



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

MESTERSÉGES INTELLIGENCIA TANSZÉK

Gyártásütemezés kis méretű fűrészüzemekben

Szerző:

Kebelei Csaba

programtervező informatikus MSc

Külső témavezető:

Dr. Hegyháti Máté

egyetemi docens

Soproni Egyetem

Belső témavezető:

Dr. Tóth Tekla

egyetemi adjunktus

Eötvös Loránd

Tudományegyetem

Budapest, 2024

Absztrakt

A fűrészüzem a faipari termelésnek azon állomása, ahol a farönkökből különféle fűrészárúkat állítanak elő. Ez a rönkfeldolgozás egy többlépcsős folyamat, mely többek között magába foglalja a rönkök osztályozását, ellenőrzéseket, fűrészelést, szabászatot, szárítást, gőzölést. A fűrészelési lépés ütemezését már több tanulmány is vizsgálta a szakirodalomban, melyek elsősorban egygépes feladatokat foglalmaztak meg, többnyire a valós körülmények sztochasztikus jellegére fókuszálva. Ezek a megoldások jól alkalmazhatók a nagyobb üzemekben, azonban a térség kisebb, kevésbé automatizált fűrészüzemeiben több olyan részlet is fontos szerepet játszhat, melyeket az általános megközelítések nem vesznek figyelembe. Jelen munka egy mélyebb, speciálisabb feladatosztályt vizsgál, ahol kulcsfontosságú tényező a humánerőforrás-hiány, valamint a különböző fűrésztechnológiák közötti különbségek kezelése. A problémaosztályra vegyes-egész lineáris programozási modell kidolgozásával adtam megoldást. A módszer gyakorlati alkalmazhatóságát ipari tapasztalatok alapján meghatározott teszteseten, hatékonyságát pedig nagyszámú generált adathalmazon vizsgáltam meg.

Tartalomjegyzék

1. Bevezetés	3
2. Szakirodalmi áttekintés	5
2.1. Ütemezési problémák	5
2.2. Fűrészüzemek	8
2.3. Fűrészüzemek ütemezési problémái	12
3. Probléma definíció	14
3.1. Informális probléma definíció	14
3.2. Formális probléma definíció	17
3.2.1. Halmazok	17
3.2.2. Paraméterek	17
4. Megoldó módszer	19
4.1. Folytonos váltási időpont alapú MILP modell	19
4.1.1. Változók	20
4.1.2. Korlátozások	21
4.1.3. Célfüggvény	24
4.2. Diszkrét időrács alapú MILP modell	24
4.2.1. Modell változtatások	25
4.2.2. Átcsúszási idő kezelése	28
5. Tesztelés	32
5.1. Teszteset készítése grafikus interfészen keresztül és a tesztelési környezet	32
5.1.1. A szoftver architektúrája és felépítése	32
5.1.2. Szoftver használata	35
5.1.3. A tesztelés környezete	37
5.2. Ipari alkalmazhatóság és hatékonyság tesztelése	38
5.2.1. Tesztelés ipari tapasztaltok alapján	38

5.2.2. Nagyszámú generált adathalmaz	45
6. Összefoglalás, konklúzió és a jövőbeli tervek	51
6.1. Összefoglalás	51
6.2. Konklúzió	52
6.3. Jövőbeli tervek	52
Köszönetnyilvánítás	55
A. C-EK modell GMPL kódja	56
B. D[k]-[EK]+ modell GMPL kódja	60
Irodalomjegyzék	64
Ábrajegyzék	70
Táblázatjegyzék	71

1. fejezet

Bevezetés

A fűrészüzemek feladata, hogy a kivágott farönkökből a keresletnek megfelelő fűrészáru készletet gyártson. Ennek folyamata több lépcsőből áll, emiatt a feldolgozást egy komplex tervezés előzi meg. A termelés ütemterve számos módon kivitelezhető. A munkám során olyan megközelítésekkel foglalkozom, melyekkel meghatározható a fűrészüzemek folyamatainak optimális ütemezése.

Az irodalomban több megoldás is megtalálható, mely ilyen üzemek ütemtervének optimalizálásával foglalkozik. A témában publikált kutatások legtöbbje általánosabb megoldásokat javasol, azonban a világon a fűrészüzemeknek számos variánsával találkozhatunk. Ebből kifolyólag több specifikusabb tényezőt is megkülönböztethetünk egy üzem esetén, melyek figyelembevétele sokszor elengedhetetlen a hatékony megoldás meghatározásához. Jelen munkám során megvizsgáltam egy olyan specifikusabb problémaosztályt, mely jobban modellezi a térség kisebb fűrészüzemeit, mint az általánosabb megközelítések.

Jelenleg a faipar munkaerőhiánnyal küzd [1, 2], ennek egy oka, hogy bizonyos pozíciókhoz speciális szaktudás szükséges, illetve az alkalmazottaknak gyakran rendkívül megterhelő fizikai feladatokat kell elvégezniük. Egy fűrészüzem esetén is jelentős problémát jelent, hogy a fűrészelést végző gépek üzemeltetéséhez külön szaktudás szükséges. A nagyobb automatizált fűrészüzemekkel szemben a kisebb üzemekben a farönkök, fűrészáruk mozgatását is az alkalmazottak végzik, mely több ember bevonásával is jelentős fizikai terhelést jelent. Ebből kifolyólag az üzemeknek különös figyelmet kell fordítani a humánerőforrás-menedzsmentre, mely az ütemezés során szintén fontos szerepet kap.

Az emberi erőforrások ütemezésén túlmenően sok szakirodalmi módszer nem ve-

szi figyelembe a fűrészüzemekben alkalmazható rönkvágó technológiák közti különbségeket. A kutatásom során megvizsgáltam a két legelterjedtebb fűrészelést végző géptípus - a keretfűrész és a szalagfűrész - sajátosságait, és modellezés szinten is figyelembe vettem azokat.

Egy fűrészüzem ütemezése esetén szem előtt kell tartani a kereslet igényeit is. Az üzemekben lévő alapgépeken döntést kell hozni arról, hogy mikor, milyen vágási mintát alkalmazzunk, melyek különböző fűrészáru hozamokat eredményeznek. Míg egy drágább automatizált rönkvágó technológiánál nem jelent számottevő időt, egy egyszerűbb keretfűrész technológia esetén elengedhetetlen a minták közti átállásokkal járó idővel való számolás.

Dolgozatomban egy olyan problémaosztályra javasoltam egzakt megoldó módszereket, melyek a fentebb említett specifikusabb tényezőket is figyelembe veszik. A módszereim olyan vegyes-egész lineáris programozási modellek, melyek a különböző paraméterek alapján több napos tervezési időtávra határozzák meg az optimális megoldást. Először egy folytonos idő változó alapú modellt dolgoztam ki, majd felírtam egy diszkrét időrács alapú megközelítést is.

Elkészült egy tesztelés előállítására alkalmas grafikus interfész, mely segítségével könnyen felvihetők a megoldó módszerekhez szükséges adatok, paraméterek. Ezenfelül a szoftver képes a megoldó módszer futtatására is. A megközelítés teszteléséhez meghatározásra került egy adathalmaz, melyet faiparral foglalkozó szakemberek tapasztalatai, szakkönyvbeli információk és piaci tervező szoftver által generált adatok alapján állítottam össze. Ezen a teszten keresztül megvizsgáltam a modellek gyakorlati alkalmazhatóságát, valamint hogy hogyan befolyásolja a munkaerőhiány a megoldásokat. A javasolt megközelítések hatékonyságát nagyszámú generált teszten keresztül is megvizsgáltam.

A dolgozatomban 2. fejezetében egy szakirodalmi áttekintés keretében mutatom be az ütemezési problémák témakörét, a fűrészüzemek általános feldolgozásának folyamatát, illetve az irodalomban lévő javasolt megoldásokat specifikusan a fűrészüzemek ütemezésére. A feldolgozott szakirodalmak és az ipari tapasztalatok alapján azonosított problémát a 3. fejezetben informálisan és formálisan is definiálom. Ezt követően a 4. fejezetben a feladatosztályra javasolt MILP modelleket írom le, melyek tesztelését az 5. fejezetben részletezem. Végezetül a 6. fejezetben összefoglalom a munkám eredményét, továbbá ismertetem a jövőbeli kutatási terveket.

2. fejezet

Szakirodalmi áttekintés

A fejezetben először a 2.1. alfejezetben áttekintést nyújtok az ütemezési problémák témaköréről. Ezt követően általánosan bemutatom a fűrészüzemeket és azok sajátosságait a 2.2. alfejezetben. A fejezet végén a 2.3. alfejezetben pedig összefoglalásra kerülnek a fűrészüzemek ütemezésével kapcsolatos kutatások és azok eredményei.

2.1. Ütemezési problémák

Az élet számos területén találkozhatunk ütemezési problémákkal, például órarendtervezésnél, a mozik műsorlistájának meghatározásakor vagy a számítógépek folyamatainak ütemezésénél. Ezek a problémák az optimalizálási problémák speciális változataiként tekinthetők. A terület sajátossága, hogy a feladatok során időzítéssel kapcsolatos döntéseket kell hozni, illetve dönteni kell arról, hogy az időzített feladatokat mely erőforrásokhoz rendeljük. Mint minden optimalizálási problémánál, szükséges egy cél megfogalmazása, mely alapján a sok alternatív megoldás közül kell meghatározni a legjobbat. Ütemezési feladatok esetén ilyen tipikus célok a befejezési idő, termelési költség minimalizálása vagy a termelés maximalizálása. A célok mellett figyelembe kell venni különböző megkötéseket is, ilyen lehet például a határidők betartása vagy a gyártásütemezéseknél gyakran előforduló raktárkészlet maximális kapacitásának betartása.

Az ütemezési feladatokat számos tulajdonság alapján lehet kategorizálni. Egyik legfontosabb tényező, melyet figyelembe kell venni, hogy determinisztikus vagy sztochasztikus az adott probléma. Az előbbi feladatok esetén minden paraméter előre

ismert, meghatározott. Ilyen problémára adott példát 1977-ben Mario J. Gonzalez, Jr. [3] publikációja, ahol egy processzor ütemezési problémát vizsgált meg. A sztochasztikus ütemezési problémákra egy áttekintést ad Pinedo [4] könyvfejezete. Ezen problémák esetén a paraméterek valószínűségi változók is lehetnek. Ilyen tipikusan a termékek iránti kereslet bizonytalansága vagy a hibalehetőségek száma egy termelési folyamatban, mely problémák kezelésére többek között J. Mula és tsai. [5] tanulmánya nyújt átfogó irodalmi áttekintést.

A szakirodalomban számos olyan optimalizálási megoldó módszer létezik, melyekkel az ütemezési problémák hatékonyan megoldhatók. A megfelelő megoldásnak a kiválasztása során az előzőekben említett tulajdonságokon kívül elengedhetetlen szempont a feladat komplexitása, mérete. Kisebb méretű feladatoknál, vagy ahol fontosabb szempont az optimum megtalálása, mint a sebesség, gyakran alkalmaznak egzakt módszereket, melyek előnye, hogy minden esetben garantálják a legjobb megoldás megtalálását. Minél bonyolultabb vagy nagyobb egy feladat, annál több időt vesz igénybe az optimális megoldásnak a megtalálása, ami bizonyos esetekben kivitelezhetetlen a jelenkor eszközeivel. Ennek elkerülése érdekében sokszor alkalmaznak a nagyobb feladatoknál úgynevezett heurisztikákat. Az ilyesfajta eljárások nem garantálják a lehető legjobb megoldást, hanem elsősorban közelítik azt. Ezenfelül ezek az approximációs megoldások sokkal kedvezőbb futási idővel rendelkezhetnek, szemben az egzakt módszerekkel, melyek elsősorban a kevésbé komplex feladatnál lehetnek előnyösebbek, ahol az optimum megtalálása az elsődleges cél. Gyakran előfordulhat, például egy online ütemezés során, hogy rövid időn belül szükségünk van egy megoldásra és ezen esetben is jobb választás lehet egy heurisztikus közelítés, mely általában gyorsabban biztosít jobb eredményt, mint egy egzakt megoldás.

Az egzakt módszerek esetén az egyik leggyakoribb módszertan a vegyes-egész lineáris programozás (Mixed Integer Linear Programming - MILP). A gyártásütemezési problémáknál alkalmazott MILP modellezési technikákról részletesen Méndez és tsai. [6], illetve újabban Harjunkoski és tsai. [7] írnak. Az ütemezésekre használt MILP modelleket két nagy csoportba lehet sorolni, az egyik az időrács (time-grid) alapú modellek [8], illetve a másik csoport a precedencia alapú modellek [9]. A lineáris programozási módszereken kívül gyakran alkalmaznak kombinatorikus módszereket, ilyen például az S-gráf módszertana, melyet Sanmartí és tsai. [10, 11] dolgoztak ki. Az S-gráf modell olyan gráf reprezentáción alapul, mely kihasználja az ütemezési feladatok speciális tulajdonságait, ezáltal hatékony megoldó algoritmusok

kidolgozását teszi lehetővé [12, 13]. A lineáris programozási modellekkel ellentétben itt nem paraméterekből és változókból készül a modell, hanem csúcsokból és súlyozott, irányított élekből. Ezenfelül számos heurisztikus megoldással is találkozhatunk a szakirodalomban, mint például az evolúciós algoritmusok, valamint a lokális keresési technikák alkalmazása. Az evolúciós algoritmusok használata egy elterjedt módszertan, amelynek működési alapja az evolúció darwini elméletében gyökerezik, ami szerint a természetben a folyamatosan változó életkörülményekhez való alkalmazkodás a fő szempont. Ezt a megközelítést [14, 15, 16], valamint a lokális keresési módszereket [17, 18] is számos kutatásban alkalmazták ütemezési problémákra.

Az ütemezés kutatási területén belül, egy nagyon fontos kategória a gyártórendszerek ütemezése. A termelési rendszerekben léteznek folyamatos és szakaszos folyamatok. A legfőbb különbség közöttük az, hogy a bemenetük/kimenetük hogyan kerül felhasználásra/előállításra. A szakaszos termelési rendszerek ellentétben a folyamatossal, a termékeket meghatározott véges adagokban dolgozzák fel. Mindkét típusnak hatalmas irodalma van és különböző modellezési technikákat igényelnek.

A gyártási feladatoknál egy másik fontos tényező az erőforrások és azok működése. Ha a termékek gyártása egy lépésből áll, akkor single stage problémáról van szó. A single stage problémák az alábbi alkategóriákba sorolhatók:

- **Single Machine:** Egyetlen gép elérhető csak, melyen minden feladatot el lehet végezni. A feladatoknak különböző feldolgozási ideje van.
- **Identical paralell machines:** m számú azonos gép áll rendelkezésre, melyeken bármelyik egylépéses feladat végrehajtható.
- **Paralell machines with different speed:** Hasonló az előző pontban említett típushoz, de itt minden gépnek meghatározott eltérő végrehajtási sebessége van.
- **Unrelated machines in paralell:** A Single stage általánosítása, ahol egy gépen végzett feladat feldolgozási ideje a bemeneti paraméter.

Több lépéses gyártási tervek esetén jellemzőek a szekvenciális folyamatok, melyek közül a legismertebbek a következők:

- **Flow shop:** A feladatoknak m lépése van, melyeket az m gépeken kell végrehajtani, minden feladat esetén ugyanabban a sorrendben.

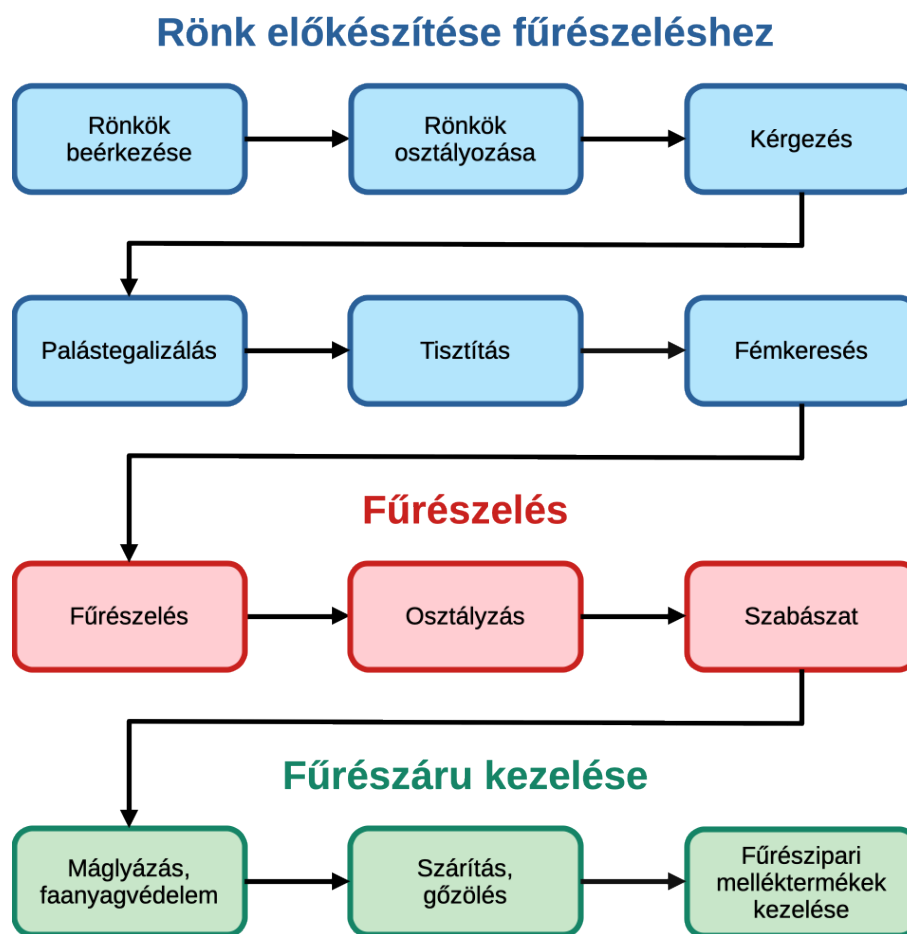
- **Flexible flow shop:** Az Flow shop és a Identical paralell machines változatának az általánosítása, ahol minden egyes feladatnak c szakaszon kell ugyanabban a sorrendben végig mennie, ahol több párhuzamos gép áll rendelkezésre.
- **Job shop:** Az Flow shop általánosítása, ahol a feladatoknak különböző sorrendje lehet a gépeken, és nem szükséges egy feladatnak az összes gépen végig mennie.
- **Flexible job shop:** A Job shop és a Flexible flow shop általánosítása, amikor a feladatoknak az szakaszokon a feladattól függő sorrendben kell végig menniük, és minden szakaszban néhány azonos gép áll rendelkezésre.
- **Open shop:** Minden feladatnak végig kell mennie az összes gépen, de ezek sorrendje nincs előre meghatározva.

Számos cikk található a szakirodalomban, melyek az erdőgazdálkodás és az ahhoz tartozó tevékenységek optimalizálásával foglalkoznak. Ebben a témában nyújt áttekintést Weintraub és tsai. által írt cikk [19]. Ennek egy válfaja a faipari termelési folyamatok ütemezése. Ide tartozhat egy fahulladék feldolgozó üzem ütemezése is, melyre Dávid és tsai. [20] adnak egy hatékony megoldást. A jelen dolgozat témáját a fűrészüzemek képzik, mely szintúgy fontos területe a faipari termelésnek.

2.2. Fűrészüzemek

A faipari termékek gyártását egy összetett feldolgozási folyamat előzi meg, melynek a fa kivágása utáni első fázisa a fűrészüzemekben a farönkök megmunkálása. Az üzemek célja, hogy a beérkezett rönkökből a keresletnek eleget tevő fűrészáru keletkezzen. Az iparban nagyjából minden fűrészüzem más specifikációkkal rendelkezik, köszönhetően a technológiai fejlődésnek és a folyamatosan változó keresletnek. Léteznek nagyobb, jelentős mértékben automatizált fűrészüzemek, de találkozhatunk kisebb, kevésbé gépesített üzemekkel is, ahol más szempontok alapján zajlik feldolgozás. Az üzemek működését jelentősen befolyásolja még azok földrajzi elhelyezkedése is, mivel a különböző területeken található erdők eltérő faállománnyal rendelkeznek. Ezek a különböző fatípusok pedig eltérő feldolgozási módszereket igényelnek.

A fűrészüzemek általános folyamatát a 2.1. ábra szemlélteti.



