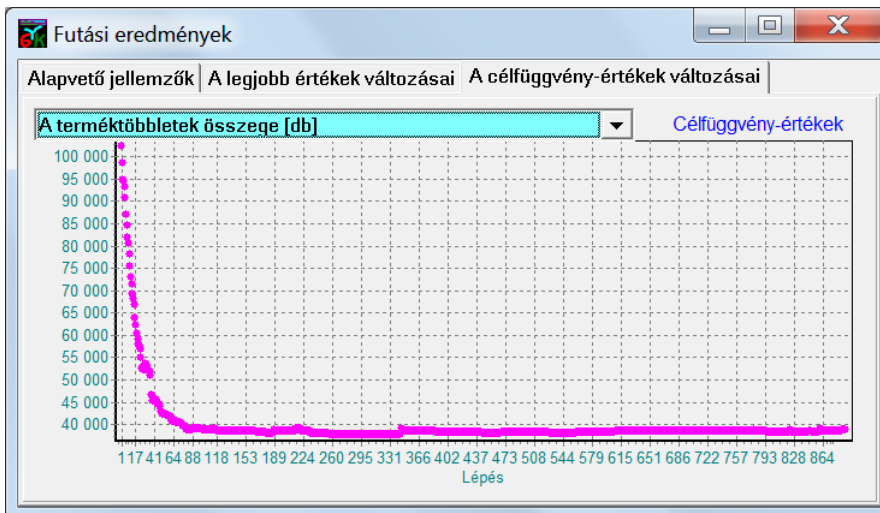
64. ábra: (f_4) A konfiguráció-cserék száma.



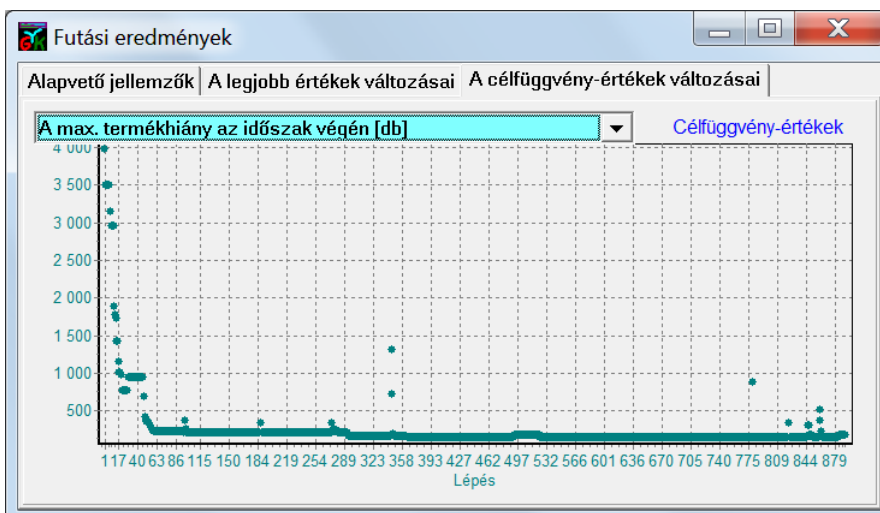
65. ábra: (f_5) A konfiguráció-cserék legnagyobb száma egy műszakban.



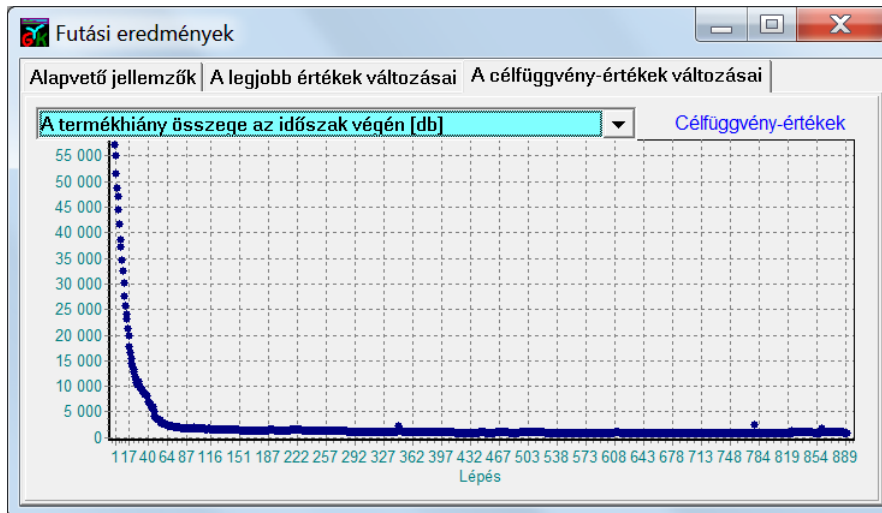
66. ábra: (f_6) A többlettel záró terméktípusok száma.