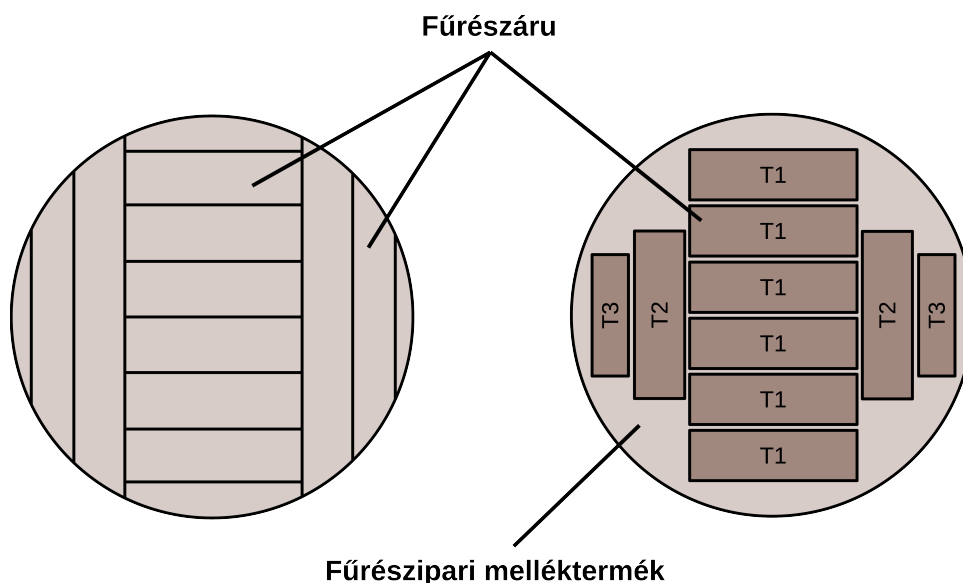
2.1. ábra. Fűrészüzemek általános feldolgozási folyamata

A különböző rönkökből másképpen és másfajta végtermékek állíthatók elő, ezért azokat a beérkezést követően osztályokba sorolják olyan tulajdonságaik alapján, mint például fafaj, átmérő, hossz vagy minőség. A fűrészelés előtt egy előkészítési eljárásen esnek át a farönkök. Az első ilyen lépés a kérgezés, ahol a kérget távolítják el a fáról. Ezt követi a palástegalizálás, melynek a célja a farönk durva külső felületének az eltávolítása, ami a felület egyenetlenségeinek csökkentésével elősegíti a fűrészelhetőséget. A tisztítás során eltávolítják a fafelületen lévő port, homokot és az egyéb szennyeződések. Fontos még figyelembe venni a faanyagban esetlegesen megbúvó fémrészeket, melyek könnyen okozhatnak sérüléseket a faanyagnak és az üzem gépeinek, ezért a rönköket át kell vizsgálni egy fémkereső géppel.

A fűrészelés során a rönköket hosszirányú vágással alakítják falécekké. A vágást különféle gépekkel végezhetik, melyeknek két leggyakoribb típusa a keretfűrész és a szalagfűrész. A két rönkvágó technológia között számos különbség van. A keretfűrészgép, avagy a gatter egy nagy teljesítményű alapgép, mely gyorsabb vágást biztosít a szalagfűrészekhez képest, azonban fix vágási mintával dolgozik, melyet csak a gép átállítással lehet megváltoztatni. Így különösen fontos a megfelelő vágási mintáról való döntés, hiszen egy ilyen átállás plusz idővel jár. Ezenfelül általában jóval több humán erőforrást igényel egy ilyen technológia működtetése, mint a szalagfűrész esetén, mivel az egész folyamatot felügyelő szakemberen túl szükség van külön segéd munkásokra a feldolgozott fűrészárak, esetleg a rönkök mozgatásához. Ez az emberi erőforrás igény jelentős tényező lehet a térségeinkben megtalálható fűrészüzemekben, ahol a jelenkor egyik legnagyobb kihívása a munkaerőhiány. Egy szalagfűrészgép a rugalmasabb feldolgozás mellett jóval lassabb vágási sebességet biztosít, így tömegtermelésre kevésbé alkalmas. Előnye viszont többek között a vékonyabb szerszám használat, és az ezzel együtt járó kisebb fűrészelési résvesztés, vagy a kisebb energia-felvétel. Ezenfelül sok más technológia tényező is szerepet játszik a fűrészelési folyamatban, melyről részletesebben Hargitai [21] ír.

A különböző vágási minták különböző fűrészáru kombinációkat és fűrészipari melléktermék mennyiségeket eredményeznek. A legjobb minta meghatározásához fontos figyelembe venni, hogy melyik rönkosztályból kerülnek a rönkök feldolgozásra. A [22, 23, 24] kutatások megvizsgálták azt a problémát, hogy hogyan lehet kiválasztani az optimális vágási mintákat a különféle típusokhoz. [25, 26] cikkek megoldást is javasolnak erre a kérdésre. A rönkvágási módok közül az egyik legegyszerűbb az élesvágás, ahol a rönkből egyetlen vágással állítják elő a szélezetlen fűrészárakat.

Ezenfelül gyakran alkalmaznak bonyolultabb vágásokat is, ahol többször feldolgozzák az adott rönköt a fűrészüzemi alapgépen. Ilyen például a prizma vágás, ahol a rönköt prizma alakúra vágják, majd ezután a prizmából állítják elő a fűrészáruk egy részét, melyek közül a két szélső darabon kívül szélezetlen áruvá válnak. Erre egy példát mutat be a 2.2. ábra.



2.2. ábra. Prizma vágás egy példája

Az alapgépeken történő vágás után az áruk a szabászaton is áteshetnek, ahol méretre vágják, így a feldolgozott fűrészárukat a keresletnek megfelelő méretűre alakíthatják. A 2.2 ábrán a jobb oldali mintán látható, ahogy a szélezetlen fűrészáruk is méretre lettek vágva. Az ilyesfajta műveletekhez különféle technológiákat alkalmaznak, ilyen például a sorozatvágó körfűrészgép. A fűrészelés után a készterméket méret és minőség alapján osztályozzák. Az utóbbit meghatározhatják például elszíneződések, gombakárosítás, rovarkárosítás, repedések vagy fűrészelési hibák alapján. A legutolsó fázisa az eljárásnak a fűrészáruk kezelése. A raktározás egyik általános módszere a termékek máglyákban történő tárolása. A faanyagok áteshetnek egy gőzölési folyamaton, melynek célja a csírátlantítás, a mechanikai tulajdonságok javítása, a fertőzések megszüntetése, a szárítási eljárás elősegítése, illetve ennek hatására a faanyag színe sötétedik, így esztétikailag is javítja a fűrészárut. A szárítás történhet a természetes módon a szabadlevegőn, de megvalósítható mesterséges szárítására alkalmazott szárítóberendezésekkel is. Ezenfelül a fűrészüzemeknek nem szabad figyelmen kívül hagyni az eljárás során keletkező fűrészipari melléktermék kezelését

sem.

A faanyagok feldolgozásnak ezen fázisa rendkívül összetett, sok döntéssel találkozhatunk benne, amit fontos mindig előre megtervezni, hogy hatékony működést és a magas minőségű fűrészárukat érjen el a fűrészüzem. A feldolgozás optimalizálása több kategóriára bontható, melyek közül az egyik legfontosabb a folyamatok megfelelő ütemezése.

2.3. Fűrészüzemek ütemezési problémái

Az erdészeti ellátási lánc esetén általában három döntéshozatali szint határozható meg, pontosabban a stratégiai, taktikai [27] és az objektív tervezés. A stratégiai döntéseket hosszú távra kell meghozni, ami jelenthet akár több évtizedet is. Ennél a szintnél olyan célokról beszélhetünk, mint például az erdei ökoszisztéma megőrzése vagy a fenntartható erdőgazdálkodás. A taktikai tervezésnél középtávú tervezésről beszélhetünk, melyek tipikusan néhány évet, hónapot jelentenek. Ilyen problémát vizsgálnak meg Pamela P. Alvarez és tsai. a cikkükben [28]. Objektív tervezésnél valós idejű, napi, heti, vagy akár egy hónapos tervezéssel foglalkozunk. Ezek általában igen részletes megoldásokat jelentenek. Az ilyen fűrészüzem ütemezési terveknel a legtöbb kutatás arra összpontosít, hogy miként lehet meghatározni a farönkök feldolgozásának a legjobb módját a termelés optimalizálása érdekében.

A fűrészüzemi ütemezési megoldások során az egyik leggyakoribb technika, a matematikai programozás modellek alkalmazása. Sergio Maturana és tsai. cikkükben [29] bemutatnak egy determinisztikus matematikai modellt egy chilei fűrészüzem ütemezésére, melynek a teljesítményét egy heurisztikával hasonlítják össze. A megoldás során 6 hetes periódusokkal dolgoztak, melyeket napi szintű részperiódusokra osztottak. A modell célfüggvénye a megrendelések elmaradásával, illetve a raktárkészletben maradó plusz termékekkel járó költségeket minimalizálja, illetve figyelembe veszi, hogy a rönkök kiválasztásánál elsősorban a jobb minőségűek kerüljenek feldolgozásra. Több teszteset összevetésében a cikk megmutatta, hogy a matematikai modell a hatékonyabb megoldás a vizsgált problémára.

Több szakirodalomban javasolt megoldás feltételezi, hogy az összes adat ismert, ami a valóságban sok esetben nem igaz. Az optimalizálási modellek során és a fűrészüzemek ütemezésénél is gyakran előjön a bizonytalanság kezelésének a problémája. Az irodalomban általában különféle megközelítéseket használnak az ilyesfajta

problémára, például alkalmazhatnak érzékenységvizsgálatot, ami a döntési modellek kiértékelését jelenti, azaz megmutatja az eredmények változását a bemeneti paraméterek változtatásával. Megoldás lehet a sztochasztikus programozás, ahol azt feltételezzük, hogy a bizonytalan paraméterek valószínűségi eloszlása ismert vagy pontosan becsülhető. A megközelítés hátránya, hogy nagyon sok adatot igényel, és sokszor közülük néhányhoz nehéz hozzájutni, ezenfelül a problémák gyakran megoldhatatlanok, vagy túl sok időt igényelnek. Népszerű megközelítés még a robusztus optimalizálás, mely bizonytalansági tartományok alapján kezeli a bizonytalanságot, így kevesebb adatot és számítási erőforrást igényel, mint a sztochasztikus programozás. Mauricio Varas és tsai. 2014-ben [30] bemutattak egy megoldást a chilei fűrészüzem problémájára, mely kezeli a farönkök készletének vagy a keresletnek a bizonytalanságát. Ez a modell Bertsimas és Sim [31] által kidolgozott megközelítésen alapul. A bizonytalanságot más kutatások is figyelembe veszik robusztus optimalizálással, például Masoumeh Kazemi Zanjani és szerzőtársainak [32] a megoldása a folyamatok hozamának és a termékek iránti keresletnek a bizonytalanságát kezeli. Ezenfelül a kutatás eredményei arra is utalnak, hogy a robusztus optimalizálási megközelítés bizonyos szituációkban előnyöket jelenthet a kétlépcsős sztochasztikus programozási megoldásokkal szemben.

A témában lévő kutatások legtöbbje az egy feldolgozási folyamat ütemezését vizsgálja. Pamela P. Alvarez és Jorge R. Vera [33] olyan megoldást mutat be, mely több fűrészüzem folyamatát tervezi meg párhuzamosan. A feldolgozás ütemezése többek között kezeli az esetleges szárítást vagy az újrafeldolgozást és ennek költséget is.

A szakirodalomban több cikkben figyelembe veszik a különböző vágási minta és rönk kombinációkat, azonban ezekre a vágási mintákra külön döntést nem fogalmaznak meg a modellekben. Azok csak a feldolgozás hozamának paraméterét befolyásolják. 2018-ban Nicolás Vanzetti és tsai. [34] mutattak olyan modellt, mely figyelembe veszi a vágási minták beállításának idejét és költségét. 2019-ben Diego Broz és szerzőtársaik a cikkükben [35] a célprogramozás módszerét alkalmazták az előbbi problémára, azonban míg Vanzettiék [34] 5 napos periódusra határozták meg, ebben a cikkben egy napos ütemezést valósítottak meg. A megoldásukban figyelembe vették a rönkök különböző feldolgozási idejét, melyet a legtöbb másik irodalomban található módszer figyelmen kívül hagy, vagy csak egyetlen konstanssal írja le.

3. fejezet

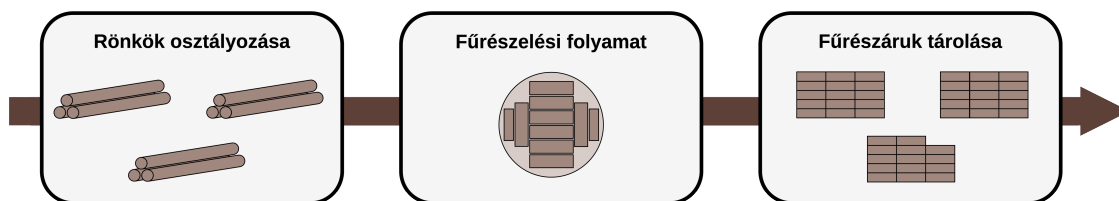
Probléma definíció

Annak érdekében, hogy egy olyan megoldást biztosíthassunk, mely közelebb áll az ipari szektor kihívásaihoz, mint az irodalmi megoldások, olyan tényezőket kell figyelembe venni, melyek gyakran fűrészüzem specifikus kérdések. Ezeket a specifikus kérdéseket különböző faiparral foglalkozó szakemberrel konzultálva határoztam meg. Így a szakirodalomban lévő megoldások, illetve az előbb említett diskurzusok alapján fogalmaztam meg a probléma definíciót mind informálisan a 3.1 szövegrészben, mind formálisan a 3.2 alfejezetben.

3.1. Informális probléma definíció

A fűrészüzem ütemezése több napos tervezésként valósul meg. Továbbá adott egy műszakhossz, mely alatt történik a fűrészüzemi feldolgozás. Megoldásom ezen fűrészüzemi feldolgozás folyamatának az átgondolásával kezdtem. Az ütemezés szempontjából a műveletek egyik legkritikusabb pontja a fűrészelési lépés. Ebből kifolyólag bizonyos folyamatok elhagyhatók, melyek nem befolyásolják jelentősen az ütemezés hatékonyságát. A feladat során feltételezzük, hogy adott a rönkkészlet, illetve nem foglalkozunk a fűrészáruk raktározásával, mint ütemezendő folyamat. Ezt az eljárást a 3.1 ábra szemlélteti.

A farönkök az egész tervezési időszakra érkeznek, így a feldolgozás során ezzel kell gazdálkodni. A beérkezést követően a legfontosabb lépés a rönkök típusok alapján történő csoportosítása. A feladat megoldásához el kell különíteni azokat a farönk osztályokat, melyeket a fák fizikai tulajdonságai alapján határoznak meg. A vizsgált problémaosztályban feltételezzük, hogy egyetlen fafajjal foglalkozunk, vala-



3.1. ábra. Az ütemezendő folyamat

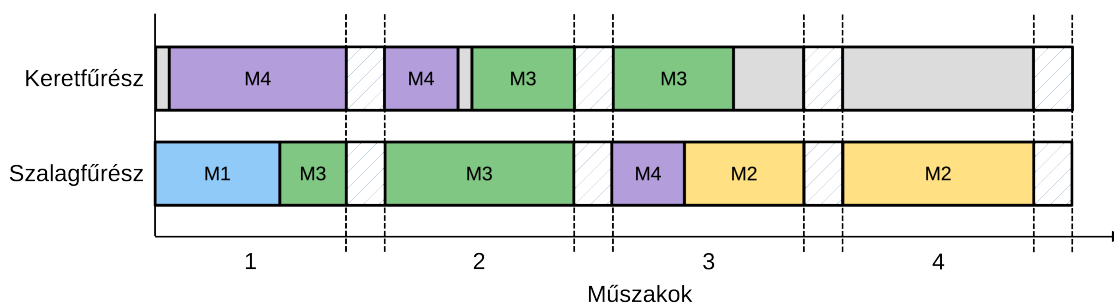
mint a rönkök hossza is megegyezik, így az átmérőjük alapján kerülnek különböző osztályokba. Ezen csoportok alapján definiálhatóak a feldolgozást befolyásoló paraméterek.

A legtöbb eddigi kutatásban javasolt megoldások elsősorban a nagyobb automatizált fűrészüzemek igényeihez igazodnak. Így a feladatosztály megfogalmazásánál a célom egy olyan fűrészüzem reprezentálása volt, ahol a rönkök fűrészelését a jóval egyszerűbb technológiák végzik. A feladat során két rönkvágó alapgép áll rendelkezésre az üzemben, egy keretfűrész és egy szalagfűrész. A munkám során szeretnék különös hangsúlyt fektetni a két technológia között lévő különbségekre, mivel a két megoldás jelentősen eltér. Így a feladat során egyik fő döntés az, hogy mikor mely gépet üzemeltetjük és hogyan.

Egy fontos tényező továbbá a vágási mintákról való döntés, melyről sok irodalmi megoldás feltételezi, hogy az adott esetben a leghatékonyabb van használatban. Ezenfelül nem foglalkoznak azzal, hogy egy keretfűrészgépen a minta váltása sokszor jelentős átállási időt eredményez. A probléma esetén minden rönk típushoz különböző vágási mintákat definiáltam, melyeket a fűrészelési folyamat során alkalmazhatunk. A gyakorlati tapasztalatok alapján feltételezzük, hogy egy műszak alatt csak egy vágási minta csere történhet a keretfűrészgépen. Ezenfelül figyelembe vesszük, hogy ha egy műszak kimarad, akkor az üres nap alatt megtörténhet az átállítás egy másik vágási mintára, így, ha váltás történik ez idő alatt, akkor az azt követő napon nem számolunk az azzal járó plusz idővel. Szalagfűrész esetében a minták közötti váltás a gyakorlatban nem jár számottevő késleltetéssel, így ezen a gépen tetszőleges számú váltás megengedett egy nap során.

A fűrészelési folyamatok ütemezésére egy példamegoldás a 3.2 Gantt diagramon látható, ahol 4 műszakra történt a tervezés. Azt, hogy mikor, milyen mintával történik az adott gépen a feldolgozás az M1, M2, M3 és az M4 jelöli, melyeket eltérő színekkel ábrázoltam. Szürke színnel jelöltem a gépek üresjáratát, melyek a minták

közi átállási időt is magába foglalja.



3.2. ábra. A fűrészüzem ütemezésének egy példája

A folyamat alatt csak szélezetlen fűrészárut gyártunk élesvágás technológiával. Így minden rönk csak egyszer kerül feldolgozásra mindkét rönkvágó alapgépre vonatkozóan. A két technológia esetén a fűrészelési idők is különbözőek, így a rönk átbecsátóképessége is eltérő a két gépnél. Ezenfelül minél több fűrészárut eredményez egy vágási minta, annyival tovább tart ez a fűrészelés, hiszen egy szalagfűrész esetében ennyiszor kell végig fűrészelni a rönköt, illetve a keretfűrésznel lassabb előtolási és szerszámsebességet kell alkalmazni a több fűrészlap esetén.

Az automatizált üzemekkel szemben a kisebb fűrészüzemekben jelentős kihívást jelent az eszközök üzemeltetéséhez szükséges humán erőforrás. Két képzettséget különböztetünk meg az üzem alkalmazottjaira vonatkozóan, melyek a szakemberek és a segéd munkások. A két rönkvágó alapgép eltérő emberi erőforrás igényekkel rendelkezik. Ahhoz, hogy döntéseket hozzassunk, hogy mikor melyik alkalmazott, melyik gépen dolgozzon, definiálni kell a dolgozók beosztását. Előfordulhat, hogy több szakember áll rendelkezésre, mint amennyi az adott gép vagy a gépek működtetéséhez szükséges lenne. Ebben az esetben a szakemberek segéd munkásként is dolgozhatnak, azonban ez fordítva nem történhet meg, mivel egy segéd munkás nem tudja ellátni a szakember feladatait. A feladatosztály megengedi továbbá, hogy extra segéd munkásokat alkalmazzunk. Ezt egy plusz paraméter szabályozza, reprezentálva a munkaerőhiányt és az azzal járó nehézségeket.

A tervezési időszak minden napjára vonatkozóan specifikált mennyiségű fűrészárú előállításra van szükség. Amennyiben egy adott napon túltermelés történik, a felhalmozott fűrészárut egy raktárban tárolhatjuk el későbbi felhasználásra. A célfüggvény megfogalmazása során szintén figyelembe vettem az emberi munkaerőhiány problémáját. Mivel az üzem szeretné minden nap teljesíteni a keresletet, de

ha nem tud minden gépet üzemeltetni, akkor gyakran hiánnyal kell számolnia. Így a cél ennek a hiánynak a minimalizálása, melyhez késedelmi díjat rendelünk.

3.2. Formális probléma definíció

A problémaosztály az alábbiakban felsorolt halmazokkal és paraméterekkel definiálható formálisan.

3.2.1. Halmazok

$\mathcal{D} = \{1, 2, \dots, |\mathcal{D}|\}$ a napok halmaza.

$\mathcal{L}$ a farönk osztályok halmaza.

$\mathcal{T}$ a fűrészáru termékek halmaza.

$\mathcal{P}$ az összes vágási minták halmaza.

$\mathcal{P}_l \subseteq \mathcal{P}$ a vágási minták halmaza minden $l \in \mathcal{L}$ farönk osztályra.

3.2.2. Paraméterek

Általános paraméterek

$H \in \mathbb{R}^+$ a műszak hossza. $[s]$

$V_l \in \mathbb{R}^+$ a térfogat minden $l \in \mathcal{L}$ farönk osztályra. $[m^3]$

$C \in \mathbb{R}^{0,+}$ a raktár kapacitása. $[m^3]$

Fűrészelési technológiákhoz köthető paraméterek

$l_p \in L$ a $p \in \mathcal{P}$ vágási mintához tartozó farönk osztály.