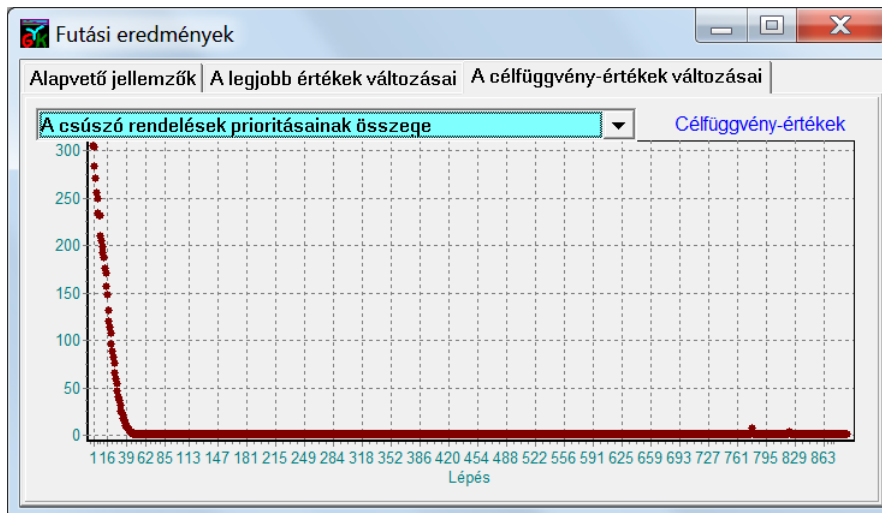
67. ábra: (f_7) A terméktöbbletek összege az időszak végén [db].



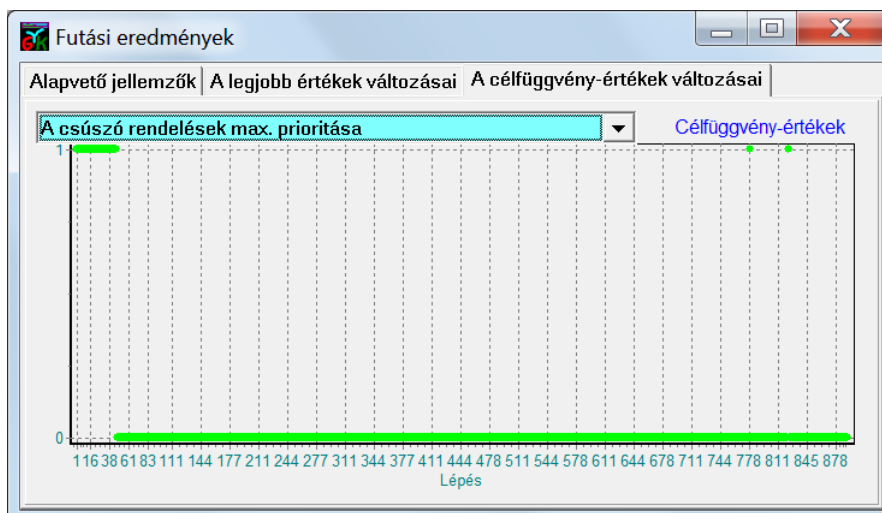
68. ábra: (f_8) A legnagyobb termékhiány az ütemezési időszak végén [db].



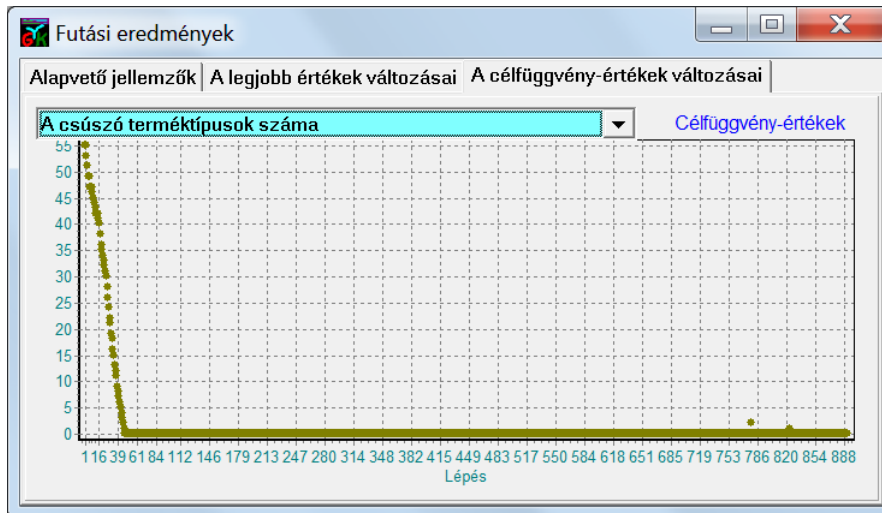
69. ábra: (f_9) A termékhiány összege az ütemezési időszak végén [db].



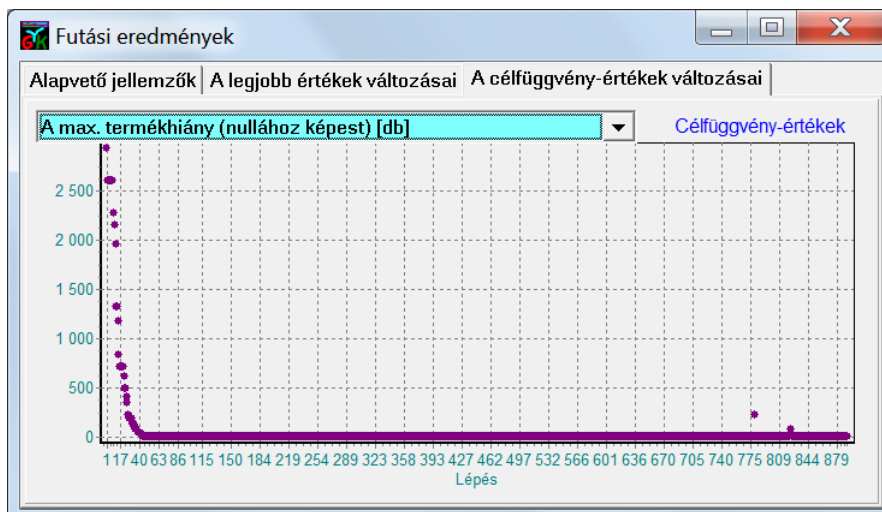
70. ábra: (f_{10}) A határidőre nem teljesített rendelések prioritásainak összege.



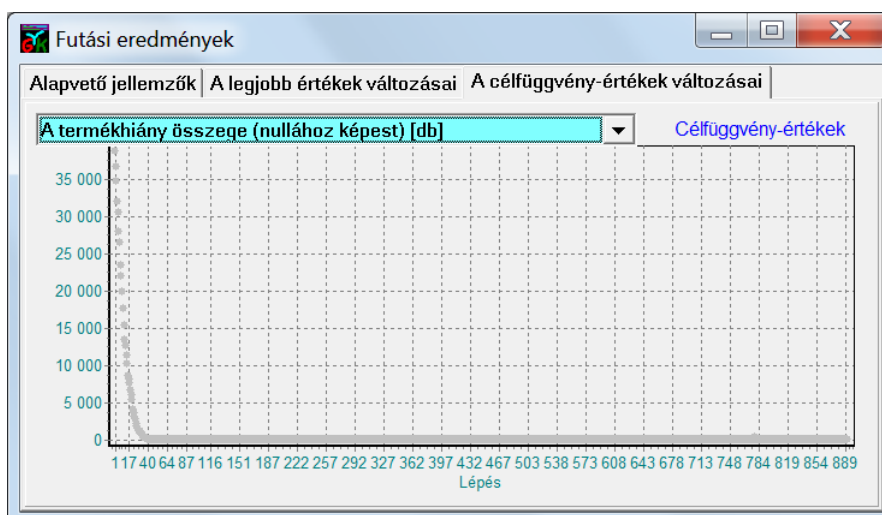
71. ábra: (f_{11}) A határidőre nem teljesített rendelések legnagyobb prioritása.



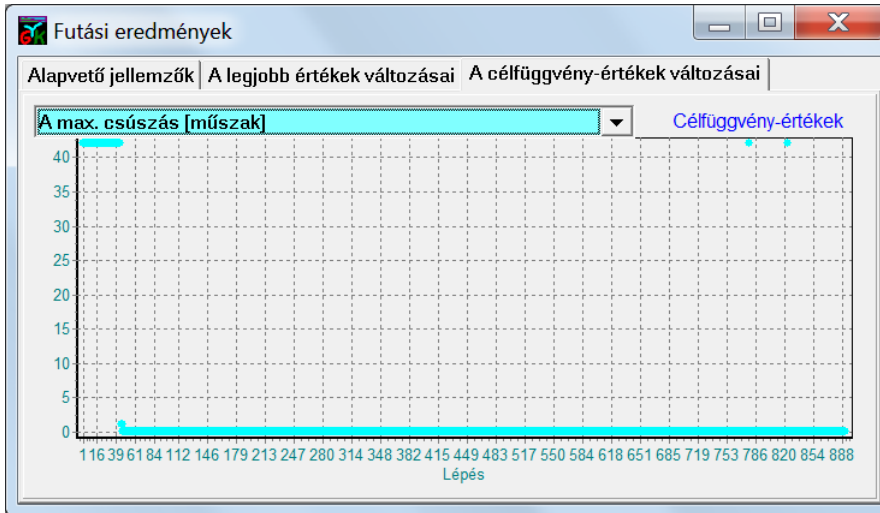
72. ábra: (f_{12}) A csúszo rendelés-teljesítések által érintett terméktípusok száma.



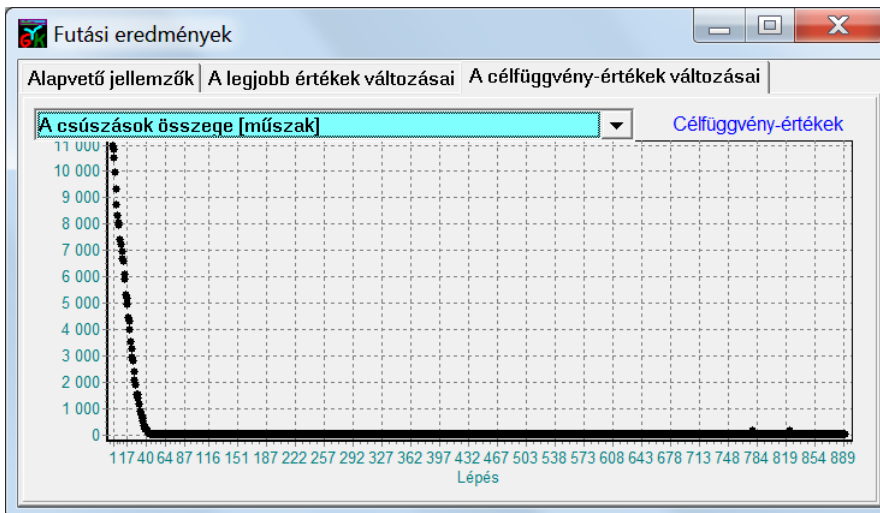
73. ábra: (f_{13}) A legnagyobb termékhiány (nullához képest) az ütemezési időszak végén [db].