$ST_p^F, ST_p^B \in \mathbb{Z}^+$ a keretfűrész és a szalagfűrész vágási ideje minden $p \in \mathcal{P}$ vágási mintára. $[sec/db]$

$CT^F \in \mathbb{Z}^+$ a keretfűrészben a vágási paraméter cserélésével járó átállási idő. $[sec]$

Humánerőforrás-menedzsmenthez köthető paraméterek

$RR^{SP,F}, RR^{SP,B} \in \mathbb{N}_0$ a keretfűrész és a szalagfűrész működtetéséhez szükséges szakember száma. $[-]$

$RR^{AW,F}, RR^{AW,B} \in \mathbb{N}_0$ a keretfűrész és a szalagfűrész működtetéséhez szükséges segéd munkások száma. $[-]$

$HR_d^{SP} \in \mathbb{N}_0$ minden $d \in \mathcal{D}$ napon a rendelkezésre álló szakember száma. $[-]$

$HR_d^{AW} \in \mathbb{N}_0$ minden $d \in \mathcal{D}$ napon a rendelkezésre álló segéd munkások száma. $[-]$

$X^{AW} \in \mathbb{N}_0$ a maximálisan megengedett extra segéd munkások száma. $[-]$

Raktározáshoz és a kereslethez köthető paraméterek

$I_l \in \mathbb{R}^{0,+}$ a rendelkezésre álló rönkök száma minden $l \in \mathcal{L}$ farönk osztályra. $[-]$

$DM_{t,d} \in \mathbb{R}^{0,+}$ a kereslet minden $t \in \mathcal{T}$ fűrészáru típusra és $d \in \mathcal{D}$ napra. $[m^3]$

$Y_{t,p} \in [0, 1]$ a hozamarány minden $t \in \mathcal{T}$ fűrészáru típusra és $p \in \mathcal{P}$ vágási mintára. $[-]$

$LF_t \in \mathbb{N}_0$ a késedelmi díj minden $t \in \mathcal{T}$ fűrészáru típusra. $[HUF/m^3]$

4. fejezet

Megoldó módszer

Ebben a fejezetben az előzőekben definiált fűrészüzem ütemezési problémára javaslok egzakt megoldásokat. Munkám során két vegyes-egész lineáris programozási modell felírási módot különítek el. A szalag- és keretfűrészek eltérő működési jellegéből fakadóan azokat különböző módon modelleztem. Mivel szalagfűrész esetében nem jelent gyártási nehézséget vagy plusz időt a vágási minta cseréje, így annak részletes modellezése nem indokolt. Keretfűrész esetében ennek az ellentéte igaz, és a későbbiekben bemutatott modellek fő eltérései a keretfűrész működésének modellezésében különböznek egymástól. Egy általános ipari "best practice", hogy a keretfűrész hosszú átállítási ideje miatt a gyártási tervekben naponta legfeljebb egyszer szerepel ilyen. Az első modell esetében ez a csere a nap bármely pontján megengedett, azaz a vágási minta csere időpontokat folytonos értékeknek kezelem. Ezt a modellt a 4.1. alfejezetben részletezem. A gyártási terv könnyű végrehajtathóságának szempontjából azonban a gyakorlatban előnyösebbek lehetnek az olyan ütemezések, melyekben ezek a váltások jól definiáltan csak műszakok elején, felénél, egész órában, stb. történhetnek csak meg. Modellezési szempontból ez a műszakok egyenlő időegységekre való felosztását jelenti. Ezen diszkrét időrács alapú variánst a 4.2. alfejezetben mutatom be.

4.1. Folytonos váltási időpont alapú MILP modell

A következőekben taglalt megoldás alapját képezi a döntéshozatal arról, hogy amennyiben vágási mintát váltunk a keretfűrészben, akkor mikor valósuljon meg az. Ennek az időpontnak a meghatározásával befolyásoljuk, hogy mennyi ideig dolgoz-

hat egy műszak alatt a gép az adott vágási mintával, hiszen ez a változó osztja két egységre az adott nap termelési időszakát. A szalagfűrész ennél egyszerűbben modelleztem, ahol nem befolyásoló tényező a vágási minta csere időpontja. Ennél a technológiánál megengedett, hogy egy termelés alatt tetszőleges számú vágási minta csere történjen.

A megközelítem az alábbiakban taglalt változókból (4.1.1. alfejezet), korlátozásokból (4.1.2. alfejezet) és célfüggvényből (4.1.3. alfejezet) áll.

4.1.1. Változók

Az alábbi bináris változó modellezi, hogy egy nap dolgozzon-e a keretfűrész, illetve a szalagfűrész:

$$w_d^F, w_d^B \in \{0, 1\} \quad \forall d \in \mathcal{D} \quad (4.1)$$

A megoldásban feltételeztem, hogy egy műszak alatt maximum egy vágási minta csere történhet a keretfűrészben. Ennek az időpontját a következő változó rögzíti:

$$t_d \in [0, H - CT^F] \quad \forall d \in \mathcal{D} \quad (4.2)$$

Annak meghatározása érdekében, hogy mikor milyen mintával dolgozzon a keretfűrész, a műszakot két szakaszra bontom. Külön döntési változót rendelek a csere előtti és az utána lévő vágási minta kiválasztásához:

$$s_{d,p}^{F,-}, s_{d,p}^{F,+} \in \{0, 1\} \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.3)$$

A keretfűrész napi rönkfeldolgozására adott a következő változó:

$$q_{d,p}^F \in \mathbb{N}_0 \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.4)$$

$q_{d,p}^F$ a többi változó értékéből az alábbi egyenlettel lenne kiszámítható:

$$q_{d,p}^F = s_{d,p}^{F,-} \cdot \frac{t_d}{ST_p^F} + s_{d,p}^{F,+} \cdot \frac{H - t_d - CT^F}{ST_p^F} \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.5)$$

A 4.5 egyenletben azonban a $s_{d,p}^{F,-} \cdot t_d$ és $s_{d,p}^{F,+} \cdot t_d$ változók szorzata szerepel, mely a lineáris programozási modell esetén nem megengedett. A quadratikus probléma elkerülése érdekében külön változók kerültek bevezetésre vágási minta csere előtti és

utáni rönkfeldolgozásra:

$$q_{d,p}^{F,-}, q_{d,p}^{F,+} \in \mathbb{N}_0 \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.6)$$

A szalagfűrész napi rönkfeldolgozása:

$$q_{d,p}^B \in \mathbb{N}_0 \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.7)$$

Az adott napon alkalmazott extra segéd munkások száma:

$$x_d \in \mathbb{N}_0 \quad \forall d \in \mathcal{D} \quad (4.8)$$

Fűrészáru termelés minden napra vonatkozóan:

$$y_{d,t} \in \mathbb{R}^{0,+} \quad \forall d \in \mathcal{D}, t \in \mathcal{T} \quad (4.9)$$

Keresleten felül termelt és emiatt eltárolt fűrészárak minden napra vonatkozóan:

$$o(p)_{d,t} \in \mathbb{R}^{0,+} \quad \forall d \in \mathcal{D} \cup \{0\}, t \in \mathcal{T} \quad (4.10)$$

A termelésben keletkezett napi hiány a kereslethez képest:

$$u(p)_{d,t} \in \mathbb{R}^{0,+} \quad \forall d \in \mathcal{D}, t \in \mathcal{T} \quad (4.11)$$

4.1.2. Korlátozások

A keretfűrész napi működését az alábbiakban részletezett korlátozások szabályozzák, melyek a tervezési időszak minden napjára meghatározzák a gép aktivitását. A 4.12 korlátozás a vágási minta csere előtti keretfűrész munkafolyamatokat írja le, míg a 4.13 a csere utáni folyamatokat rögzíti. E két kifejezés határozza meg, hogy ha bármely vágási mintával működik a keretfűrész az adott napon, akkor a gép üzemel, illetve ha a gép üzemel, akkor pontosan egy vágási mintát használ a váltás előtt és után.

$$\sum_{p \in \mathcal{P}} s_{d,p}^{F,-} = w_d^F \quad \forall d \in \mathcal{D} \quad (4.12)$$

$$\sum_{p \in \mathcal{P}} s_{d,p}^{F,+} = w_d^F \quad \forall d \in \mathcal{D} \quad (4.13)$$

A műszak kezdésekor figyelembe kell venni, hogy mi volt az előző nap végén lévő

vágási minta. A 4.14 és a 4.15 korlátozások biztosítják, hogy amennyiben a keretfűrészgép két egymást követő munkanapon is üzemel, a műszak kezdetén azonos vágási mintát használjon, mint az előző munkanap végén. A feltételek bevezetésével az $s_{d,p}^{F,+}$ és $s_{d,p}^{F,-}$ változók értékei magukkal hordozzák azt az információt, hogy a keretfűrész az adott napon üzemel-e, ezáltal a későbbi termelésre vonatkozó számításoknál nem kell külön figyelembe venni w_d^F értékét, mely bizonyos esetekben a modell komplexitását növelné. Ez a kikötés azonban nem vonatkozik az olyan esetekre, ahol a két műszak között van olyan nap, amikor nem üzemel a gép. Ilyenkor feltételezhető, hogy a szünet alatt megtörténhet az átállás két minta között.

$$s_{d,p}^{F,+} \geq s_{d+1,p}^{F,-} - 1 \cdot (2 - w_d^F - w_{d+1}^F) \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.14)$$

$$s_{d,p}^{F,+} \leq s_{d+1,p}^{F,-} + 1 \cdot (2 - w_d^F - w_{d+1}^F) \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.15)$$

A keretfűrész rönkfeldolgozásának modellezésekor úgynevezett "big-M" korlátozásokat alkalmaztam, melyek figyelembe veszik, hogy a gép mikor milyen mintával dolgozik. Az $\mathbf{M}$ paraméter bevezetése helyett a $\frac{H}{ST_p^F}$ kifejezést használtam, mely az alábbi korlátozásoknál kellően nagy konstans, a relaxáció biztosítására. A napi vágási minta csere előtti rönkfeldolgozás kifejezésére a 4.16 és a 4.17 egyenletek szolgálnak, míg az átállás utáni feldolgozás számítását a 4.18 és a 4.19 korlátozások biztosítják. Adott napon előfordulhat, hogy nem történik vágási minta csere a keretfűrészben. Ekkor a $q_{d,p}^{F,+}$ vagy a $q_{d,p}^{F,-}$ változók valamelyike nulla értéket vesz fel.

$$q_{d,p}^{F,-} \leq \frac{t_d}{ST_p^F} + \frac{H}{ST_p^F} \cdot (1 - s_{d,p}^{F,-}) \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.16)$$

$$q_{d,p}^{F,-} \leq \frac{H}{ST_p^F} \cdot s_{d,p}^{F,-} \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.17)$$

$$q_{d,p}^{F,+} \leq \frac{H - t_d - CT^F \cdot (1 - s_{d,p}^{F,-})}{ST_p^F} + \frac{H}{ST_p^F} \cdot (1 - s_{d,p}^{F,+}) \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.18)$$

$$q_{d,p}^{F,+} \leq \frac{H}{ST_p^F} \cdot s_{d,p}^{F,+} \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.19)$$

A szalagfűrész technológia esetén a 4.20 korlátozás szabályozza a napi rönkfeldolgozást, kezelve, hogy adott napon üzemel-e az eszköz.

$$\sum_{p \in \mathcal{P}} q_{d,p}^B \cdot ST_p^B \leq H \cdot w_d^B \quad \forall d \in \mathcal{D} \quad (4.20)$$

A következőkben részletezett korlátozások kezelik az üzem humán erőforrás-

menedzsmentjét. Mind a szalagfűrész, mind a keretfűrész esetén teljesíteni kell a feltételt, miszerint az adott gép csak akkor üzemeltethető, ha tudunk elég emberi erőforrást biztosítani a működéséhez. Ezt a követelményt fejezi ki a 4.21 korlátozás a szakemberekre vonatkozóan, ahol figyelembe veszem az adott napon rendelkezésre álló dolgozók számát. A segéd munkások korlátját a 4.22 egyenlet írja le, mely során megengedett, hogy a rendelkezésre álló szakemberek is alkalmazhatók legyenek segéd munkásként. Ezenfelül a korlátozás lehetőséget biztosít extra segéd munkások behívására is.

$$RR^{SP,F} \cdot w_d^F + RR^{SP,B} \cdot w_d^B \leq HR_d^{SP} \quad \forall d \in \mathcal{D} \quad (4.21)$$

$$RR^{AW,F} \cdot w_d^F + RR^{AW,B} \cdot w_d^B \leq HR_d^{AW} + x_d + HR_d^{SP} - (RR^{SP,F} \cdot w_d^F - RR^{SP,B} \cdot w_d^B) \quad \forall d \in \mathcal{D} \quad (4.22)$$

Az extra segéd munkás műszakok maximális számát egy paraméter határozza meg, mely összességében szabályozza a teljes tervezési időszakban alkalmazható segéd munkások számát. A modell a 4.23 egyenlettel menedzseli a napok között a pluszban megengedett segéd munkásokat.

$$\sum_{d \in \mathcal{D}} x_d \leq X^{AW} \quad (4.23)$$

Az üzem az egész tervezési időszakra egy előre meghatározott rönkkészletből gazdálkodik. Ezt a 4.24 egyenlet kezeli, ahol a rönkvágó alapgépek feldolgozását szabályozom.

$$\sum_{d \in \mathcal{D}} \sum_{\substack{p \in \mathcal{P} \\ l_p = l}} (q_{d,p}^{F,-} + q_{d,p}^{F,+} + q_{d,p}^B) \leq I_l \quad \forall l \in \mathcal{L} \quad (4.24)$$

Annak kiszámításáért, hogy melyik nap mennyi fűrészáru hozam keletkezett, a 4.43 kifejezés felel. Itt a vágási mintához tartozó hozamokat összegzem minden gépre vonatkozóan.

$$y_{d,t} = \sum_{p \in \mathcal{P}} Y_{t,p} \cdot V_{l_p} \cdot (q_{d,p}^{F,-} + q_{d,p}^{F,+} + q_{d,p}^B) \quad \forall d \in \mathcal{D}, t \in \mathcal{T} \quad (4.25)$$

Minden tervezési napra adott egy kereslet, melyet az adott műszakban teljesítenie kellene a fűrészüzemnek. Muszáj számolni azzal, hogy egy adott műszak alatt hiány is keletkezhet a kereslethez képest. Ekkor pótolható a fűrészáru a raktárból,

amennyiben az előző napokban a kereslet igényei felett termelt az üzem. Ha erre nincs lehetőség, akkor a hiányként kell kezelni. Ennek korlátozásáért a 4.26 egyenlet felel.

$$y_{d,t} + o(p)_{d-1,t} - o(p)_{d,t} + u(p)_{d,t} \geq DM_{t,d} \quad \forall d \in \mathcal{D}, t \in \mathcal{T} \quad (4.26)$$

A modellben a 4.27 egyenlettel határoztam meg a tervezési időszakra vonatkozó raktár kezdeti állapotát, melyet alapértelmezetten minden fűrészáru típusra nulla értékre állítottam.

$$o(p)_{0,t} = 0 \quad \forall t \in \mathcal{T} \quad (4.27)$$

A fűrészárak raktározásának egy maximum kapacitás paraméterrel szabtam határt, melyet a 4.28 korlátozás vesz figyelembe minden napra vonatkozóan.

$$\sum_{t \in \mathcal{T}} o(p)_{d,t} \leq C \quad \forall d \in \mathcal{D} \quad (4.28)$$

4.1.3. Célfüggvény

A tapasztalat alapján feltételezhető, hogy nem tudja az üzem teljesíteni a megrendeléseket, így mindenképp hiány keletkezik, melyhez hiányköltséget rendelek. A modell futtatása során ennek a minimalizálása a cél, melyet a 4.29 formulával definiáltam.

$$\min \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} LF_t \cdot u(p)_{d,t} \quad (4.29)$$

4.2. Diszkrét időrács alapú MILP modell

Ebben az alfejezetben az előzőekben definiált megoldás (4.1. alfejezet) egy másik megközelítését mutatom be. A faiparral foglalkozó szakemberekkel folytatott konzultációk alatt azt az információt gyűjtöttem, hogy viszonylag gyakori jelenség a keretfűrészgépek működésénél, hogy egy vágási minta csere történik és az a műszakok felénél. Ennek a modellezése egyszerűbb, mint a előzőekben bemutatott folytonos időváltozó alapú modell. Továbbá ez az új megközelítés ebből a szempontból szűkebb megoldási térben keres optimális megoldást, ezért potenciálisabban kevésbé hatékony tervezést ad, de a számításigénye jóval kisebb. Ugyanezt a felosztást

alkalmazom a szalagfűrésznel is, ezáltal könnyebben modellezhetővé válik, hogy egy műszakon belül különböző időpontokban más felosztásba üzemeljenek az eszközök. Így a munkaerőforrások is átszervezhetővé válnak a műszakokon belül. A munkám során megvizsgálom, hogy érdemesebb lehet-e ilyen modellt alkalmazni, vagy a folytonos időváltozó alapú megközelítés jobb választás. Mivel modellezési szempontból mindegy, hogy a műszakot hány azonos méretű részre osztjuk, ezért egy általános modellt írtam fel, melyben tetszőlegesen paraméterezhető ezen műszakrészek száma. Ezzel a módszerrel a vágási minta csere időpontját diszkrét időpontokra korlátozom, melyek a műszakok elején történhetnek.

A modellek egyértelmű megkülönböztetése érdekében az előző folytonos időváltozó alapú modellt "C-EK"-val jelölöm, míg a diszkrét időváltozó alapú modellt "D[k]"-val, ahol a k reprezentálja, hány részre lett felosztva a műszak. Ezenfelül megkülönböztetek egy olyan variánst is, ahol figyelembe veszem azt a kikötést, hogy egy műszak alatt egy alkalommal történhet vágási minta csere a keretfűrészben. Ezt a modellt "D[k]-EK"-val fogom jelölni.

4.2.1. Modell változtatások

Először bevezetek egy új $\mathcal{K} = \{1, 2, \dots, |\mathcal{K}|\}$ halmazt, mely az egyenlő méretű műszakrészek halmazát jelöli. Így a keretfűrész esetén nem jelölöm külön változókkal a vágási minta csere előtti és utáni időszakot, hanem egy új dimenziót vezetek be a műszakrészekre vonatkozóan. Emellett a szalagfűrész műszakjait ugyanígy felbontom ilyen részekre, melyekre azért van szükség, mert az alkalmazottak műszakon belüli mozgásával nem biztos, hogy minden műszakrészben üzemel. A következőekben bemutatom, hogy ez alapján, hogyan változtattam a korábbi modellt.

Változók

Eddig a 4.1 változók jelölték azt, hogy mely napokon mely gépek üzemelnek. Az új modell esetében ezeket kibővítettem egy új index bevezetésével a műszakrészekre vonatkozóan:

$$w_{d,k}^F, w_{d,k}^B \in \{0, 1\} \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \quad (4.30)$$

A keretfűrészben történő vágási minta cserét a 4.3 változó segítségével modelleztem, ahol egyrészt újra kellett gondolni, hogy nincsenek vágási minta előtti és utáni

időintervallumok. Másrészt a rönkfeldolgozás kiszámítása során a quadratikusan kifejezés elkerülése érdekében, külön változót vezettem be, hogy az adott műszakrészben folytatódik-e a termelés ugyanazzal a vágási mintával, mint ahogy az előzőben történt, illetve külön változót arra, hogy átállással kezdődik-e az időszak:

$$s_{d,k,p}^F, s_{d,k,p}^{c,F} \in \{0, 1\} \quad \forall d \in \mathcal{D}, k \in \mathcal{K}, p \in \mathcal{P} \quad (4.31)$$

Míg eddig a keretfűrész rönkfeldolgozást is napi szinten modelleztem a 4.4 változóval, a következőkben ez a változó is műszakrészekre lesz elkülönítve:

$$q_{d,k,p}^F \in \mathbb{N}_0 \quad \forall d \in \mathcal{D}, k \in \mathcal{K}, p \in \mathcal{P} \quad (4.32)$$

Korlátozások

Az új diszkrét időrács alapú megközelítés alapján jelentős változást kellett végrehajtanom a keretfűrész működésére vonatkozó korlátozásokban. Az előző modellben a 4.12 - 4.19 egyenletekkel definiáltam a keretfűrész működését, melyek a jelenlegi modellben nem alkalmazhatóak. Az alábbiakban taglalom, hogy milyen korlátozásokat definiáltam a keretfűrész technológia aktivitásának kezelésére.