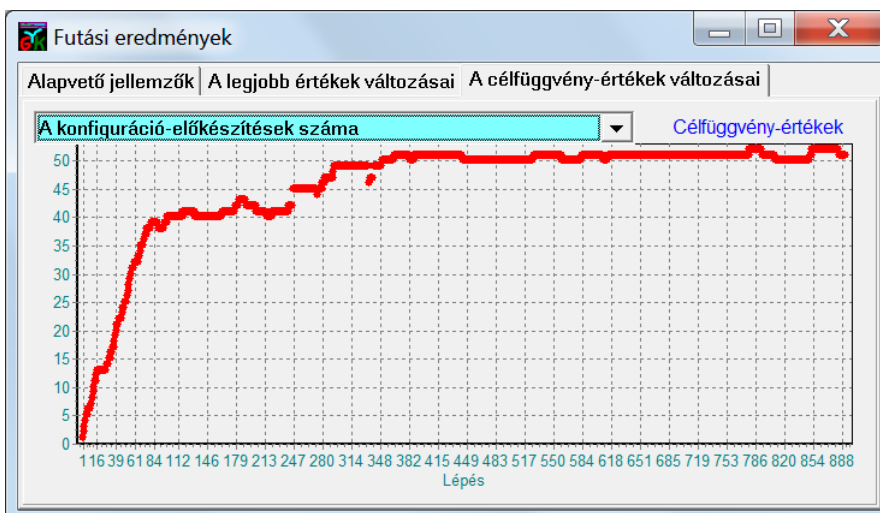
74. ábra: (f_{14}) A termékhiány összege (nullához képest) az ütemezési időszak végén [db].



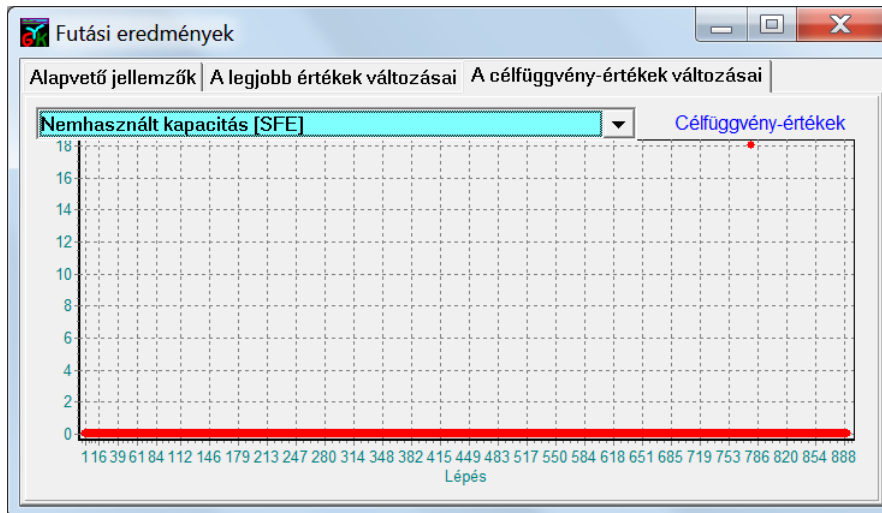
75. ábra: (f_{15}) A rendelés-teljesítések legnagyobb csúszása (határidő túllépése) [műszak].



76. ábra: (f_{16}) A rendelés-teljesítések csúszásainak (határidő-túllépés) összege [műszak].



77. ábra: (f_{17}) A konfiguráció-előkészítési munkák száma.



78. ábra: (f_{18}) Nem használt pályakapacitás [SPU].

II. Melléklet: A prioritási sémák szerepe

Az 4.4.6. alfejezetben bemutatott konkrét példát az *EasyForce* kereső algoritmussal többször megoldottam különböző prioritási sémákat használva. A megoldásokat jellemző célfüggvény-értékek a 10.-14. táblázatokban szerepelnek.

10. táblázat: A Num of Tardy PO sémával elért célfüggvény-értékek.

<i>1. teszt</i>	<i>A megoldások célfüggvény-értékei</i>									
<i>Cél- függvény</i>	<i>#1</i>	<i>#2</i>	<i>#3</i>	<i>#4</i>	<i>#5</i>	<i>#6</i>	<i>#7</i>	<i>#8</i>	<i>#9</i>	<i>#10</i>
<i>f₁</i>	0	5243	4410	2570	3675	4410	4998	4214	0	4459
<i>f₂</i>	0	10422	14179	7212	13334	14089	14020	14693	0	13954
<i>f₃</i>	0	24	29	10	29	29	29	29	0	30
<i>f₄</i>	52	46	53	56	47	48	48	49	67	57
<i>f₅</i>	5	2	2	6	5	3	3	3	4	4
<i>f₆</i>	84	87	88	90	85	91	90	91	82	85
<i>f₇</i>	58562	58220	57118	59739	60928	57058	62983	60642	54464	62995
<i>f₈</i>	1300	6143	5261	3130	4477	5212	5849	5016	796	5310
<i>f₉</i>	10673	19317	18795	13079	20889	21807	20690	20908	9398	22790
<i>f₁₀</i>	0	24	29	10	29	29	29	29	0	30
<i>f₁₁</i>	0	1	1	1	1	1	1	1	0	1
<i>f₁₂</i>	0	2	3	2	3	3	3	3	0	4
<i>f₁₃</i>	0	5243	4361	2424	3577	4312	4949	4116	0	4410
<i>f₁₄</i>	0	10422	10817	4504	9143	12097	13318	12235	0	11021
<i>f₁₅</i>	0	42	42	42	42	42	42	42	0	42
<i>f₁₆</i>	0	1008	1186	367	1139	1218	1218	1184	0	1194
<i>f₁₇</i>	24	25	30	25	22	24	25	26	30	32
<i>f₁₈</i>	6	52	11	3	0	38	1	5	11	15