A 4.33 egyenlet biztosítja, hogy amennyiben az üzem úgy dönt, hogy az adott műszakrészben üzemeljen a keretfűrész, akkor pontosan egy vágási mintát használhat a gép. Ezenfelül garantálja, hogy vagy vágási minta cserével kezdődik az időszak vagy az előző műszakrészben használt mintával dolgozik tovább a gép. Pontosabban a korlátozás biztosítja, hogy ha üzemel a gép, akkor a két eset közül az egyik mindenképp megvalósuljon, de egyszerre a kettő sosem.

$$\sum_{p \in \mathcal{P}} s_{d,k,p}^{c,F} + s_{d,k,p}^F = w_{d,k}^F \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \quad (4.33)$$

Az ütemezés során feltételezhető, hogy az első munkanap a keretfűrészgép nem átállással kezdi a műszakot, melyet a 4.34 korlátozás biztosít.

$$s_{1,1,p}^{c,F} = 0 \quad \forall p \in \mathcal{P} \quad (4.34)$$

Külön megszorítást fogalmaztam meg a műszakrészek közötti kapcsolatokra. Pontosabban arra, hogy figyelembe kell venni az előző időszakban alkalmazott vágási mintát. A 4.35 egyenlettel biztosítom, hogy a keretfűrész akkor folytathatja egy vágási mintával való termelést, amennyiben az azt megelőző műszakrészben is

ugyanazzal üzemelt. Mindezt függetlenül attól, hogy az előző műszakrészben történt-e vágási minta csere. Ezenfelül, akkor sem kalkulál átállással, amennyiben az előtte lévő időszakban nem üzemelt a gép.

$$s_{d,k,p}^F \leq s_{d,k-1,p}^F + s_{d,k-1,p}^{c,F} + (1 - w_{d,k-1}^F) \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \setminus \{1\}, p \in \mathcal{P} \quad (4.35)$$

Az előző megszorításnál kivételt képeztek a napok első műszakrészei, ugyanis az első nap első műszakrészét a 4.34 korlátozással állítom be, illetve külön korlátozást vezettem be a többi napra vonatkozóan. A 4.36 egyenlettel a napi szintű vágási minta cserét modelleztem. A 4.35 korlátozással szemben itt a napok első műszakrészét viszonyítom az előző nap utolsó műszakrészéhez.

$$s_{d,1,p}^F \leq s_{d-1,|\mathcal{K}|,p}^F + s_{d-1,|\mathcal{K}|,p}^{c,F} + (1 - w_{d-1,|\mathcal{K}|}^F) \quad \forall d \in \mathcal{D} \setminus \{1\}, p \in \mathcal{P} \quad (4.36)$$

Az új gondolatmenet megköveteli a korábbi modellben felírt rönkfeldolgozás-hoz köthető a 4.16 - 4.19 korlátozások átdolgozását is. Jelen megközelítésben ez a 4.37 egyenletre egyszerűsödik, melyben figyelembe veszem minden műszakrészben a maximum feldolgozható rönkök mennyiségét, továbbá biztosítom a vágási minta cserékkel járó plusz idővesztéssel való kalkulációt.

$$q_{d,k,p}^F \leq \frac{\frac{H}{|\mathcal{K}|} - CT^F}{ST_p^F} \cdot s_{d,k,p}^{c,F} + \frac{H}{|\mathcal{K}| \cdot ST_p^F} \cdot s_{d,k,p}^F \quad \forall d \in \mathcal{D}, k \in \mathcal{K}, p \in \mathcal{P} \quad (4.37)$$

Annak érdekében, hogy kezelni tudjam a műszakrészek közötti emberi erőforrások átszervezési lehetőségét, a korábban 4.20 felírt egyenletet is újra kellett gondolnom. Ennek eredménye a 4.38 korlátozás lett, ahol figyelembe veszem, hogy a $w_{d,k}^B$ változó minden műszakrészre van definiálva és ez alapján összegzem a szalagfűrésznapi rönkfeldolgozását.

$$\sum_{p \in \mathcal{P}} q_{d,p}^B \cdot ST_p^B \leq \sum_{k \in \mathcal{K}} \frac{H}{|\mathcal{K}|} \cdot w_{d,k}^B \quad \forall d \in \mathcal{D} \quad (4.38)$$

A műszakrészekre bontás és a köztük lévő rugalmas erőforrás menedzsment magával vonja a 4.21 és 4.22 korlátozások átdolgozását is. A szakemberekre vonatkozó megszorítást a 4.39 egyenlettel modelleztem, míg a segéd munkásokra vonatkozót

a 4.22 korlátozással.

$$RR^{SP,F} \cdot w_{d,k}^F + RR^{SP,B} \cdot w_{d,k}^B \leq HR_d^{SP} \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \quad (4.39)$$

$$RR^{AW,F} \cdot w_{d,k}^F + RR^{AW,B} \cdot w_{d,k}^B \leq HR_d^{AW} + x_d +$$

$$HR_d^{SP} - (RR^{SP,F} \cdot w_{d,k}^F - RR^{SP,B} \cdot w_{d,k}^B) \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \quad (4.40)$$

Előzőleg meghatározott a 4.24 és a 4.43 korlátozások, melyek a rönkkészlet felhasználás szabályozásáért és a fűrészáru hozam kalkulációjáért felelősek, szintén bővítésre szorultak. Előbbinek a bővítése a 4.41 egyenletnél látható, míg utóbbi továbbfejlesztet változata a 4.42 korlátozás.

$$\sum_{d \in \mathcal{D}} \sum_{\substack{p \in \mathcal{P} \\ l_p = l}} (q_{d,p}^B + \sum_{k \in \mathcal{K}} q_{d,k,p}^F) \leq I_l \quad \forall l \in \mathcal{L} \quad (4.41)$$

$$y_{d,t} = \sum_{p \in \mathcal{P}} Y_{t,p} \cdot V_{l_p} \cdot (q_{d,p}^B + \sum_{k \in \mathcal{K}} q_{d,k,p}^F) \quad \forall d \in \mathcal{D}, t \in \mathcal{T} \quad (4.42)$$

Annak kiszámításáért, hogy melyik nap mennyi fűrészáru hozam keletkezett, a 4.43 kifejezés felel. Itt a vágási mintához tartozó hozamokat összegzem minden gépre vonatkozóan.

$$y_{d,t} = \sum_{p \in \mathcal{P}} Y_{t,p} \cdot V_{l_p} \cdot (q_{d,p}^{F,-} + q_{d,p}^{F,+} + q_{d,p}^B) \quad \forall d \in \mathcal{D}, t \in \mathcal{T} \quad (4.43)$$

A $D[k]$ és a $D[k]$ -EK modell között az a következő korlátozás bevezetése a különbség. Mivel a korábbi modellben és a 3 fejezetben meghatározott probléma definiációban is az szerepel, hogy egy műszakban egyetlen vágási minta csere történhet, a következő 4.44 korlátozással biztosítom ezt a megszorítást.

$$\sum_{k \in \mathcal{K}} s_{d,k,p}^{c,F} \leq 1 \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \quad (4.44)$$

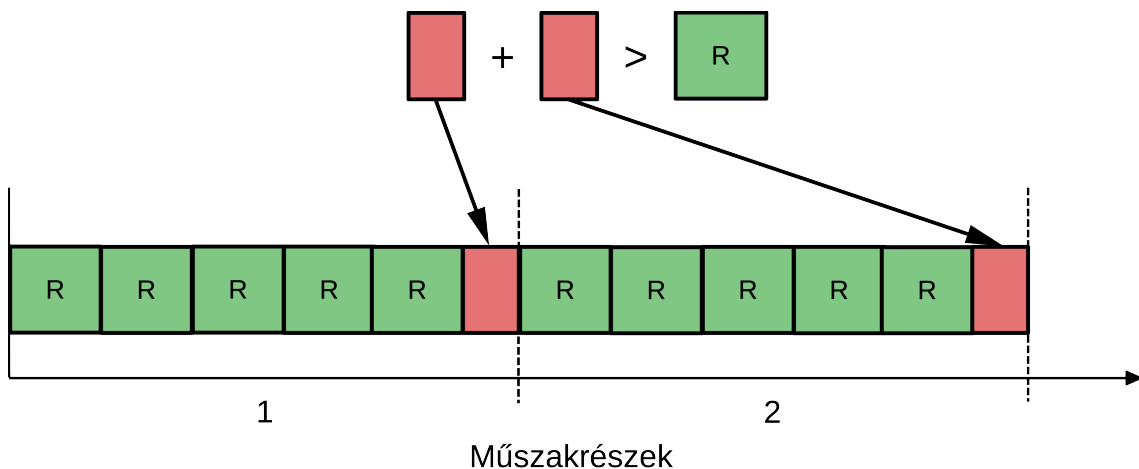
Az ipari tapasztalat alapján ez a bevett gyakorlat, azonban a későbbiekben megvizsgálom, hogy ennek a megkötésnek az elhagyása milyen hatással van a megközelítés hatékonyságára.

4.2.2. Átcúsúzási idő kezelése

Az előzőekben a 4.2.1. alfejezetben bemutatott modell könnyen skálázható 2 helyett több műszakrészre is, azonban a tesztek során megmutatkozott, hogy ezekben

az esetekben előfordul a következő jelenség.

A keretfűrész esetén jelentős változást hozott a műszakok egyenlő részekre való felosztása. A modellnél számolni kell azzal, hogy a műszakrészek végén maradhatnak törtidők, melyek nem elegendőek egy rönk feldolgozásához külön-külön. Ez egyrészt problémát jelenthet, egy olyan szélsőséges esetben, ahol olyan kis méretű egy műszakrész időtartama, hogy nem képes a gép egyetlen rönk feldolgozására sem. Ez nem egy valószínű eset, de ezenfelül gyakrabban előfordulhat, hogy két egymást követő azonos vágási mintával üzemelő műszakrész végén lévő üres idők összege meghaladhatja egy rönk feldolgozásának idejét. Amennyiben ugyanaz a vágási minta van ezekben a kisebb műszakrészekben, akkor több ilyen kis rész van, melyek külön-külön nem elegendők egy rönk felvágásához, de gyakorlatban ilyenkor a gyártás folytonos, így ezek összeadhatók. A modellben ezért szeretném megengedni, hogy egy az egy műszakrészből ez a törtidő "átcsúsztható" legyen a következőre. Az így átvitt időt fogom a továbbiakban átcúszási időként nevezni. Erre mutatok példát a 4.1. ábrán, ahol a zöld téglalapok reprezentálják a keretfűrészben feldolgozandó rönköket és a piros téglalapok a műszakrészben lévő üres járatokat.



4.1. ábra. Átcúszási idő számolásának egy példája

Az ábrán látható, hogy a műszakrészek végén lévő kihasználatlan időintervallumok összege, nagyobb mint egy rönk feldolgozásával járó idő. Tehát ennek az időnek a hozzáadása a gép üzemelési idejéhez, megengedné legalább egy további rönk feldolgozását. A gyakorlatban ez azt jelentené, hogy a műszakrész vége nem korlátozza a termelést, hanem egyik műszakrészben elkezdődhet egy rönk feldolgozás, mely átcúszhat a következő műszakrészbe is. Ezért a modellben bevezettem egy új változót,

mely a műszakrészek között így az átcúsztatott időt reprezentálja:

$$\sigma_{d,k,p} \in [0, \frac{H}{|\mathcal{K}|}] \quad \forall d \in \mathcal{D}, k \in \mathcal{K}, p \in \mathcal{P} \quad (4.45)$$

A következő 4.46 korlátozás segítségével kerülöm el, hogy a modell nagyobb részt is átcúsztatási időként kezeljen, mint amennyi az egy rönk vágási ideje. Ezenfelül kezelem, hogy csak, akkor számoljunk a változóval, ha az aktuális műszakrészt követő részben is megegyező vágási mintával folytatódik a termelés.

$$\sigma_{d,k,p} \leq ST_p^F \cdot s_{d,k+1,p}^F \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \setminus \{|\mathcal{K}|\}, p \in \mathcal{P} \quad (4.46)$$

Ez a plusz idő a keretfűrész rönkfeldolgozás számítása során használható fel, melyet eddig a 4.37 egyenlettel számoltam. A következőekben a célom, hogy ezt az extra időt is beleszámoljam abba az időbe, mely alatt a gép üzemel. Ezt két esetre bontottam, ahol az első jelenti a munkanapok első munkarészét, míg a második az összes többi műszakrészt. A műszakrészek elején az átcúsztatási idő levonásra kerül biztosítva ezzel, hogy a feldolgozási idő ne haladja meg a rendelkezésre álló időt. Ekkor ugyanis nincs előző műszakrész, ahonnan ez a plusz idő áthozható lenne. Ezt garantálja a 4.47 egyenlet. A többi egységre vonatkozóan, viszont figyelembe vehető az előző műszakrész végén megmaradt extra üres idő is. Ennek módját a 4.48 egyenlet írja le, ahol az időt, melyet szeretnénk az adott műszakrésztől a következő részre csúsztatni, az levonásra kerül, míg ami az előzőből maradt, az hozzáadódik. Ez adja meg a fűrészelésre rendelkezésre álló időintervallumot, melyet a műszakrészhez rendelt munkamennyiség és feldolgozási idő nem haladhat meg.

$$\begin{aligned} ST_p^F \cdot q_{d,1,p}^F &\leq (\frac{H}{|\mathcal{K}|} - CT^F) \cdot s_{d,1,p}^{c,F} \\ &\quad + \frac{H}{|\mathcal{K}|} \cdot s_{d,1,p}^F \quad \forall d \in \mathcal{D}, p \in \mathcal{P} \\ &\quad - \sigma_{d,1,p}, \end{aligned} \quad (4.47)$$

$$\begin{aligned} ST_p^F \cdot q_{d,k,p}^F &\leq (\frac{H}{|\mathcal{K}|} - CT^F) \cdot s_{d,k,p}^{c,F} \\ &\quad + \frac{H}{|\mathcal{K}|} \cdot s_{d,k,p}^F \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \setminus \{1\}, p \in \mathcal{P} \\ &\quad + \sigma_{d,k-1,p} \\ &\quad - \sigma_{d,k,p}, \end{aligned} \quad (4.48)$$

Ez a megközelítés a műszakrészre osztott feldolgozásnak egy rugalmasabb model-

lezését eredményezi, ezáltal növelve a keretfűrészgép kihasználtságát és így a termelés hatékonyságát is. A megoldás alkalmazásának hatásait a későbbiekben a tesztelésnél is bemutatom, emiatt külön jelölést vezetek be erre a modell variánsra, melyet "D[k]+", illetve "D[k]-EK+" elnevezésekkel jelölök.

5. fejezet

Tesztelés

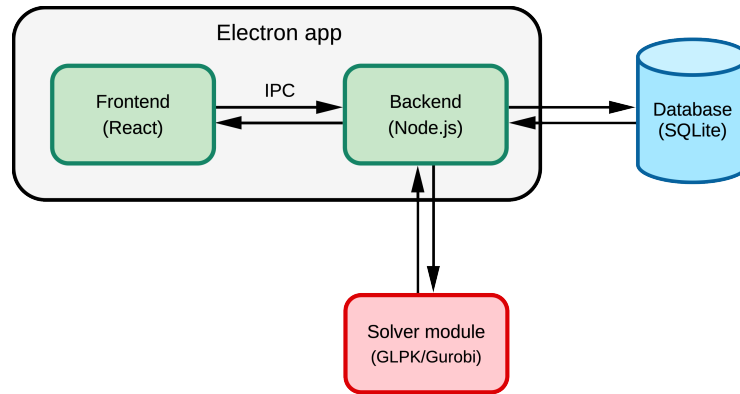
A 4. fejezetben javasolt megoldásaim több szemszögből is teszteltem. Először az 5.1 alfejezetben mutatok be egy grafikus interfészt, melyet a tesztadatok felhasználóbarát megadására és a megoldás kényelmes futtatásának céljából implementáltam. Továbbá a saját teszteseteim meghatározását és az azokra kapott eredményeket az 5.2 alfejezetben ismertetem.

5.1. Teszteset készítése grafikus interfészen keresztül és a tesztelési környezet

A fejezetben bemutatásra kerül egy grafikus interfész szoftveres felépítése, melyet az 5.1.1 szövegrészben ismertetem, míg felhasználói szemszögből való használatát az 5.1.2 alfejezetben. Ezenfelül ismertetem a tesztelés környezetét az 5.1.3 alfejezetben.

5.1.1. A szoftver architektúrája és felépítése

A szoftver implementálása során egy olyan felhasználóbarát interfész volt a célom, mely egy asztali számítógépen futtatható és egyszerű, hatékony kezelést biztosít. A megoldásnak a felhasználói felületen túlmenően funkciót kell biztosítania a felhasználó által bevitt adatok tárolására, illetve képes kell, hogy legyen futtatni a lineáris programozási modell megoldót. A szoftverrendszer architektúrája az 5.1 ábrán látható, mely a főbb komponenseket és azok kapcsolatát mutatja be.

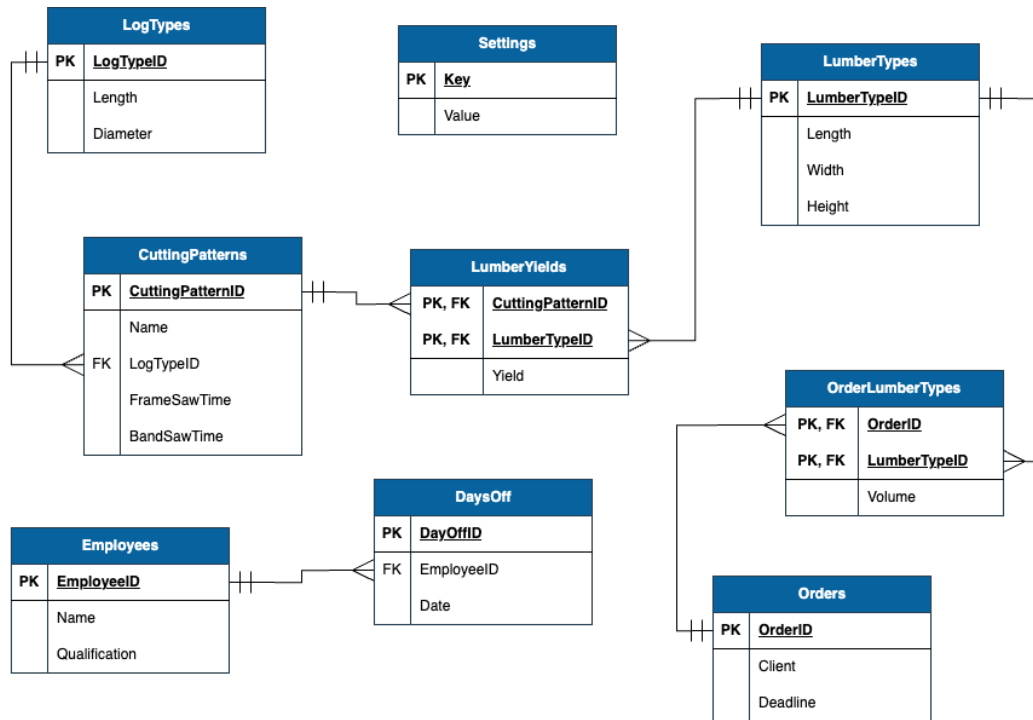


5.1. ábra. Rendszerarchitektúra diagram

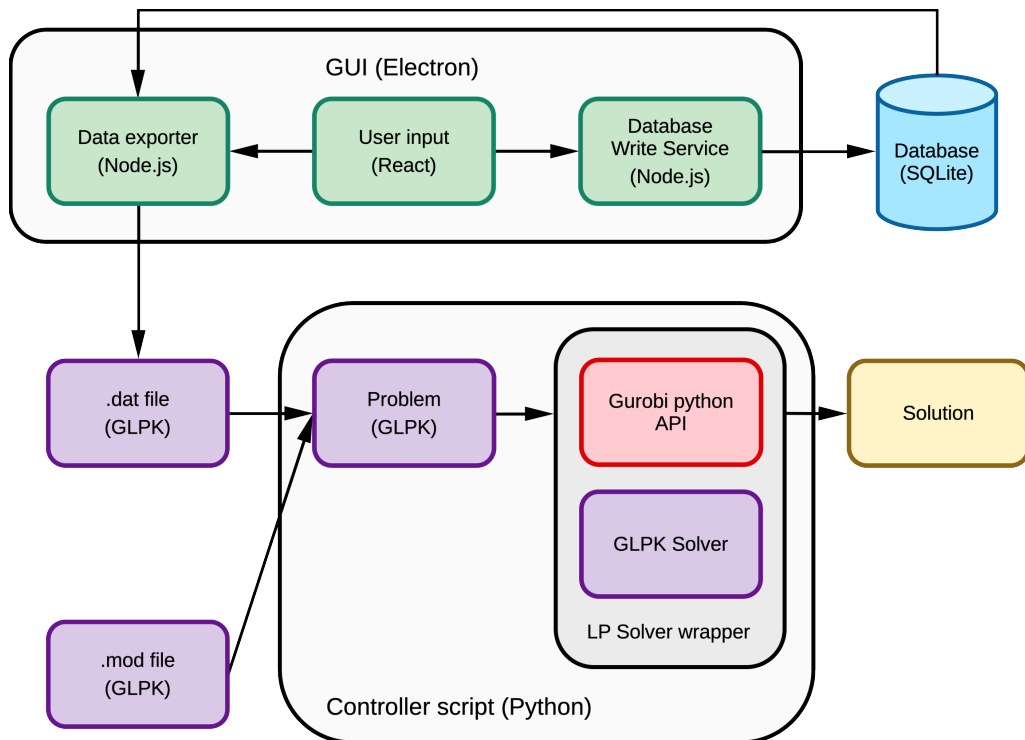
A grafikus interfész megvalósításához az Electron keretrendszert használtam, mely lehetővé teszi a modern webes technológiák alkalmazását asztali alkalmazások fejlesztésére. Az applikáció frontend részéhez React JavaScript-keretrendszert és Tailwind CSS nyílt forráskódú CSS-keretrendszert használtam, míg a backend folyamatok implementálásához Node.js-t választottam. Ezen két komponens közti kommunikációt az úgynevezett IPC (Inter Process Communication) csatorna teszi lehetővé. A MILP modell futtatását egy Node.js által kezelt Python szkript végzi, melyet az ábrán "Solver module" elnevezéssel jelöltem. A szkript képes mind a GLPK, mind a Gurobi optimalizáló szoftver használatára. Ezen túlmenően a backend felelős az adatkezelési folyamatokért is, beleértve az adatok tárolását és lekérdezését.

Az adattárolás egy SQLite adatbázisban történik, mely egy egyszerű és könnyen kezelhető megoldást nyújt az információk tárolására és kezelésére. Az adatok struktúráját és azok közötti kapcsolatokat az 5.2 ábrán látható egyed-kapcsolat diagram szemlélteti.

A diagramon szereplő struktúra néhol más logikát követ, mint a 3.2 fejezetben definiált paraméterek. A tervezés során fő szempont volt, hogy felhasználói szemszögből rögzítsük az adatokat nem pedig a modell szempontjából. Ez többek között kiterjed a humánerőforrás-menedzsmentre, ahol külön tároljuk az alkalmazottakat és a hozzájuk tartozó szabadságokat, vagy a keresletre, ahol külön megrendeléseket rögzítünk. Ezenfelül vannak olyan ütemezés specifikus információk, melyeket nem tárolunk el az adatbázisban, de modell futtatásához szükségesek, így a felületen keresztül ugyanúgy megadjuk azokat. Efféle funkcionalitások miatt egy komplexebb folyamatot kell kezelni a szoftver megoldásában, melyet az 5.3 ábra szemléltet.



5.2. ábra. Egyed-kapcsolat diagram



5.3. ábra. A szoftver működésének folyamata

A folyamat diagramon külön komponensként szerepel egy úgynevezett "exporter", mely adatbázisban tárolt adatokat és a felhasználói felületről származó információkat együttesen alakítja át a modell számára értelmezhető formátumra. Bizonyos adatokat az adatbázisból és bizonyos információkat felhasználói felületről közvetlenül konvertálja át a megfelelő adattípusra. A lineáris programozási modellt GMPL formátumban kezeljük, melyet az ábrán a "Problem" komponens reprezentál. A megoldás futtatását az "LP solver wrapper" doboz szemlélteti, mely magába foglalja a GLPK nyílt forráskódú megoldójának a lehetőségét, illetve döntést enged az Gurobi Python API használatára is. Ezt követően a megoldó meghatároz egy megoldást, melyből a szoftver egy Gantt diagramot készít a Python Matplotlib könyvtár alkalmazásával.

5.1.2. Szoftver használata

A szoftver lehetőséget biztosít, hogy az üzem ütemezéshez szükséges adatokat különböző beviteli mezők segítségével megadhassuk és elmenthessük. Ennek eredményeképpen az alkalmazás újra indítását követően se vesznek el az előzőleg megadott adatok. A felület megjelenése az 5.4 és az 5.5 képernyőképen látható.

The screenshot shows the 'Sawmill Scheduler' application window. On the left is a sidebar with navigation links: 'Sawmill Settings', 'Employees', 'Orders', and 'Scheduling'. The main area is divided into sections for data entry:

- Logs:** A table with columns 'Length (m)', 'Diameter (m)', and 'Volume (m³)'. It contains one row with values 4.00, 0.50, and 0.79. Above the table are input fields for '5' and '0.5', and an 'Add' button.
- Lumbers:** A table with columns 'Length (m)', 'Width (m)', 'Height (m)', and 'Volume (m³)'. It contains two rows with values (4.00, 0.30, 0.20, 0.24) and (4.00, 0.40, 0.20, 0.32). Above the table are input fields for 'Length (m)', 'Width (m)', and 'Height (m)', and an 'Add' button.
- Cutting Patterns:** A section with a 'New Cutting Pattern' button and a table with columns 'Name' and 'Log'. It contains one row with 'CP' and 'Length 4m - Diameter 0.5m'.

5.4. ábra. Az adatrögzítés egy példája a grafikus felületen

A felület beviteli mezőit és a hozzá tartozó funkcióit külön csoportokba osztottam, melyek között a bal oldalon lévő menürendszer segítségével navigálhatunk, ezzel befolyásolva az alkalmazás jobb oldalán található tartalmat. Ezen csoport első pontja a "Sawmill settings", melyben a fűrészüzemhez tartozó alapvető információkat rögzíthetjük és szerkeszthetjük, ennek egy példáját az 5.4 képernyőkép szemlélteti. Ezen az oldalon többek között lehetőség van a rönkökre meghatározott vágási minták kezelésére is. Ilyen funkció megvalósítása látható az 5.5 képen, ahol a felhasználó egy új mintát vehet fel.

The screenshot shows a web application titled "Sawmill Scheduler". On the left is a sidebar menu with options: "Sawmill Settings", "Employees", "Orders", and "Scheduling". The main content area is titled "New Cutting Pattern" and contains the following form elements:

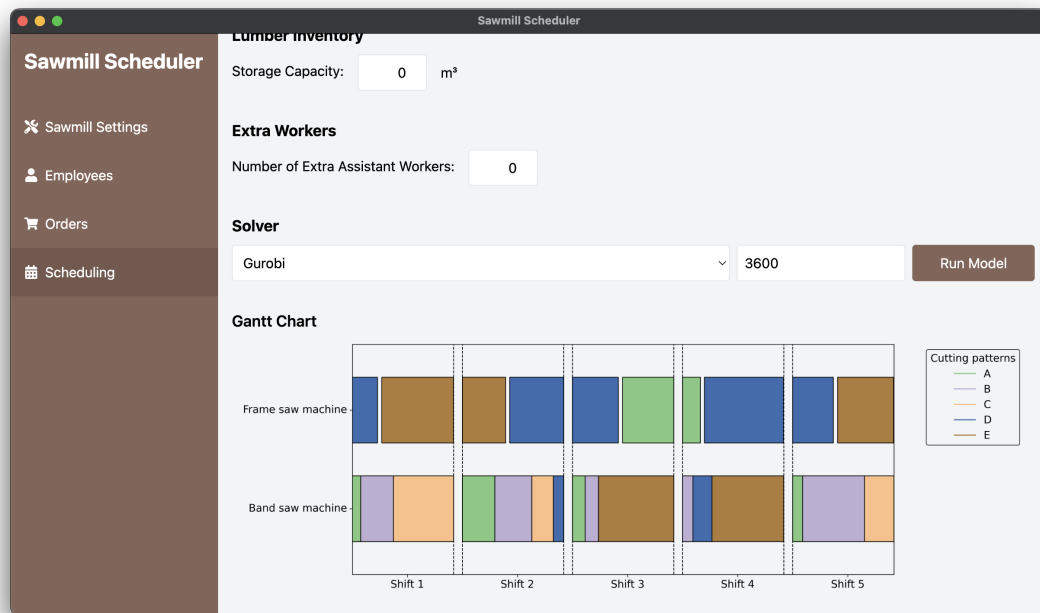
- Name:** A text input field with the placeholder "Name".
- Log:** A dropdown menu currently showing "Length 4m - Diameter 0.5m".
- Frame Saw Machine Cutting Time (mm:ss):** Two input fields for minutes and seconds, both currently set to "0".
- Band Saw Machine Cutting Time (mm:ss):** Two input fields for minutes and seconds, both currently set to "0".
- Lumber Yields:** A section with four rows, each for a specific dimension:
 - 4m x 0.2m x 0.2m: Input field "0" followed by a "%" sign.
 - 4m x 0.3m x 0.2m: Input field "0" followed by a "%" sign.
 - 4m x 0.4m x 0.4m: Input field "0" followed by a "%" sign.
 - 4m x 0.4m x 0.2m: Input field "0" followed by a "%" sign.
- Save:** A button at the bottom left of the form.

5.5. ábra. Új vágási minta rögzítése a grafikus felületen

Ezenfelül elkülönítettem az alkalmazottakkal kapcsolatos funkciókat az "Employees" menüpont alatt és a kereslethez vagy rendelésekhez köthető beviteli mezőket az "Orders" fül alatt. Az utolsó menüponton keresztül pedig az ütemezés specifikus paramétereket határozhatjuk meg. Többek között kiválasztható az LP megoldó szoftver, valamint megadható egy futási idő korlát is. Az adatok rögzítését követően futtatható a megoldó.

A futtatás eredményeképp születik egy Gantt diagram, melynek egy példája az 5.6 képernyőképen található. A diagramról leolvasható, mind a keretfűrész mind a szalagfűrész működésének ütemezése. Mindkét rönkvágó géphez tartozik egy-egy sáv, mely azt szemlélteti, hogy mikor melyik vágási mintával üzemeljen az adott

eszköz. Ezek a sávok kisebb téglalapokra lettek bontva, melyek közötti fehér üres rész jelöli az üresjáratokat, illetve a vágási minták közti átállásokat. A mintákat különböző színekkel azonosíthatjuk be, ehhez egy segédlet a diagram mellett található. A műszakokat az ábra x tengelyén különítettük el, melyek közti üres rész jelöli a napoknak azon szakaszát mikor nem üzemel a fűrészüzem.



5.6. ábra. Példa megoldás a grafikus felületen

5.1.3. A tesztelés környezete

Egy lineáris programozási modell megoldására számos szoftvercsomag létezik. Ilyen optimalizálási eszköz például a GLPK nyílt forráskódú megoldója. Azonban az iparban igen elterjedt a CPLEX, illetve a Gurobi szoftvere, melyek rendkívül összetettek, de jóval eredményesebbek a komplexebb modelleknél, mint az előbb említett GLPK. A javasolt modellt minden esetben a Gurobi 10.0.1-es verziójával oldottam meg, a Gurobi Python API-t használva. A modelleket GMPL nyelven implementáltam. A 4.1 C-EK modell implementációja az A függelékben található, míg a 4.2 D[k]-[EK]+ megközelítés kódját a B függelékben írtam le. A megírt modellt lp típusú fájlra fordítottam, melyet a Gurobi megoldóval futtattam. Előfordulhat olyan eset, mikor ezen az időkorláton belül nem sikerül megtalálni a megoldónak az optimális megoldást, ekkor az addig talált legjobb megoldás kerül meghatározás-

ra, mint a leghatékonyabb megtalált ütemezés. Az eredményeket Python szkripttel dolgoztam fel, majd a Matplotlib külső könyvtár segítségével ábrázoltam. A teszteseteket egyetlen számítógépen futtattam, Apple M2 modellű processzorral és 16 GB-os memóriával.

5.2. Ipari alkalmazhatóság és hatékonyság tesztelése

Ebben a fejezetben megvizsgálom mind a megoldás gyakorlati alkalmazhatóságát, mind a modell hatékonyságát. Az előbbi szemponthoz a szakterület szakértőitől származó adatokon alapuló tesztesetet határoztam meg (5.2.1. alfejezet) és teszteltem részletesebben, míg utóbbihoz egy nagyszámú generált adathalmazt állítottam elő (5.2.2. alfejezet).

5.2.1. Tesztelés ipari tapasztaltok alapján

Mivel a munkám során vizsgált problémaosztályra tesztadatok sem a szakirodalomból, sem ipari partnertől nem álltak a rendelkezésemre, saját tesztadathalmazt kellett készítenem. Annak érdekében, hogy a tesztesetek minél közelebb álljanak a valósághoz, faiparral foglalkozó szakemberekkel konzultáltam. Ezenfelül Hargitai László Fűrészáru [21] című könyvét használtam fel a reális paraméterek deklarálásához.

A tesztadathalmaz meghatározásánál 3 rönk osztályt és 12 fűrészáru típust különböztettem meg. Ehhez 5 eltérő vágási mintát terveztem meg a Pitago Optimizers [36] szoftver segítségével. Az egyes mintákra vonatkozó vágási időket, valamint a keretfűrész átállási idejét szakértőkkel folytatott konzultáció alapján állítottam be. A tervezési időszakot 1 hétként definiáltam, azaz összesen 5 darab 8 órás műszakot tartalmaz a tervezés. Az üzemről azt feltételeztem, hogy mindkét gép működtetéséhez egy szakember, ezenfelül a szalagfűrészhez további 1, míg a keretfűrészhez 3 dolgozó szükséges. Különös figyelmet fordítottam a munkaerőhiány megvizsgálására, melyhez a az 5.1 táblázatban bemutatott munkavállalói rendelkezésre állási adatokat használtam.

Az 5.1 táblázatról leolvasható, hogy melyik műszakban mennyi szakember és segédmunkás állt rendelkezésre. A cél a munkaerőhiány hatásának vizsgálata volt, így egyes műszakokban a szakemberek száma lehetett 1 is, ami azt eredményezi, hogy

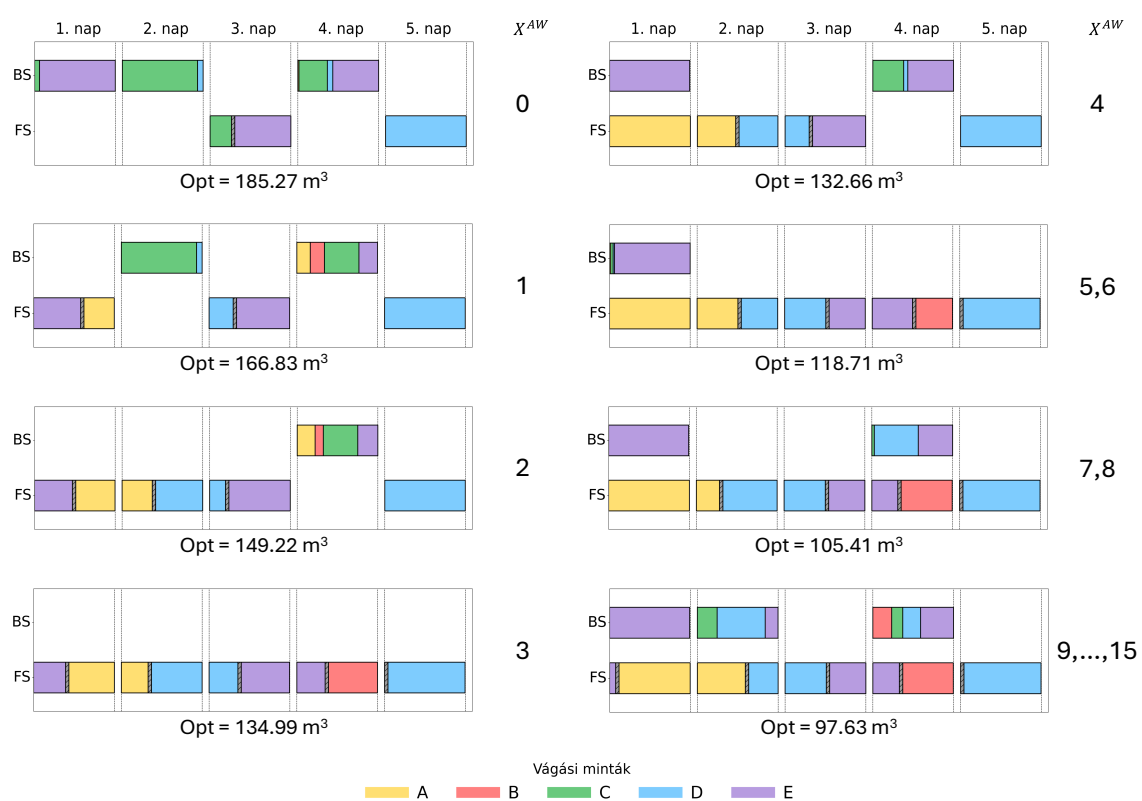
Műszak	Szakember	Segédmunkás
1	2	1
2	2	1
3	1	3
4	2	1
5	1	3

5.1. táblázat. Az üzem alkalmazottainak a rendelkezésre állása

az adott műszakban csak egy gép üzemelhetett. Ezenfelül a segédmunkások száma szintén eltérően határoztam meg az egyes műszakokra vonatkozóan, ezzel szintén befolyásolva a gépek üzemeltetését. A tesztelés során megvizsgáltam, hogy több extra segédmunkás behívása, milyen módon változtathatja meg az eredményt, így a fenti rendelkezésre állás a legszélsőségesebb munkaerőhiányt reprezentálja. A teljes tervezési horizontra nézve a fűrészáru kereslet 366 m^3 -re lett beállítva, melyet a napokra egyenletesen osztottam el 20 m^3 -es raktárkapacitás mellett. Az LF_t késedelmi költséget minden fűrészáru esetében egységesen 1 köbméterenként állítottam be, így a célfüggvény célja a nem teljesített mennyiség minimalizálása.

A tesztelés célja, hogy szimuláljam a munkaerőhiányt, melyet a maximálisan rendelkezésre álló extra segédmunkások számának variálásával valósítottam meg. A futtatások során a 3.2 fejezetben megfogalmazott X^{AW} paramétert növeltem 0-tól kezdődően. Így az első esetben nem engedélyeztem az üzem számára a plusz munkások behívását, és az előre meghatározott beosztás alapján kellett ütemezni az alkalmazottakat. Ezáltal azt vizsgáltam meg, hogy mennyivel jobb megoldás érhető el a munkaerőhiány csökkentésével. A futtatást először a C-EK folytonos váltási időpont alapú megközelítéssel végeztem el. Első körben ennek eredményeit Gantt diagramokon vizualizáltam az 5.7 ábrán.

A Gantt diagramok segítségével jól látható, hogy a munkaerőhiány csökkentésével hogyan változik az ütemezés és a nem teljesített kereslet mennyisége. Az ábrán 2 oszlopba rendeztem a megoldásokat: bal oldalon láthatók a 0-tól 3-ig növelt X^{AW} értékhez tartozó eredmények, míg jobb oldalon az 4-től 15-ig terjedők, ahol az utolsó 6 esetben megegyező eredményt kaptam, ezért ezeket csak egyszer ábrázoltam. Minden gantt diagramon BS-el jelöltem a szalagfűrész, míg a FS-el a keretfűrész. Az első eset mutatja meg, hogy milyen megoldást talált a modell, mikor nincsen



5.7. ábra. C-EK modellel tervezett ütemezések gantt diagramjai az extra munkaerő növelésnek változtatásával

lehetőség plusz munkások behívására. Leolvasható, hogy ekkor a szalagfűrész 3 műszakban, míg a keretfűrész 2 műszakban üzemel, illetve csak a C, D, E vágási mintákat alkalmazza a két gép. A javasolt modell mintegy 1,04 másodperc alatt egy olyan ütemezést talált, mely a kereslet 49,37%-át képes kielégíteni. Mikor az X^{AW} paraméter értékét 1-re növeltem, majd így ismételtén futtattam a modellt, olyan eredményt kaptam, ahol az látható, hogy az első napon már a keretfűrész is üzemeltethető volt a szalagfűrész helyett. Ezenfelül minden vágási mintát is képes volt alkalmazni a két gép. Ezekkel a változásokkal növekedett a termelés és csökkent a nem teljesített kereslet mintegy 5,04%-kal. A második extra munkás behívása hasonló mértékben csökkentette az optimumot, 4,81%-kal. Ezáltal az üzem képes volt a keretfűrész működtetésére a második műszakban is a szalagfűrész helyett. A harmadik alkalmazott behívása esetén az üzem csak a keretfűrész üzemeltette, így 36,88 százalékra csökkentve a hiányt. A negyedik extra munkásnál viszont a 4. napon csak a szalagfűrész üzemelt, melynek kevesebb humán erőforrás igénye van, ezáltal az üzem képes lehetett áthelyezni munkaerőt egy másik napra és így az első napon mindkét gépet tudta működtetni, mely az optimumot 0,64%-kal csökkentette. Öt extra alkalmazottnál, már viszont képes volt úgy mindkét gépet használni az első napon, hogy mellette a több nap mindegyikén is képes volt a keretfűrészszel termelni, így a nem teljesített kereslet már csak 32,43%-ra csökkent. Ezt követően csak a hetedik alkalmazott behívásával sikerült javítani a megoldást, ahol már két napon is párhuzamosan üzemelt a két gép. Ekkor az optimum érték 28,8%-ra csökkent. További javulást a nyolcadik alkalmazott nem hozott, azonban a kilencedik segédmunkás behívása már csak 26,67%-ra csökkentette a hiányt. Itt már csak két olyan nap van a tervezésben, mikor a szalagfűrész nem üzemel. Ezeken a napokon csak 1 szakember állt rendelkezésre (ez az 5.1 bemutatott táblázatban látható), és mivel mindkét technológia működtetése megköveteli, hogy legalább 1 szakember felügyelje, így csak egyetlen gép üzemelhet a segédmunkások számától függetlenül. A termelésben létrejövő hiány így egészen idáig a kilencedik extra segédmunkás behívásáig csökkent. Ez egy értékes kimutatás lehet a gyártásvezető számára, hogy érdemes-e további munkásokat behívni vagy sem.