11. táblázat: A PO Tardiness sémával elért célfüggvény-értékek.

2. teszt	<i>A megoldások célfüggvény-értékei</i>									
<i>Cél- függvény</i>	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
f_1	136	230	0	136	2498	143	601	2191	62	71
f_2	749	546	0	586	4780	531	954	4378	72	283
f_3	8	7	0	12	7	13	5	8	2	8
f_4	74	64	59	70	70	66	86	80	58	65
f_5	2	3	2	3	3	3	5	5	3	4
f_6	81	86	80	85	81	80	80	82	82	79
f_7	53562	59695	55911	53992	49262	60372	52865	52616	61134	58197
f_8	942	1308	1265	1148	1536	1606	875	1200	1200	1200
f_9	10782	13299	9568	10693	11162	15071	9211	9338	13159	10741
f_{10}	8	7	0	12	7	13	5	8	2	8
f_{11}	1	1	0	1	1	1	1	1	1	1
f_{12}	4	4	0	5	1	9	3	2	2	4
f_{13}	0	0	0	0	1205	0	0	728	0	0
f_{14}	0	0	0	0	1205	0	0	728	0	0
f_{15}	1	1	0	1	42	1	2	42	1	1
f_{16}	2	1	0	6	156	7	5	116	1	3
f_{17}	40	35	37	34	40	35	56	44	28	33
f_{18}	16	17	7	38	5	11	21	4	10	8

12. táblázat: A Product Shortage sémával elért célfüggvény-értékek.

3. teszt	<i>A megoldások célfüggvény-értékei</i>									
<i>Cél- függvény</i>	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
f_1	0	10	0	0	0	0	207	0	0	10
f_2	0	10	0	0	0	0	318	0	0	10
f_3	0	1	0	0	0	0	3	0	0	1
f_4	51	54	55	57	49	50	60	52	50	58
f_5	5	4	3	5	4	5	4	4	3	4
f_6	84	78	86	80	84	84	78	82	81	87
f_7	57746	55168	50565	53439	53680	59469	52530	54629	57169	48558
f_8	1396	1396	810	1816	1200	1270	1070	1302	1676	1746
f_9	12668	11967	9388	10335	12008	13159	8536	9740	12668	11296
f_{10}	0	1	0	0	0	0	3	0	0	1
f_{11}	0	1	0	0	0	0	1	0	0	1
f_{12}	0	1	0	0	0	0	2	0	0	1
f_{13}	0	10	0	0	0	0	10	0	0	10
f_{14}	0	10	0	0	0	0	10	0	0	10
f_{15}	0	42	0	0	0	0	42	0	0	42
f_{16}	0	42	0	0	0	0	51	0	0	42
f_{17}	27	27	26	31	25	23	33	22	25	28
f_{18}	2	11	2	10	0	45	0	0	0	44

13. táblázat: A Setup for PO sémával elért célfüggvény-értékek.

4. teszt	<i>A megoldások célfüggvény-értékei</i>									
<i>Cél- függvény</i>	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
f_1	0	0	0	0	0	0	12	77	0	77
f_2	0	0	0	0	0	0	12	199	0	155
f_3	0	0	0	0	0	0	1	4	0	3
f_4	54	55	53	60	54	50	61	52	49	63
f_5	6	4	2	4	3	2	3	2	2	3
f_6	84	78	83	80	86	85	81	84	82	82
f_7	51905	51938	56496	55542	50355	55081	56211	51054	54417	49906
f_8	1236	1304	1070	1676	1254	1326	1200	1200	1606	1005
f_9	10016	11597	11434	10846	9248	10818	11889	11451	14051	8107
f_{10}	0	0	0	0	0	0	1	4	0	3
f_{11}	0	0	0	0	0	0	1	1	0	1
f_{12}	0	0	0	0	0	0	1	3	0	1
f_{13}	0	0	0	0	0	0	0	0	0	0
f_{14}	0	0	0	0	0	0	0	0	0	0
f_{15}	0	0	0	0	0	0	1	1	0	9
f_{16}	0	0	0	0	0	0	1	4	0	8
f_{17}	24	24	24	28	27	24	28	26	25	31
f_{18}	2	3	2	2	3	2	2	1	0	2

14. táblázat: Az Agile Control sémával elért célfüggvény-értékek.