További futtatásokat elvégeztem a többi D2, D2-EK, D4, D4-EK, D8 és D8-EK diszkrét időrács alapú modellekre is. Az eredményeket az 5.2. táblázatban foglaltam össze, ahol minden megközelítésre vonatkozóan egyik oszlopban az m^3 -ben értendő Opt értékek mutatják meg a termelésben keletkezett hiányt, ezenfelül egy másik

oszlopban másodpercben mért Időértékek a modellek futási idejét jelölik.

X^{AW}	C-EK		D2		D2-EK		D4		D4-EK		D8		D8-EK	
	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő
0	185.27	1.04	186.40	0.08	186.40	0.08	186.08	0.11	186.08	0.10	186.88	0.15	186.88	0.17
1	166.83	2.16	168.05	0.09	168.05	0.17	168.15	0.21	168.15	0.40	168.87	1.02	168.87	0.52
2	149.22	0.80	151.38	0.14	151.38	0.18	151.62	0.48	151.62	0.49	151.76	1.56	151.76	1.46
3	134.99	0.52	138.96	0.21	138.96	0.29	138.52	0.81	139.40	0.99	138.60	13.94	138.60	6.46
4	132.66	0.82	134.27	0.19	134.27	0.27	134.28	0.63	134.28	0.74	134.76	3.25	134.76	3.95
5	118.71	0.55	119.67	0.12	119.67	0.14	120.04	0.38	120.24	0.34	121.15	2.97	121.15	3.58
6	118.71	0.74	119.67	0.19	119.67	0.26	120.04	0.89	120.24	1.10	121.15	16.83	121.15	12.96
7	105.41	0.59	108.19	0.25	108.29	0.24	106.11	0.33	106.40	0.43	105.09	0.49	106.25	0.81
8	105.41	0.68	108.19	0.28	108.29	0.43	106.11	0.32	106.40	0.50	105.09	0.87	106.25	1.38
9	97.63	1.33	101.69	0.28	101.69	0.22	99.28	0.43	99.38	0.35	98.72	0.73	99.51	0.78
...														
15	97.63	1.58	101.69	0.29	101.69	0.20	99.28	0.28	99.38	0.45	98.72	0.80	99.51	1.37

5.2. táblázat. Optimum (m^3) és futási idő (sec) eredmények C-EK, D[2, 4, 8], D[2, 4, 8]-EK modellek esetén az extra munkaerő növelésnek változtatásával

A táblázat alapján látható, hogy a D2 és D2-EK modellek esetén az optimum értékek nagyobbak, mint a C-EK eredményei. A futási időn viszont javulás látható, mely alapján úgy tűnik ezzel a megoldással érhető el leggyorsabban eredmény. Egyúttal az is megfigyelhető, hogy az egyetlen vágási minta csere megszorítás alkalmazásával itt csak 7-re és 8-ra paraméterezett X^{AW} értékek esetén rontott az eredményen. A D4 és D4-EK modellek esetén az optimum értékeknél az látható, hogy néhány esetben rosszabb eredményt értek el, mint a D2 és D2-EK modellek. Ugyanez megfigyelhető a D8 és D8-EK modellek esetén is, ahol az optimum értékek még rosszabbak, mint a C-EK modelleknél, ahol egészen 6-ig növelve az X^{AW} paramétert az optimum értékek nagyobbak, mint az előző D[2, 4], D[2, 4]-EK megközelítésekkel elértek. A futási idők tekintetében, a műszakrészek számának növelése mutat enyhe lassulást, de leginkább csak 8 részre osztott műszak esetében figyelhető meg 2 számjegyű futási idő.

A műszakrészek számának növelésével az optimum értékek csökkenése is várható, mivel ezáltal növekedik a keresési tér, így növelve az olyan kombináció előfordulásnak az esélyét, mely hatékonyabb tervezést eredményez. Ezért megvizsgáltam azt a megközelítést, mellyel az 5.2 eredményeit kaptam. Mint kiderült a 4.2.2. alfejezetben taglalt átcsúszási idő hiánya okozta a fentebb leírt váratlan eredményeket. Emiatt kibővítettem a modelleket ennek figyelembe vételével, majd újra futtattam a tesztet. Ezt követően az 5.3. táblázatban szereplő eredményeket kaptam, mely

az előző táblázattal megegyező formátumban tartalmazza az Opt (m^3) és az Idő (sec) értékeket.

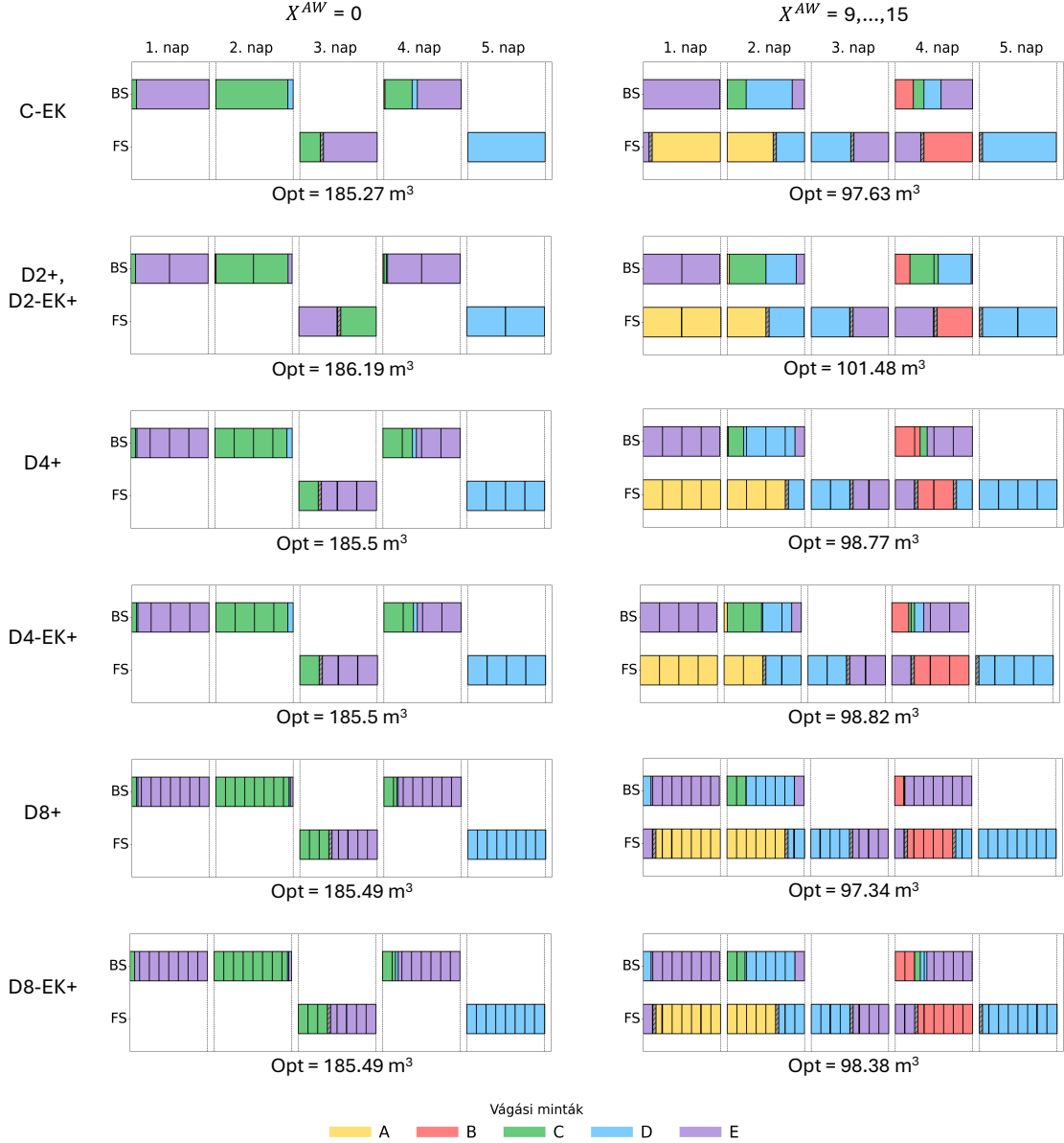
X^{AW}	C-EK+		D2+		D2-EK+		D4+		D4-EK+		D8+		D8-EK+	
	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő	Opt	Idő
0	185.27	1.04	186.19	0.08	186.19	0.09	185.50	0.18	185.50	0.19	185.49	0.75	185.49	0.74
1	166.83	2.16	167.85	0.16	167.85	0.18	167.69	0.62	167.69	0.47	167.01	2.56	167.01	3.28
2	149.22	0.80	151.17	0.18	151.17	0.24	151.13	0.72	151.13	0.55	149.95	9.85	149.95	12.86
3	134.99	0.52	138.75	0.18	138.75	0.22	137.72	1.52	138.75	1.48	135.29	33.64	135.29	37.03
4	132.66	0.82	134.06	0.24	134.06	0.31	133.67	1.06	133.67	1.27	133.35	19.55	133.35	23.38
5	118.71	0.55	119.46	0.17	119.46	0.17	119.33	1.00	119.46	0.93	119.23	19.00	119.23	17.58
6	118.71	0.74	119.46	0.19	119.46	0.53	119.33	3.92	119.46	1.48	119.23	49.92	119.23	42.08
7	105.41	0.59	108.09	0.17	108.10	0.16	105.44	0.69	105.75	0.64	103.80	1.88	104.90	4.10
8	105.41	0.68	108.09	0.29	108.10	0.22	105.44	0.58	105.75	0.71	103.80	3.10	104.90	9.87
9	97.63	1.33	101.48	0.29	101.48	0.20	98.77	0.64	98.82	0.51	97.34	46.45	98.38	93.06
...														
15	97.63	1.58	101.48	0.32	101.48	0.30	98.77	0.70	98.82	0.41	97.34	31.47	98.38	63.52

5.3. táblázat. Optimum (m^3) és futási idő (sec) eredmények C-EK, D[2, 4, 8]+, D[2, 4, 8]-EK+ modellek esetén az extra munkaerő növelésnek változtatásával

Az újra futtatott eredmények az optimumok tekintetében jóval monotonabbak. Az új táblázat alapján elmondható, hogy az átcsúszási idő kezelése egyértelmű javulást hozott a tervezések hatékonyságában az 5.2. táblázatban bemutatott modell variánsokhoz képest. Ezenfelül a D[8]+, D[8]-EK+ megoldó módszerek futásai viszont kisebb időbeli romlást is eredményeztek a D[8] és a D[8]-EK modellekhez képest. Ez a többi modell variáns esetében nem számottevő. A diszkrét időrács alapú modellek esetében az értékek jól mutatják, hogy a tervezés hatékonysága növekedik a műszakrészek számának növelésével. A C-EK megközelítéshez képest ebben a tesztelésben a többi modell a legtöbb esetben rosszabb optimumokat ért el. De sejthető, hogy nagyobb példák esetén nagyobb számítási igényrel rendelkezhet a folytonos váltási időpont alapú megoldás. Hiszen ezen kisebb példák esetén a futási idők tekintetében a D2+ többszöröse található meg a C-EK modellek oszlopában mindamellett, hogy a termelésben keletkező hiány szempontjából a különbségek minimálisak. Az 5.2. táblázathoz hasonlóan számítási igény növekedést továbbá az előző a műszakrészek számának növelése is okozott, mely megfigyelhető a D2-[EK]+ és a D4-[EK]+ között, de jelentős különbség a 8 műszakrésszel számoló megközelítések esetében tapasztalható. További érdekesség, hogy a D[k]+ modellek több esetben hatékonyabb megoldásokat adnak, mint a D[k]-EK+ megközelítések amellet, hogy az utóbbiakkal kapott futási idők alapján nem mondható el, hogy gyorsulást ért

volna el ez a plusz megszorítás.

A különböző modelleket szerettem volna gantt diagramokon is összehasonlítani, így kiemeltem a 0 extra bérelt segédmunkás esetét, illetve a 9 és annál több plusz alkalmazottal történő tervezéseket. Ennek az eredménye látható az 5.8. ábrán.



5.8. ábra. Gantt diagramok a különböző MILP variánsok megoldásairól

A gantt diagramok hasonlóan kerültek ábrázolásra mint az előzőekben. Bal oldali oszlopban található az $X^{AW} = 0$ paraméterrel kapott diagramok, míg a jobb oldalon az $X^{AW} = 9, \dots, 15$ beállítással elért tervek. Minden sor egy modell variánszt reprezentál. Kivételt képez ez alól a "D2+, D2-EK+"-al jelölt 2. sor, ahol a két modell eredményei megegyeztek, így egyetlen sorként ábrázoltam őket.

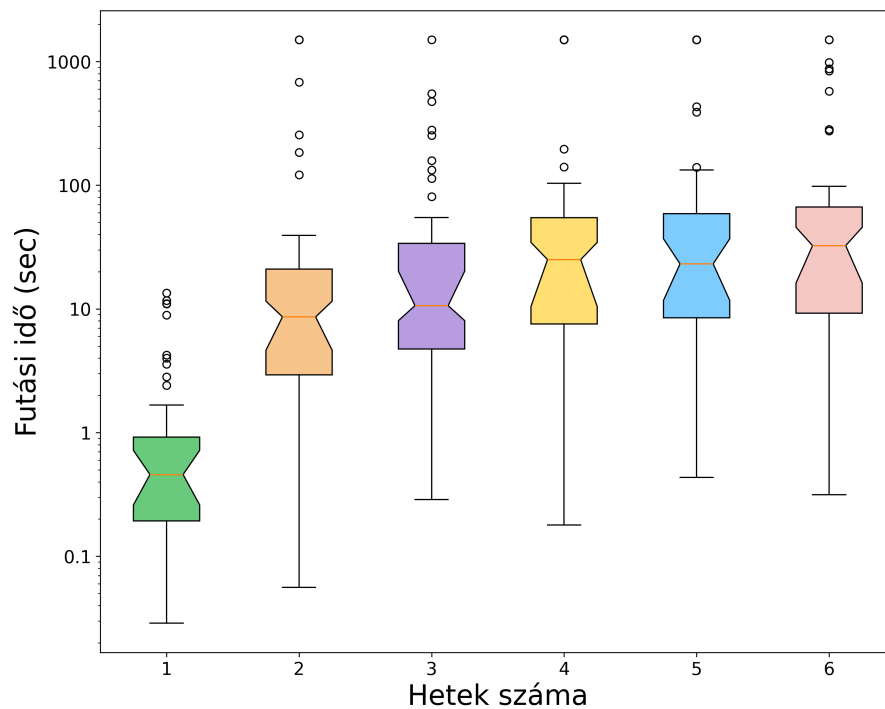
Jól látható, hogy a diszkrét időrács alapú modellek hasonló tervezéseket próbálnak elérni, mint a C-EK által tervezett ütemezés. A D2+, D2-EK+ modell eredményei esetén jól látható, hogy sokkal korlátozottabban dönt a megoldás a keretfűrészben történő vágási minta csere időpontokról, mint a többi megközelítés. A 0 extra alkalmazottra paraméterezett ütemezés 3. napján látható jól, hogy míg a C-EK inkább a műszak 1/4-nél vált C vágási mintáról E-re, addig a 2 egyenlő műszakrészre osztott megoldó módszerek csak a műszak felére rakhatják a csere időpontját. Ez a 4 műszakrészre osztás esetén előnyre fordul, hiszen itt már megoldható, hogy a gép 1 műszakrészt dolgozzon 1 mintával és 3-at egy másikkal. Így az $X^{AW} = 0$ paraméterrel futtatott D4+ és D4-EK+ modellek eredményei már jóval közelebb vannak a $185,5 \text{ m}^3$ optimummal a C-EK $185,27 \text{ m}^3$ értékéhez. A 9 és annál több plusz bérelt munkás esetén megfigyelhető a D4+ és a D4-EK+ megoldók közötti különbség, ugyanis az előbbi tervezése szerint a 4. munkanapon kétszer is vágási mintát cserél a keretfűrész, ezzel $0,05 \text{ m}^3$ -el jobb eredményt elérve. Ugyanez a jelenség látható a D8+ és a D8-EK+ között, ahol 8 műszakrész miatt jelentősen több tervezési lehetőség áll rendelkezésre. A megszorítás alkalmazása, miszerint egy műszakban egyszer történhet átállítás a vágási minták között, $1,04 \text{ m}^3$ -el kevesebb fűrészárut eredményezett, mint a D8+ a korlátozás nélkül. A 0 plusz bérelt alkalmazott esetén már egyáltalán nem okozott számottevő különbséget a 4-nél többre felosztott műszakrész szám. Ezenfelül nem volt olyan ütemezés, ahol többször váltott volna mintát egy műszakrészben így az előbb említett megkötés nem okozott változást a tervezésben.

5.2.2. Nagyszámú generált adathalmaz

A javasolt modellek hatékonyságának vizsgálatához egy Python szkriptet implementáltam, mellyel nagyszámú véletlenszerűen generált adathalmazt állítottam elő. Az adathalmazt 6 különböző csoportra osztottam. Ezek a csoportok különböző tervezési időtávokat fednek le, pontosabban mind az 1, 2, 3, 4, 5 és 6 hetes időszakra vonatkozóan 50-50 példát tartalmaznak. Így összesen 300 példát generáltam, ahol a fűrészüzem technikai paraméterei az előzőekben (5.2.1. alfejezet) meghatározottakkal megegyezően lettek beállítva. Ezenfelül nincs megengedve az extra segédmunkások behívása. Minden tervezési időszakra vonatkozó példacsoport esetén a kezdeti készlet méret és a kumulált kereslet mennyisége rögzített volt. Mindkét érték az időtáv hosszához mérten arányosan nőtt. Az 1 hetes periódus esetén minden rönk

osztályra vonatkozóan 250 darab rönköt tartalmazó kezdeti készlet és a fűrészáruk-ra összesen 400 m^3 kereslet lett beállítva. Az egyes csoport példái között csupán a kereslet eloszlása és a dolgozók rendelkezésre állása változott.

Először a C-EK módszerrel az összes példára futtattam a Gurobi megoldó szoftverét, ahol az időkorlátot 1500 másodpercre paramétereztem fel, melyet csak néhány kiugró esetben ért el a modell. Az eredményeket az 5.9. doboz-ábrán jelenítettem meg, ahol az y tengelyen lévő időértékeknél exponenciális skálázást használtam az értékek jobb értelmezhetősége érdekében, míg az x tengelyen a tervezési időtávokat jelöltem.

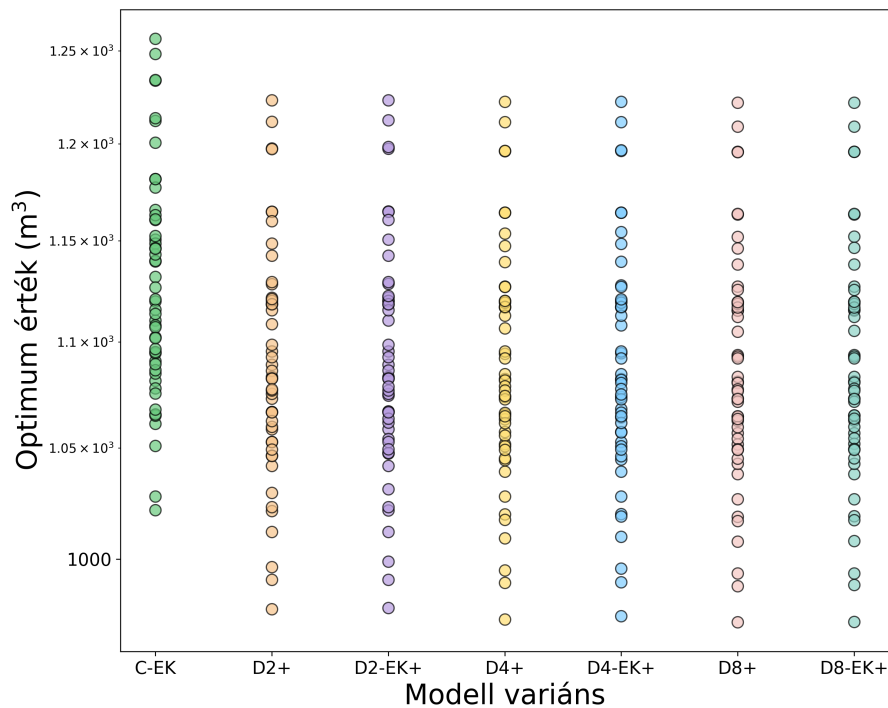


5.9. ábra. C-EK modell futási idejei (sec) 1-6 hetes tervezési időszakokra nézve nagyszámú véletlenszerűen generált példák esetén dobozdiagramon ábrázolva

Az eredmények meglepőek voltak, mivel a modellben szereplő bináris változók számának növekedésével az volt várható, hogy a szükséges számítási idő exponenciálisan növekedni fog. Ez a viselkedés megfigyelhető az 1 és 2 hetes tervezési időszakok közötti példacsoportok esetén. Látható, hogy bár a tervezési időszak növelése továbbra is növelte a számítási időt, azonban ez a változás kevésbé drasztikus, és az 5. héten ez a tendencia kiegyenesedni látszik. A hosszú távú tervezésre szánt nagyobb példák valószínűleg továbbra is megoldhatatlanok ésszerű időn belül. Elmondható, hogy a rövid és középtávú tervezési időszakok gyakorlati jelentősége sokkal nagyobb,

így a tesztek leginkább a másfél hónapos tervezési időszakokra fókuszáltak, ahol a javasolt modell alkalmazhatónak bizonyult. Az eredmény azt is mutatja, hogy van lehetőség további gyakorlati részletek beépítésére, a futási idők ésszerű szinten tartása mellett.

Ezt követően kiválasztottam a 4 hét hosszú tervezési időszakkal rendelkező példákat, majd megvizsgáltam azokat a D2+, D2-EK+, D4+, D4-EK+, D8+ és D8-EK+ modellekkel is. A vizsgálatok során a C-EK modell esetén az átlagos optimum érték 1128.98 m^3 , míg a többi modellre az átlagok a következőképpen alakultak: D2+: 1090.93 m^3 , D2-EK+: 1091.42 m^3 , D4+: 1088.69 m^3 , D4-EK+: 1089.13 m^3 , D8+: 1087.67 m^3 , D8-EK+: 1087.80 m^3 . A futtatások során elért optimális értékeket az 5.10. szórásdiagramon ábrázoltam, ahol x tengely jelöli a termelésben elért hiány mértékét m^3 -ben, míg az y tengelyen a modell variánsokat, melyekhez tartozó futtatások optimum értékeit reprezentálják a pontok.



5.10. ábra. C-EK, D[2, 4, 8]+ és D[2, 4, 8]-EK+ optimum (m^3) értékei 4 hetes tervezési időszakra nézve nagyszámú véletlenszerűen generált példák esetén szórásdiagramon ábrázolva

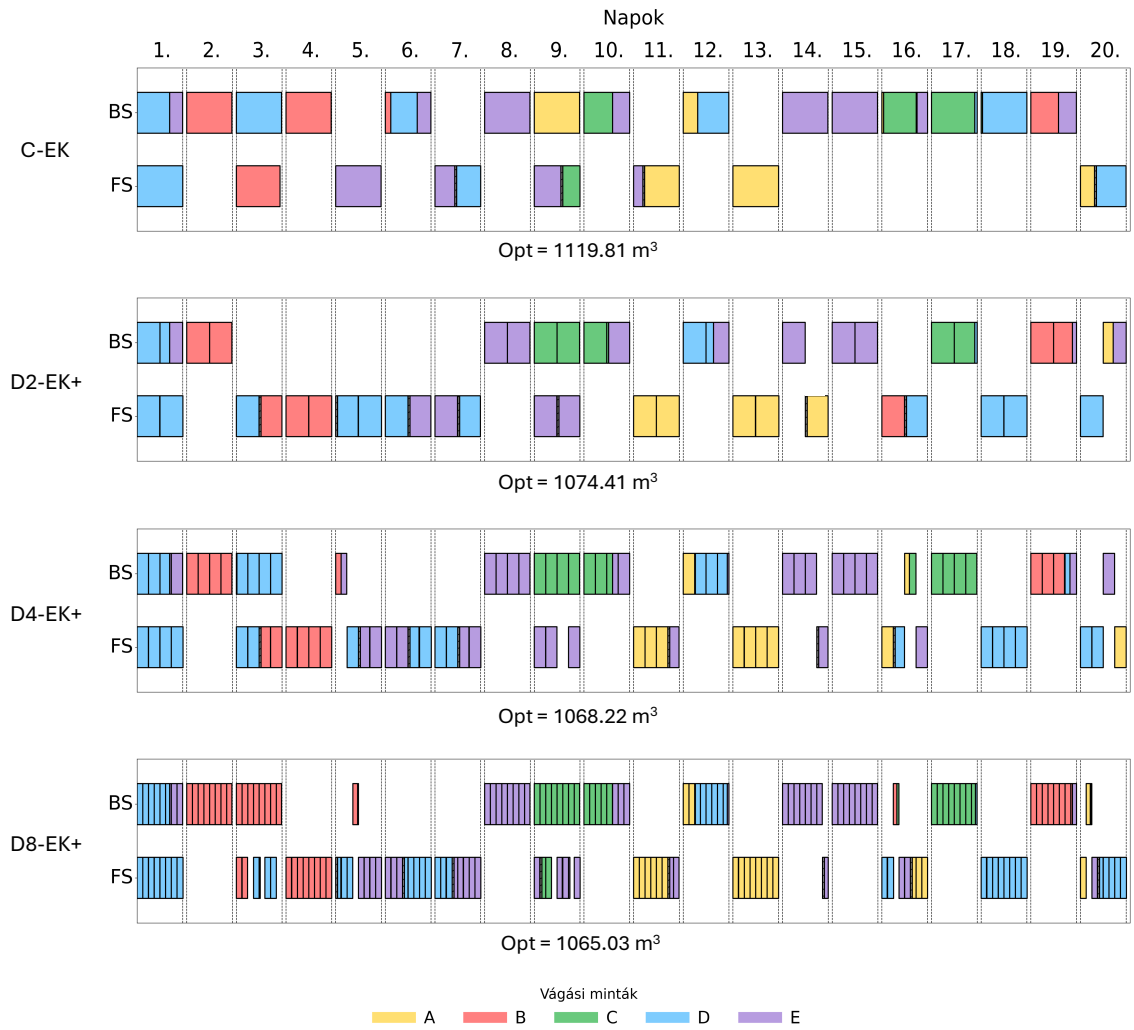
A diagramon látható, hogy a C-EK módszer optimális értékei gyengébbek, mint a többi műszakrészekre felosztott logikát követő modell variánsok értékei. Továbbá észrevehető az előzőekben, már az 5.2.1. alfejezetben bemutatott teszteseteknél azo-

nosított tendencia, hogy az egyenlő egységekre bontott megközelítésnél, amennyiben növeljük a műszakrészek számát javul a tervezés hatékonyságba. Jelen esetben ennek a javulásnak a mértéke kevésbé számottevő a C-EK és a D2+ vagy a D2-EK+ optimum értékei között megjelent különbség. Hasonló mód, mint kisebb példánál itt is sok esetben minimális mértékben nőtt a termelésben keletkezett hiány a D[k]-EK+ modelleknél a D[k]+ variánsokhoz képest.

A kisebb példák esetén ritkán fordult elő olyan, hogy a diszkrét időrács alapú módszer olyan ütemezést eredményezett volna, ahol egy műszakon belül, a különböző műszakrészekben eltérő módon üzemelnek a gépek. Ez a tényező magába rejti, hogy jobb tervezés is elérhető, mint a C-EK modellel, ahol nincs erre lehetőség. Vélhetően a véletlenszerűen generált példák esetén úgy alakult a paraméterezés, hogy ennek nagyobb létjogosultsága volt, így többször jobb eredményeket értek el a D[k]+ és a D[k]-EK+ megoldó módszerek. A jelenség megvizsgálása érdekében kiválasztottam egy példát, majd vizualizáltam a C-EK+, D2-EK+, D4-EK+ és a D8-EK+ modell variánsok által tervezett ütemezéseket az 5.11. ábrán gantt diagramokon keresztül.

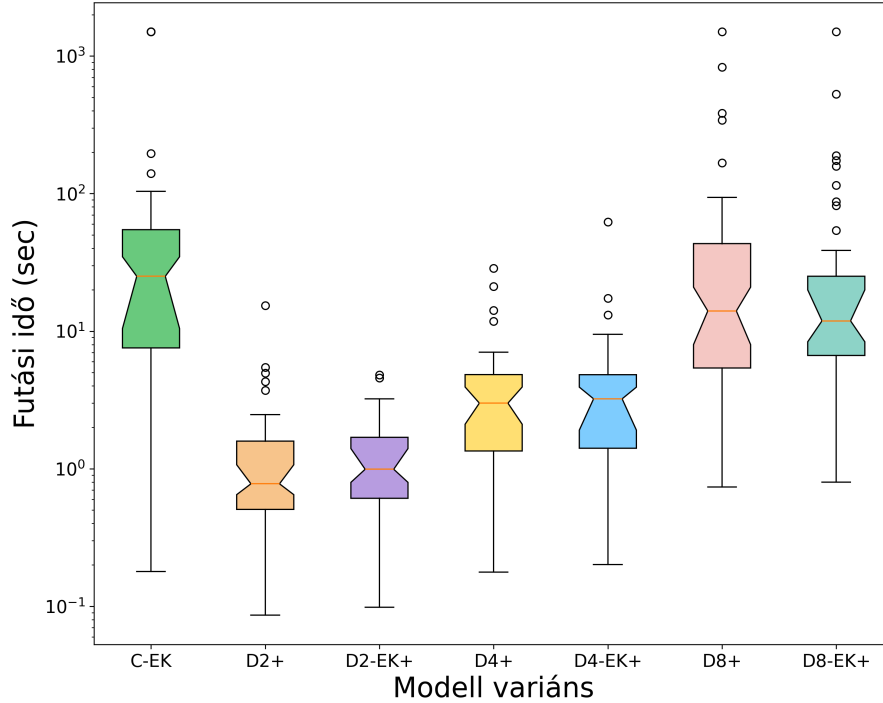
Az első sorban a C-EK modell ütemezése található, ahol az látszik, hogy a legtöbb műszakban a rönkök feldolgozása a szalagfűrészszel történik, illetve 3 munkanap van, mikor mindkét technológia párhuzamosan fűrészszel. Ezt követi a D2-EK+ módszer ütemezése, ahol $45,4 \text{ m}^3$ -el több fűrészáru került előállításra, mint az előző megoldásban. A diagramon látható, hogy a 14. és a 15. munkanapon is átszerveződés történt. Előbbinél például az üzem a szalagfűrészszel dolgozik a műszak feléig, majd áttér a második műszakrészben a keretfűrész rönkvágó alapgépre. Azáltal, hogy ezt képes megtenni az üzem, a napi kereslet igényeket is hatékonyabban tudja kielégíteni. Ugyanez a jelenség figyelhető meg a 4 műszakrésszel számoló D4-EK+ esetén is, ahol már az 5. és 16. napon is előfordul, hogy gépeket váltunk a műszakrészek között. Megfigyelhető, hogy ezen munkanapok mindegyikén az üzem a vágási minta csere átállításának az idejével spórol, hiszen az üres időkben a keretfűrészszel megtörténhet ez a művelet. Ez a 4 részre történő felosztás így tovább növelve a tervezés hatékonyságát $6,19 \text{ m}^3$ -el jobb eredményt ért el, mint a 2 műszakrészre osztott megközelítés. Hasonlóan alakul a D8-EK+ modell variáns által meghatározott ütemezés, ahol még részletesebbnek mondható a tervezés. Ezáltal további $3,19 \text{ m}^3$ -el sikerült csökkenteni a feldolgozás során keletkezett hiányt.

A célfüggvény értékeken felül megvizsgáltam a nagyszámú generált tesztesetnél, hogyan alakultak a futási idők. Az 5.12. doboz-ábrán keresztül mutatom meg a C-



5.11. ábra. C-EK és D[2, 4, 8]-EK+ modellekkel tervezett ütemezések gantt diagramjai egy 4 hetes tervezési időszakkal rendelkező példára

EK, D2+, D2-EK+, D4+, D4-EK+, D8+ és D8-EK+ modell variánsokkal történt futtatások időigényeit, ahol az 5.9. ábrához hasonlóan olvashatóak le az értékek, azzal a különbséggel, hogy az x tengelyen a modell variánsokat különítettem el.



5.12. ábra. C-EK és D[2, 4, 8]+, D[2, 4, 8]-EK+ modellek futási idejei (sec) 4 hetes tervezési időszakra nézve nagyszámú véletlenszerűen generált példák esetén dobozdiagramon ábrázolva

A diagramon látható, hogy a C-EK modellhez képest a diszkrét időrács alapú modellek gyorsabban találták meg az optimális megoldást, a D8+ és a D8-EK+ modellek kivételével, ahol kisebb szórással, de hasonló eredmények születtek. Egyértelműen látszik, hogy a műszakrészek számának növelésével a számításigény is növekszik, így 2 felé osztott munkanapokkal kalkulált módszerek hozták a leggyorsabb eredményeket. Érdekes továbbá, hogy a $D[k]_+$ és a $D[k]_{-EK+}$ modellek között nincs számottevő különbség a $k = 8$ kivételével, ahol néhány esetben gyorsítani tudtam azzal, hogy csökkentettem a megoldások számát az extra korlátozás behozásával.

6. fejezet

Összefoglalás, konklúzió és a jövőbeli tervek

6.1. Összefoglalás

Dolgozatomban megvizsgáltam a fűrészüzemek ütemezését és annak több sajátosságát. Az irodalomban lévő tanulmányokban több megoldás is létezik, melyek különféle kihívásokkal foglalkoznak. Munkám egyik eredménye egy olyan probléma azonosítása, mely figyelembe vesz néhány, kisebb fűrészüzemeket érintő gyakorlati aspektust. Ilyen tényezők a fűrészüzemekben alkalmazható rönkvágó technológiák közti különbségek, valamint az egyre gyakrabban problémát jelentő humánerőforrás fluktuációk kezelése. Ezeket a szempontokat a szakirodalomban lévő, nagyobb, automatizáltabb üzemekre kidolgozott megoldások általánosabban vagy egyáltalán nem kezelik. A feladatosztály megoldásához először kidolgozásra került egy MILP modell, melyről 2024 júliusában született egy publikáció [37]. A dolgozatom további eredménye egy, más munkaszervezési szempontokat figyelembe vevő diszkrét időrács alapú megközelítés is, melyekben a keretfűrészen történő átállások időzítése korlátozottabb, azonban nagyobb flexibilitást engednek meg gépek közötti váltásokra. Ezen modellnek több variánsát is megvizsgáltam, összehasonlítva azokat az előzőekben említett modellel.

Munkám eredménye egy külön tesztet előállításra alkalmas grafikus interfész, mely a teszteléshez szükséges adatok kényelmes szerkesztésén felül képes a modellek futtatására is. Egy ipari adatokon alapuló teszten keresztül megvizsgáltam a modellek gyakorlati alkalmazhatóságát, valamint hogy hogyan befolyásolja a módszerek

által adott megoldásokat a munkaerőhiány. A modellek hatékonyságát, valamint az azokban implementált gyakorlati szempontok optimumra gyakorolt hatását nagy számú generált teszteseten vizsgáltam.

6.2. Konklúzió

A tesztelések alapján megállapítható, hogy a javasolt modellek bármelyike kellően jól alkalmazható rövid és középtávú ütemezések esetén egyaránt. Azt, hogy a C-EK+ vagy a D[k]-EK+ valamelyikét érdemes-e alkalmazni, azt elsősorban az befolyásolja, hogy az adott fűrészüzemben a munkaszervezés mire van felkészülve. Ha kellően kiszámíthatóan működik, és képesek tartani olyan gyártási tervet, ami-
ben naponta több váltás is van, akkor D[k]-EK+ által nyújtott flexibilitás miatt jobb eredmények érhetőek el. A példák alapján azonban nem feltétlen jobbak vagy amennyiben jobbak is, akkor sem nagyságrendekkel. Amennyiben az üzemben gyakoriak például a csúszások, nem elég rugalmas a munkaszervezés a műszakok közötti váltásokhoz, akkor a C-EK+ megközelítés alkalmazása lehet célszerűbb. Ebben az esetben olyan tervek születnek, melyek azt határozzák meg keretfűrészsre vonatkozóan, hogy adott műszakban adott vágási mintával adott darabszámú rönk feldolgozásával kell kezdeni, majd amennyiben vágási minta csere történik, akkor az átállást követő részre ad egy ugyanilyen tervezetet. Amennyiben csúszás van, akkor az átállás időpontja csúszik és így legrosszabb esetben csak a műszakrészben feldolgozott rönkök száma csökken a tervhez képest. Ha viszont olyan tervet használ az üzem, ahol műszakokon belül mozgás van, akkor nagyobb bonyodalmakat is okozhat egy esetleges csúszás. Mivel performancia szempontjából nincsenek jelentős különbségek, így a fő választási szempont a fentiek lehetnek. Ezenfelül a plusz alkalmazási lehetőség lehet, hogy a cég megtudja vizsgálni, hogy amennyiben változtatna a fentebb említett szokásain, akkor azzal esetlegesen mennyit tudna nyerni, megéri-e az ilyesfajta fejlesztés, hogy bonyolultabb tervek alapján is tudjon termelni.

6.3. Jövőbeli tervek

A téma rengeteg lehetőséget rejthet még a jövőben, melyeket későbbi kutatásomban szeretném még részletesebben feltérképezni és körüljárni. Egyik ilyen a C-EK modell olyan variánsa, ahol nem 1 vágási minta csere engedélyezett, hanem 2, és ezek

ugyanúgy bármikor megtörténhetnek. Ezenfelül érdekes lenne megvizsgálni, hogy a folytonos változó alapú modellben is lehetőséget adunk a műszakon belüli átszervezésre a gépek között.

A kutatásom jövőbeli céljai közé tartozik konkrét ipari partnerrel való együttműködés, melynek keretében a modellek további tesztelése és finomítása történne. A munkámnak fontos része volt a gyakorlati alkalmazhatóság is, így érdemes lehet további iparban fontos tényezők figyelembevételével bővíteni a problémát. Ezek származhatnának ugyanúgy ipari partner oldaláról, de az eddigi tapasztalatok alapján van néhány, már most felvetődött kiegészítés is:

Melléktermékek Egy ilyen lehetséges bővítés lehet a fűrészelések során keletkező fűrészipari melléktermékek kezelése, mely mint második célfüggvény jelenhet meg a modellben.

Rönkvágási módok Az élesvágáson túlmenően sok más rönkvágási mód is létezik, mint például a prizmavágás, ahol egy rönk több fűrészelési lépést igényel, ezáltal az ütemezési feladatra is kihatással van.

Farönk tulajdonságok Fűrészelési sebességet, termékminőséget jelentősen befolyásolhatják a farönkök olyan fizikai tulajdonságai is, mint például a fafaj, szárazság, stb.

Kötelező karbantartás Gyártási gépek számára gyakran előírt a karbantartási munkák rendszeres elvégzése. Ezek figyelembevétele és megfelelő időzítése szintén fontos lehet egy ilyen ütemterv esetén.

Ezenfelül tovább bővíthető a feladatosztály több feldolgozási lépéssel, melyek alkalmazását egyaránt befolyásolhatják a farönkök különféle jellemzői. Az ilyesfajta továbbfejlesztések egy lényegesen komplexebb problémát eredményeznének, így a modellek futási ideje is jelentősen megnőhet. Ennek érdekében célszerű olyan megoldást is kidolgozni, mely hatékonyabb futási idővel rendelkezik a komplexebb problémák esetén is, ilyen lehet egy heurisztikus algoritmus. Érdekes kihívás lenne az online ütemezés lehetőségét is vizsgálni, ahol hetente újra kell tervezni a gyártást, reagálva a változó körülményekre, vagy a sztochasztikus működésből eredő eltérésekre, mint például a csúszások. Ezek a kiegészítések, új megközelítések további

teszteléseket, összehasonlításokat is magukkal vonnak, így ez is fontos része lenne a jövőbeli munkámnak.

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani külső témavezetőimnek, Dr. Hegyháti Máténak, aki időt nem sajnálva támogatta a munkám és rengeteg értékes szakmai tanáccsal látott el. Köszönettel tartozom belső témavezetőmnek Dr. Tóth Teklának, hogy rendszeres konzultációk során segítette a munkámat. Továbbá köszönöm a Soproni Egyetem és a szlovéniai InnoRenew CoE kutatóintézet munkatársainak, hogy értékes szakmai tapasztalataikkal segítették a kutatásom.