5. teszt	A megoldások célfüggvény-értékei									
Cél- függvény	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
f_1	0	0	0	0	0	0	0	0	0	0
f_2	0	0	0	0	0	0	0	0	0	0
f_3	0	0	0	0	0	0	0	0	0	0
f_4	94	107	99	115	114	97	104	110	98	84
f_5	5	3	5	6	6	5	4	4	4	2
f_6	69	70	72	78	73	66	70	76	68	71
f_7	37021	40073	42783	37805	40842	38050	35680	43264	42141	36947
f_8	44	279	44	84	84	44	199	99	225	134
f_9	105	1922	94	268	301	86	879	445	1741	413
f_{10}	0	0	0	0	0	0	0	0	0	0
f_{11}	0	0	0	0	0	0	0	0	0	0
f_{12}	0	0	0	0	0	0	0	0	0	0
f_{13}	0	0	0	0	0	0	0	0	0	0
f_{14}	0	0	0	0	0	0	0	0	0	0
f_{15}	0	0	0	0	0	0	0	0	0	0
f_{16}	0	0	0	0	0	0	0	0	0	0
f_{17}	44	51	49	58	55	42	41	58	49	44
f_{18}	0	4	2	0	0	0	2	3	2	0

Az 1. teszt során a rendelések csúszása, a 2. teszt során a csúszó rendelések száma vezérelte a keresési folyamatot. Ezek a célfüggvények a klasszikus határidő-orientált megközelítést képviselték. A 3. teszt során az általam javasolt új megközelítést használtam, melyben a csúszás helyett a rendelések előírt teljesítési időpontjában fennálló termékhiány vezérli a keresést. Összehasonlítva a 10.-12. táblázatok adatait látható, hogy az új megközelítést képviselő séma sokkal hatékonyabban működött a klasszikus megközelítést képviselő másik két sémához képest.

A 4. és az 5. teszt eredményei azt mutatják (13.-14. táblázat), hogy a határidők betartását célzó célfüggvények együtt jobban teljesítenek a keresési folyamatban mint külön-külön. A 4. tesztben a rendelések teljesítéséig tervez be új konfiguráció-cseréket a rendszer, míg az 5. tesztben az alkalmazott séma szerint a rendelések teljesítését követően a készlet szintek javítása érdekében további konfiguráció-cseréket is betervez az *EasyForce* ütemező algoritmus. Az *Agile_Control* séma alkalmazásával az 5. tesztben mind a tíz esetben hiánytalanul határidőre teljesültek a rendelések.

III. Melléklet: Tanaka-benchmark feladatok megoldása MOMOTS algoritmussal

A 3. fejezetben bemutatott új kiterjesztett ütemezési modellt és megoldási módszert alkalmaztam jól ismert klasszikus ütemezési feladatok megoldására. A tesztelés során a *MOMOTS* keresési módszert a $P/d_i/\sum T_i$ (Total Tardiness Problem on Identical Parallel Machines) típusú Shunji Tanaka által definiált benchmark feladat-instanciákon futtattam. A feladatokat leíró bemeneti (input) adatok és az optimális megoldást jellemző célfüggvényértékek az alábbi weboldalon érhetők el:

<https://sites.google.com/site/shunjitanaka/pmtt>

Példaként ebben a mellékletben a legnagyobb méretű Tanaka-benchmarkokon végzett futtatások eredményeit foglalom össze. A vizsgált tesztfeladatokban 10 gép és 25 munka szerepelt. A vizsgált benchmark 125 feladat-instanciát tartalmazott ötös csoportokra osztva a műveleti idők és a határidők generálásának paramétereit alapján.

A vizsgált $P/d_i/\sum T_i$ ütemezési feladattípus jellemzői a következők:

- Minden egyes munkához egyetlen művelet (operáció) tartozik.
- A gépek folyamatosan rendelkezésre állnak.
- Egy gép egyszerre csak egy munkán dolgozhat.
- Egy munkán egyszerre csak egy gép dolgozhat.
- A gépek párhuzamosan működhetnek és teljesen egyenértékűek.
- A munkák nem szakíthatók meg.
- A munkák egymástól függetlenek.
- A munkáknak nincs indítási időkorlátja (minden munka a 0 időponttól kezdve indítható).
- Nincs definiált átállítási idő a munkák között (a gépek átállítási ideje 0).
- Minden egyes munkának saját határideje és műveleti ideje van.
- Az ütemezés célja a határidő-túllépések (csúszások) összegének minimalizálása.

A *MOMOTS* algoritmus futtatásakor az alábbi keresési paramétereket használtam:

- A legnagyobb megtehető lépésszám (leállási feltétel): 2000.
- Egy lépésben megvizsgálható szomszédos megoldások megengedett száma: 100.
- A tabulistán tárolt megoldások megengedett legnagyobb száma: 150.
- A szomszédosági operátorok prioritásai: az N_1 - N_5 operátorok prioritása 1 és az N_6 - N_8 operátorok prioritása 0 (a műszakok beosztását módosító operátorok ki vannak kapcsolva).
- A célfüggvények prioritása: Ebben a speciális esetben a csúszások időösszegének ($\sum T_i$) minimalizálása a cél, így az f_2 célfüggvény prioritása 10 és a többi célfüggvény prioritása 0.
- Az iterációk (futtatások) száma: 10. Minden egyes feladat-instancián 10-szer futott le a *MOMOTS* algoritmus véletlenszerűen választott kezdeti megoldásból kiindulva.

A tesztfeladatok vizsgálata során azt tapasztaltam, hogy a *MOMOTS* algoritmus mind a 125 feladat-instancia esetében megtalálta az ismert optimális megoldást.

15. táblázat: A Tanaka-benchmark feladatok megoldásának eredményei.

<i>Az optimális megoldás megtalálásának gyakorisága 10 iterációban</i>		
<i>Az optimum elérésének száma</i>	<i>Instanciák száma</i>	<i>Instanciák aránya</i>
0	0	0
1	3	0,024
2	0	0
3	0	0
4	0	0
5	0	0
6	2	0,016
7	3	0,024
8	0	0
9	5	0,04
10	112	0,896

A 125 feladat-instanciából 112 példány esetében 10 futtatásból mind a 10 futtatás optimális megoldást talált (15. táblázat). A maradék 13 instancia esetében a 16. táblázat részletezi az egyes futtatások során előállított megoldások célfüggvény-értékeit és a 17. táblázat foglalja össze a megoldások átlagos célfüggvény-értékének optimumtól való relatív eltérését. Látható, hogy még a legnagyobb átlagos eltérés is csupán 0,75% alatt marad (25_10_02_02_001 instancia).

16. táblázat: Példák a MOMOTS algoritmus konkrét futási eredményeire.

<i>Tanaka-benchmark</i>	<i>A Tanaka-optimum</i> <i>célfüggvény-értéke</i>	<i>A MOMOTS algoritmussal előállított megoldások</i> <i>célfüggvény-értékei</i>									
		<i>#1</i>	<i>#2</i>	<i>#3</i>	<i>#4</i>	<i>#5</i>	<i>#6</i>	<i>#7</i>	<i>#8</i>	<i>#9</i>	<i>#10</i>
<i>25_10_02_02_001</i>	174	174	174	174	187	174	174	174	174	174	174
<i>25_10_02_04_004</i>	85	86	86	87	85	85	85	85	85	86	85
<i>25_10_02_06_001</i>	78	78	78	78	81	78	78	78	78	78	78
<i>25_10_02_06_003</i>	53	53	53	53	53	53	54	54	53	53	54
<i>25_10_02_06_005</i>	113	113	113	113	113	113	114	113	113	113	113
<i>25_10_02_08_001</i>	76	76	76	79	76	76	76	76	76	76	76
<i>25_10_04_04_001</i>	444	444	444	451	444	444	444	447	444	447	444
<i>25_10_04_06_002</i>	325	326	326	326	325	326	326	327	326	326	326
<i>25_10_04_08_005</i>	495	495	495	495	495	496	495	495	497	495	496
<i>25_10_04_10_003</i>	336	338	338	337	337	339	338	338	336	339	338
<i>25_10_06_04_004</i>	635	636	636	636	636	635	636	636	636	636	636
<i>25_10_06_10_001</i>	697	697	697	697	697	697	697	697	700	697	697
<i>25_10_06_10_003</i>	650	650	652	650	652	652	650	650	650	650	651

17. táblázat: Példák a MOMOTS algoritmus futási eredményeinek átlagértékeire.

<i>Tanaka-benchmark</i>	<i>Értékelés a megoldás célfüggvény-értéke alapján</i>				
<i>Instancia</i>	<i>Tanaka optimum</i>	<i>legjobb megtalált</i>	<i>10 megoldás átlaga</i>	<i>az átlag relatív távolsága az optimumtól</i>	<i>az optimum elérésének gyakorisága 10 futtatás során</i>
25_10_02_02_001	174	174	175,3	0,007471	9
25_10_02_04_004	85	85	85,5	0,005882	6
25_10_02_06_001	78	78	78,3	0,003846	9
25_10_02_06_003	53	53	53,3	0,00566	7
25_10_02_06_005	113	113	113,1	0,000885	9
25_10_02_08_001	76	76	76,3	0,003947	9
25_10_04_04_001	444	444	445,3	0,002928	7
25_10_04_06_002	325	325	326	0,003077	1
25_10_04_08_005	495	495	495,4	0,000808	7
25_10_04_10_003	336	336	337,8	0,005357	1
25_10_06_04_004	635	635	635,9	0,001417	1
25_10_06_10_001	697	697	697,3	0,00043	9
25_10_06_10_003	650	650	650,7	0,001077	6

A 15. táblázatból az is kiolvasható, hogy 5 instancia esetében 9-szer, 3 esetében 7-szer, 2 esetében 6-szor adódott optimális célfüggvény-érték. A 125 instanciából csupán 3 esetében fordult az elő, hogy 10 futtatásból egyszer találta meg az optimumot a kereső algoritmus. Ugyanakkor a 16. és a 17. táblázat adatainak összevetéséből az is látható, hogy ebben a három esetben minden egyes futtatás során nagyon kicsi az optimális értéktől való eltérés mértéke.

A 3. fejezetben bemutatott kiterjesztett modell és a *MOMOTS* algoritmus – annak ellenére, hogy nem a most vizsgált $P/d_i/\sum T_i$ feladattípusra koncentrálnak került kifejlesztésre – kiváló eredményeket ért el a Tanaka-benchmark feladatok megoldása során. Ezek a tények a korábban bemutatott ipari feladatokon végzett tesztelések eredményeihez hasonlóan alátámasztják azt a következtetést, hogy a *MOMOTS* algoritmus a célul kitűzött nehezebb és összetettebb $FF2, M_g, Q_{i,m}, Set_{px,py,m}, Cal_m, B_p/R_i^{02}, D_i, A_{i,g}, T_{mx,my} / f_1, f_2, \dots, f_K$ feladatok esetében is feltételezhetően az optimumhoz (a célfüggvény-értékek szempontjából) közeli megoldásokat állít elő.

New Scientific Results

Thesis 1 Demand-driven Planning Method for Shift Assignments in Production Fine Schedules

In order to synchronize the human and machinery resources, I developed a demand-driven shift-assignment model that is able to manage the availability time intervals of the resources as decision variables and/or constraints in production fine scheduling problems. To solve this planning problem of fine scheduling, I elaborated a new method through integrated solving of the scheduling and the shift-assignment problems of discrete manufacturing-assembly systems.

The most important features of the integrated planning method are as follows:

- 1.1. The model extension concerning the shift assignment can be applied to any type of production scheduling problem.
- 1.2. The synchronized solving of the problem consists of two phases: in the first phase, predefined time intervals as constants (constraints) can be assigned to the resources individually. In the second phase, the assignments of active shifts (decision variables) can be finalized based on the actual loads that appear while solving the scheduling problem.
- 1.3. In order to minimize the cost of shifts, I developed the concept of a new objective function that uses "relative cost of shift" approach.
- 1.4. Using the developed method, the scheduling problem and the shift-assignment problem can be solved in the same time by applying an advanced multi-operator and multi-objective searching technique (MOMOTS, Multi-Operator and Multi-Objective Tabu Search). The neighboring operator (or operators) of the selected search algorithm of the initial (default) scheduling problem should be complemented by the shift-assignment operators and the original objective functions should be complemented by the objective functions concerning to the shift assignment.
- 1.5. The integrated solving method supports, in addition to the above, the management of alternative technological routing variants and the consideration of material handling times and intermediate buffers with limited sizes, as well as the distinction and management of the sizes of production order series, manufacturing series and logistical series.
- 1.6. By solving a large real industrial problem, I verified that the new approach is suitable for integrated solving of production scheduling and shift-assignment problems in discrete component manufacturing and assembly processes.

The results summarized in Thesis 1 are described in detail in Chapter 3 of the dissertation and have been published in 12 papers: [s4], [s3], [s5], [s7], [s9], [s10], [s11], [s12], [s14], [s15], [s17], [s16].

Thesis 2 Proactive Approach Based Model and Combined Scheduling Method for Solving Fine Scheduling Problems of "Drawdown"-oriented Component Manufacturing Systems

For integrated scheduling of make-to-order and make-to-stock manufacturing, I developed a new model and solving method that is able to maintain the product-dependent minimum stock levels defined by supplier contracts and to fulfill direct orders with strict due dates (drawdowns) simultaneously. In addition to the integration of production scheduling and demand-driven inventory management, the proposed modeling approach makes it possible to solve the combined scheduling of the manufacturing main processes and their related preparatory activities.

The most important features of the model are as follows:

- 2.1 The developed IPSIC model is fundamentally based on the production features and requirements of automotive seat elements (a model of circle-shaped parallel production lines), but it can be quickly adapted to other manufacturing environments.
- 2.2 The model manages the fulfillment of due date production orders and the optimization of product-dependent stock levels in a combined (integrated) manner.
- 2.3 The model takes into account the specialties and the limited availabilities of resources.
- 2.4 It implements scheduling of synchronized set-up tasks for production lines taking into account even the preparatory tasks of the production.
- 2.5 It supports the demand-driven management of the configuration of shared accessible tools and devices necessary to perform the manufacturing tasks according to the current technological standards and alternatives.

The results summarized in Thesis 2 are described in detail in subsections 4.1, 4.2, and 4.4 of the dissertation and have been published in papers [s1], [s2], [s6], [s13].

Thesis 3 A New Model and Solving Method to Solve Single Operation Scheduling Problems with Timely-Variable Number of Resources

I developed the model $P(s) | p_i=1; r_i=integer; d_i=integer | L_{max}$ and its exact solving algorithm to schedule single operation jobs with due dates. The most important novelty of the model is that the actual number of the available parallel resources can depend on time. I developed an improved JIT-oriented algorithm that reduces too-early completion of jobs while due dates are met by using safety control parameters.

The most important features of the model and the solving approach are as follows:

- 3.1 Based on a demand motivated by a real industrial problem, I developed a new generalized model that can be considered as an extension and generalization of the model $P | p_i=1; r_i=integer; d_i=integer | L_{max}$ found in the literature.

- 3.2 To solve the problems described with the new model, I developed a polynomial run-time solving algorithm: Slack-Oriented Algorithm.
- 3.3 To reduce the adverse effects of unexpected events arising from external and internal sources, I developed an improved JIT-oriented algorithm that reduces too-early completion of the jobs while meeting due dates using safety control parameters.
- 3.4 For applying the model and solving methods in practice, I developed a special problem-transformation procedure that provides a clear mapping of the formal model and a real workshop scheduling problem type.

The results summarized in Thesis 3 are described in detail in subsection 4.3 of the dissertation and have been published in papers [s1], [s2], [s6], [s13].