A. függelék

C-EK modell GMPL kódja

```
set Logs;
param logVolume{Logs};

set Lumbers;
set AllCuttingPatterns;
param logParam{AllCuttingPatterns} symbolic in Logs;
set CuttingPatterns{l in Logs} := setof{cp in AllCuttingPatterns:
    logParam[cp] == 1} cp;

param maxExtraAssistantWorker;

param frameSawCuttingTime{AllCuttingPatterns};
param bandSawCuttingTime{AllCuttingPatterns};
param frameSawSpecialistRequirement;
param frameSawAssistantWorkerRequirement;
param bandSawSpecialistRequirement;
param bandSawAssistantWorkerRequirement;
param frameSawSetupTime;
param numberOfDays;
set Days := 1..numberOfDays;
param shiftLength;
param specialistAvailability{Days};
param assistantWorkerAvailability{Days};
```

```
param logInventory{Logs};
param demand{Lumbers, Days};
param yield{Lumbers, AllCuttingPatterns};
param extraCost{Lumbers} default 1;
param storageCapacity;

var frameSawBefore{Days, AllCuttingPatterns} binary;
var frameSawAfter{Days, AllCuttingPatterns} binary;
var changeTime{Days} >= 0, <= shiftLength-frameSawSetupTime;
var frameSawWorking{Days} binary;
var bandSawWorking{Days} binary;
var extraAssistantWorker{Days} >= 0;
var countFrameSawBefore{Days, AllCuttingPatterns} >= 0, integer;
var countFrameSawAfter{Days, AllCuttingPatterns} >= 0, integer;
var countBandSaw{Days, AllCuttingPatterns} >= 0, integer;
var production{day in Days, lumber in Lumbers} >= 0;
var shortage{day in Days, lumber in Lumbers} >= 0;
var storage{day in Days union {0}, lumber in Lumbers} >= 0;

s.t. InitializeStorage{lumber in Lumbers}:
    storage[0, lumber] = 0;

s.t. FrameSawWorkingBeforeConstraint{day in Days}:
    sum{cp in AllCuttingPatterns} frameSawBefore[day, cp] =
        frameSawWorking[day];

s.t. FrameSawWorkingAfterConstraint{day in Days}:
    sum{cp in AllCuttingPatterns} frameSawAfter[day, cp] =
        frameSawWorking[day];

s.t. NoChangeIfBothDaysWork1{day in Days, cp in AllCuttingPatterns:
    day != numberOfDays}:
    frameSawAfter[day, cp] >= frameSawBefore[day+1, cp] - 1 * (2 -
        frameSawWorking[day] - frameSawWorking[day + 1]);
```

```
s.t. NoChangeIfBothDaysWork2{day in Days, cp in AllCuttingPatterns:
    day != numberOfDays}:
    frameSawAfter[day, cp] <= frameSawBefore[day+1, cp] + 1 * (2 -
        frameSawWorking[day] - frameSawWorking[day + 1]);

s.t. BeforeChangeCount1{day in Days, cp in AllCuttingPatterns}:
    countFrameSawBefore[day, cp] <= changeTime[day] /
        frameSawCuttingTime[cp] + shiftLength /
        frameSawCuttingTime[cp] * (1 - frameSawBefore[day, cp]);

s.t. BeforeChangeCount2{day in Days, cp in AllCuttingPatterns}:
    countFrameSawBefore[day, cp] <= shiftLength /
        frameSawCuttingTime[cp] * frameSawBefore[day, cp];

s.t. AfterChangeCount1{day in Days, cp in AllCuttingPatterns}:
    countFrameSawAfter[day, cp] <= (shiftLength - changeTime[day] -
        frameSawSetupTime * (1-frameSawBefore[day, cp])) /
        frameSawCuttingTime[cp] + shiftLength /
        frameSawCuttingTime[cp] * (1 - frameSawAfter[day, cp]);

s.t. AfterChangeCount2{day in Days, cp in AllCuttingPatterns}:
    countFrameSawAfter[day, cp] <= shiftLength /
        frameSawCuttingTime[cp] * frameSawAfter[day, cp];

s.t. MaxBandSawCutting{day in Days}:
    sum{cp in AllCuttingPatterns} countBandSaw[day, cp] *
        bandSawCuttingTime[cp] <= shiftLength * bandSawWorking[day];

s.t. SpecialistAvailabilityConstraint{day in Days}:
    frameSawSpecialistRequirement * frameSawWorking[day] +
        bandSawSpecialistRequirement * bandSawWorking[day] <=
        specialistAvailability[day];
```

```
s.t. ExtraWorkerAvailability{day in Days}:
frameSawAssistantWorkerRequirement * frameSawWorking[day] +
    bandSawAssistantWorkerRequirement * bandSawWorking[day] <=
    assistantWorkerAvailability[day] + extraAssistantWorker[day] +
    specialistAvailability[day] - (frameSawSpecialistRequirement *
    frameSawWorking[day] + bandSawSpecialistRequirement *
    bandSawWorking[day]);

s.t. MaxExtraWorkersHired:
    sum{day in Days} extraAssistantWorker[day] =
        maxExtraAssistantWorker;

s.t. NextDayStorageLimit{day in Days}:
    sum{lumber in Lumbers} storage[day, lumber] <= storageCapacity;

s.t. MaxInventory{l in Logs}:
    sum{day in Days, cp in AllCuttingPatterns: logParam[cp]==1}
        (countFrameSawBefore[day, cp] + countFrameSawAfter[day, cp] +
        countBandSaw[day, cp]) <= logInventory[l];

s.t. ProductionCalculation{day in Days, lumber in Lumbers}:
    production[day, lumber] = sum{cp in AllCuttingPatterns}
        yield[lumber, cp] * logVolume[logParam[cp]] *
        (countFrameSawBefore[day, cp] + countFrameSawAfter[day, cp] +
        countBandSaw[day, cp]);

s.t. ShortageCalculation{day in Days, lumber in Lumbers}:
    production[day, lumber] + storage[day-1, lumber] - storage[day,
        lumber] + shortage[day, lumber] >= demand[lumber, day];

minimize ShortageCost:
    sum{day in Days, lumber in Lumbers} extraCost[lumber] *
        shortage[day, lumber];
```

B. függelék

D[k]-[EK]+ modell GMPL kódja

```
set Logs;
param logVolume{Logs};

set Lumbers;
set AllCuttingPatterns;
param logParam{AllCuttingPatterns} symbolic in Logs;
set CuttingPatterns{l in Logs} := setof{cp in AllCuttingPatterns:
    logParam[cp] == 1} cp;

param maxExtraAssistantWorker;

param frameSawCuttingTime{AllCuttingPatterns};
param bandSawCuttingTime{AllCuttingPatterns};
param frameSawSpecialistRequirement;
param frameSawAssistantWorkerRequirement;
param bandSawSpecialistRequirement;
param bandSawAssistantWorkerRequirement;
param frameSawSetupTime;
param numberOfDays;
set Days := 1..numberOfDays;
param shiftLength;
param specialistAvailability{Days};
param assistantworkerAvailability{Days};
```

```
param logInventory{Logs};
param demand{Lumbers, Days};
param yield{Lumbers, AllCuttingPatterns};
param extraCost{Lumbers} default 1;
param storageCapacity;

param countShiftParts;
set ShiftParts := 1..countShiftParts;
param maxPatternChange default countShiftParts;

var frameSawContinue{Days, AllCuttingPatterns, ShiftParts} binary;
var frameSawChange{Days, AllCuttingPatterns, ShiftParts} binary;
var frameSawWorking{Days, ShiftParts} binary;
var spareTime{Days, AllCuttingPatterns, ShiftParts} >= 0 <=
    shiftLength / countShiftParts;
var bandSawWorking{Days, ShiftParts} binary;
var extraAssistantWorker{Days} >= 0, integer;
var countFrameSaw{Days, AllCuttingPatterns, ShiftParts} >= 0,
    integer;
var countBandSaw{Days, AllCuttingPatterns} >= 0, integer;
var production{day in Days, lumber in Lumbers} >= 0;
var shortage{day in Days, lumber in Lumbers} >= 0;
var storage{day in Days union {0}, lumber in Lumbers} >= 0;

s.t. MaxPatternChange{day in Days}:
    sum{part in ShiftParts, cp in AllCuttingPatterns}
        frameSawChange[day, cp, part] <= maxPatternChange;

s.t. FrameSawWorking{day in Days, part in ShiftParts}:
    sum{cp in AllCuttingPatterns} (frameSawChange[day, cp, part] +
        frameSawContinue[day, cp, part]) = frameSawWorking[day, part];

s.t. FrameSawCutting{day in Days, cp in AllCuttingPatterns, part in
    ShiftParts}:
```

```
frameSawCuttingTime[cp] * countFrameSaw[day, cp, part]
<=
(shiftLength / countShiftParts - frameSawSetupTime) *
    frameSawChange[day, cp, part] + shiftLength / countShiftParts
    * frameSawContinue[day, cp, part]
+ (if part > 1 then spareTime[day, cp, part-1] else 0)
- spareTime[day, cp, part]
;

s.t. SpareTimeCalc{day in Days, cp in AllCuttingPatterns, part in
ShiftParts: part < countShiftParts}:
    spareTime[day, cp, part] <= frameSawCuttingTime[cp] *
        frameSawContinue[day, cp, part+1];

s.t. ChangeBetweenShiftParts{day in Days, cp in AllCuttingPatterns,
part in ShiftParts: part > 1}:
    frameSawContinue[day, cp, part] <= frameSawContinue[day, cp,
        part-1] + frameSawChange[day, cp, part-1] + (1 -
        frameSawWorking[day, part-1]);

s.t. ChangeBetweenDays{day in Days, cp in AllCuttingPatterns: day >
1}:
    frameSawContinue[day, cp, 1] <= frameSawContinue[day-1, cp,
        countShiftParts] + frameSawChange[day-1, cp, countShiftParts]
        + (1 - frameSawWorking[day-1, countShiftParts]);

s.t. StartWithoutChange:
    sum{cp in AllCuttingPatterns} frameSawChange[1, cp, 1] = 0;

s.t. MaxBandSawCutting{day in Days}:
    sum{cp in AllCuttingPatterns} countBandSaw[day, cp] *
        bandSawCuttingTime[cp] <= sum{part in ShiftParts}
        (shiftLength / countShiftParts) * bandSawWorking[day, part];
```

```
s.t. SpecialistAvailabilityConstraint{day in Days, part in
    ShiftParts}:
    frameSawSpecialistRequirement * frameSawWorking[day, part] +
        bandSawSpecialistRequirement * bandSawWorking[day, part] <=
        specialistAvailability[day];

s.t. ExtraWorkerAvailability{day in Days, part in ShiftParts}:
    frameSawAssistantWorkerRequirement * frameSawWorking[day, part]
        + bandSawAssistantWorkerRequirement * bandSawWorking[day,
            part] <= assistantworkerAvailability[day] +
            extraAssistantWorker[day] + specialistAvailability[day] -
            (frameSawSpecialistRequirement * frameSawWorking[day, part] +
            bandSawSpecialistRequirement * bandSawWorking[day, part]);

s.t. MaxExtraWorkersHired:
    sum{day in Days} extraAssistantWorker[day] =
        maxExtraAssistantWorker;

s.t. InitializeStorage{lumber in Lumbers}:
    storage[0, lumber] = 0;

s.t. NextDayStorageLimit{day in Days}:
    sum{lumber in Lumbers} storage[day, lumber] <= storageCapacity;

s.t. MaxInventory{l in Logs}:
    sum{day in Days, cp in AllCuttingPatterns: logParam[cp]==1}
        (countBandSaw[day, cp] + sum{part in ShiftParts}
            countFrameSaw[day, cp, part]) <= logInventory[l];

s.t. ProductionCalculation{day in Days, lumber in Lumbers}:
    production[day, lumber] = sum{cp in AllCuttingPatterns}
        logVolume[logParam[cp]] * (yield[lumber, cp] * sum{part in
            ShiftParts} countFrameSaw[day, cp, part] + yield[lumber, cp]
            * countBandSaw[day, cp]) / 100;
```

```
s.t. ShortageCalculation{day in Days, lumber in Lumbers}:  
    production[day, lumber] + storage[day-1, lumber] - storage[day,  
        lumber] + shortage[day, lumber] >= demand[lumber, day];  
  
minimize ShortageCost:  
    sum{day in Days, lumber in Lumbers} extraCost[lumber] *  
        shortage[day, lumber];
```

Irodalomjegyzék

- [1] *Bútoripar munkaerőpiaci kihívásai*. Hozzáférés dátuma: 2024-12-04. 2022. URL: <https://fataj.hu/2022/09/butoripar-munkaeropiaci-kihivasai/>.
- [2] *Munkaerőhiány a faiparban is*. Hozzáférés dátuma: 2024-12-04. 2022. URL: <https://faipar.hu/cikkek/gazdasag/7922/munkaerohiany-a-faiparban-is>.
- [3] Mario J. Gonzalez. „Deterministic Processor Scheduling”. *ACM Comput. Surv.* 9.3 (1977. szept.), 173–204. ISSN: 0360-0300. DOI: <https://doi.org/10.1145/356698.356700>.
- [4] Michael L. Pinedo. „Overview of Stochastic Scheduling Problems”. *Scheduling: Theory, Algorithms, and Systems*. Boston, MA: Springer US, 2012, 607–610. old. ISBN: 978-1-4614-2361-4. DOI: https://doi.org/10.1007/978-1-4614-2361-4_26.
- [5] J. Mula és tsai. „Models for production planning under uncertainty: A review”. *International Journal of Production Economics* 103.1 (2006), 271–285. old. ISSN: 0925-5273. DOI: <https://doi.org/10.1016/j.ijpe.2005.09.001>.
- [6] Carlos A. Méndez és tsai. „State-of-the-art review of optimization methods for short-term scheduling of batch processes”. *Computers & Chemical Engineering* 30.6 (2006), 913–946. old. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2006.02.008>.
- [7] Iiro Harjunkoski és tsai. „Scope for industrial applications of production scheduling models and solution methods”. *Computers & Chemical Engineering* 62 (2014), 161–193. old. ISSN: 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2013.12.001>.

- [8] Christodoulos A Floudas és Xiaoxia Lin. „Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review”. *Computers & Chemical Engineering* 28.11 (2004), 2109–2129. old.
- [9] C.A. Méndez, G.P. Henning és J. Cerdá. „An MILP continuous-time approach to short-term scheduling of resource-constrained multistage flowshop batch facilities”. *Computers & Chemical Engineering* 25.4 (2001), 701–711. old. ISSN: 0098-1354. DOI: [https://doi.org/10.1016/S0098-1354\(01\)00671-8](https://doi.org/10.1016/S0098-1354(01)00671-8).
- [10] E. Sanmartí, F. Friedler és L. Puigjaner. „Combinatorial technique for short term scheduling of multipurpose batch plants based on schedule-graph representation”. *Computers & Chemical Engineering* 22 (1998). European Symposium on Computer Aided Process Engineering-8, S847–S850. ISSN: 0098-1354. DOI: [https://doi.org/10.1016/S0098-1354\(98\)00163-X](https://doi.org/10.1016/S0098-1354(98)00163-X).
- [11] E. Sanmartí és tsai. „Combinatorial framework for effective scheduling of multipurpose batch plants”. *AIChE Journal* 48.11 (), 2557–2570. old. DOI: <https://doi.org/10.1002/aic.690481115>.
- [12] Olivér Ősz, Dominic CY Foo és Máté Hegyháti. „Minimizing Freshwater Usage in Batch Process Scheduling: S-Graph Approach”. *Process Integration and Optimization for Sustainability* 5.1 (2021), 31–42. old. DOI: <https://doi.org/10.1007/s41660-020-00142-7>.
- [13] Olivér Osz. „Methods for scheduling industrial processes”. Dissz. Széchenyi István University, Hungary, 2022.
- [14] Khaled Mesghouni, Slim Hammadi és Pierre Borne. „Evolutionary algorithms for job-shop scheduling”. *International Journal of Applied Mathematics and Computer Science* 14.1 (2004), 91–104. old.
- [15] John Lancaster és Mustafa Ozbayrak. „Evolutionary algorithms applied to project scheduling problems—a survey of the state-of-the-art”. *International Journal of Production Research* 45.2 (2007), 425–450. old. DOI: <https://doi.org/10.1080/00207540600800326>.
- [16] Olivér Ősz és tsai. „Long-term plant-level scheduling with uncertainty in the plywood industry”. (2022).
- [17] Luca Di Gaspero és tsai. „Local search techniques for scheduling problems: algorithms and software tool”. Forum. 2003.

- [18] Edward Tsang és Chris Voudouris. „Fast local search and guided local search and their application to British Telecom’s workforce scheduling problem”. *Operations Research Letters* 20.3 (1997), 119–127. old. ISSN: 0167-6377. DOI: [https://doi.org/10.1016/S0167-6377\(96\)00042-9](https://doi.org/10.1016/S0167-6377(96)00042-9).
- [19] Andrés Weintraub és Carlos Romero. „Operations Research Models and the Management of Agricultural and Forestry Resources: A Review and Comparison”. *Interfaces* 36.5 (2006), 446–457. old. DOI: <https://doi.org/10.1287/inte.1060.0222>.
- [20] Balázs Dávid, Olivér Ősz és Máté Hegyháti. „Robust Scheduling of Waste Wood Processing Plants with Uncertain Delivery Sources and Quality”. *Sustainability* 13.9 (2021). ISSN: 2071-1050. DOI: <https://doi.org/10.3390/su13095007>.
- [21] László Hargitai. *Fűrészáru*. Budapest: Szaktudás Kiadó Ház Rt., 2003, 172. old. ISBN: 9789639422711.
- [22] Luis G Occena és Daniel L Schmoldt. „GRASP: a prototype interactive graphic sawing program”. *Forest Products Journal* 46.11/12 (1996), 40. old.
- [23] C L Todoroki és E M Rönnqvist. „Combined primary and secondary log breakdown optimisation”. *Journal of the Operational Research Society* 50.3 (1999), 219–229. old. DOI: <https://doi.org/10.1057/palgrave.jors.2600699>.
- [24] Matthew F Winn, Randolph H Wynne, Philip A Araman és tsai. „ALOG: A spreadsheet-based program for generating artificial logs”. *Forest products journal* 54.1 (2004), 62–66. old.
- [25] Celso Carnieri és Guillermo A Mendoza. „A fractional algorithm for optimal cutting of lumber into dimension parts”. *Annals of Operations Research* 95 (2000), 83–92. old. DOI: <https://doi.org/10.1023/A:1018954108382>.
- [26] Ignacio Hinostroza, Lorena Pradenas és Víctor Parada. „Board cutting from logs: Optimal and heuristic approaches for the problem of packing rectangles in a circle”. *International Journal of Production Economics* 145.2 (2013), 541–546. old. ISSN: 0925-5273. DOI: <https://doi.org/10.1016/j.ijpe.2013.04.047>.

- [27] Li Luo és tsai. „An integrated strategic and tactical optimization model for forest supply chain planning”. *Forest Policy and Economics* 131 (2021), 102571. old. ISSN: 1389-9341. DOI: <https://doi.org/10.1016/j.forpol.2021.102571>.
- [28] Pamela P. Alvarez és tsai. „Improving consistency in hierarchical tactical and operational planning using Robust Optimization”. *Computers & Industrial Engineering* 139 (2020), 106112. old. ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2019.106112>.
- [29] Sergio Maturana, Enzo Pizani és Jorge Vera. „Scheduling production for a sawmill: A comparison of a mathematical model versus a heuristic”. *Computers & Industrial Engineering* 59.4 (2010), 667–674. old. ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2010.07.016>.
- [30] Mauricio Varas és tsai. „Scheduling production for a sawmill: A robust optimization approach”. *International Journal of Production Economics* 150 (2014), 37–51. old. ISSN: 0925-5273. DOI: <https://doi.org/10.1016/j.ijpe.2013.11.028>.
- [31] Dimitris Bertsimas és Melvyn Sim. „The Price of Robustness”. *Operations Research* 52.1 (2004), 35–53. old. DOI: <https://doi.org/10.1287/opre.1030.0065>.
- [32] Masoumeh Kazemi Zanjani, Daoud Ait-Kadi és Mustapha Nourelfath. „Robust production planning in a manufacturing environment with random yield: A case in sawmill production planning”. *European Journal of Operational Research* 201.3 (2010), 882–891. old. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2009.03.041>.
- [33] Pamela P Alvarez és Jorge R Vera. „Application of robust optimization to the sawmill planning problem”. *Annals of Operations Research* 219 (2014), 457–475. old. DOI: <https://doi.org/10.1007/s10479-011-1002-4>.
- [34] Nicolás Vanzetti és tsai. „An optimization approach for multiperiod production planning in a sawmill”. *Forest Policy and Economics* 97 (2018), 1–8. old. ISSN: 1389-9341. DOI: <https://doi.org/10.1016/j.forpol.2018.09.001>.

- [35] Diego Broz és tsai. „Goal programming application for the decision support in the daily production planning of sawmills”. *Forest Policy and Economics* 102 (2019), 29–40. old. ISSN: 1389-9341. DOI: <https://doi.org/10.1016/j.forpol.2019.02.004>.
- [36] *Pitago Optimizers*. <https://pitago.eu/>. Hozzáférés: 2024. október 31.
- [37] Csaba Kebelei és Mate Hegyhati. „Sawmill scheduling: an application-oriented model”. *Acta Technica Jaurinensis* 17.3 (2024), 104–110. DOI: <https://doi.org/10.14513/actatechjaur.00743>.

Ábrák jegyzéke

2.1. Fűrészüzemek általános feldolgozási folyamata	9
2.2. Prizma vágás egy példája	11
3.1. Az ütemezendő folyamat	15
3.2. A fűrészüzem ütemezésének egy példája	16
4.1. Átcsúszási idő számolásának egy példája	29
5.1. Rendszerarchitektúra diagram	33
5.2. Egyed-kapcsolat diagram	34
5.3. A szoftver működésének folyamata	34
5.4. Az adatrögzítés egy példája a grafikus felületen	35
5.5. Új vágási minta rögzítése a grafikus felületen	36
5.6. Példa megoldás a grafikus felületen	37
5.7. C-EK modellel tervezett ütemezések gantt diagramjai az extra munkaerő növelésnek változtatásával	40
5.8. Gantt diagramok a különböző MILP variánsok megoldásairól	44
5.9. C-EK modell futási idejei (sec) 1-6 hetes tervezési időszakokra nézve nagyszámú véletlenszerűen generált példák esetén dobozdiagramon ábrázolva	46
5.10. C-EK, $D[2, 4, 8]^+$ és $D[2, 4, 8]^-$ -EK+ optimum (m^3) értékei 4 hetes tervezési időszakokra nézve nagyszámú véletlenszerűen generált példák esetén szórásdiagramon ábrázolva	47
5.11. C-EK és $D[2, 4, 8]^-$ -EK+ modellekkel tervezett ütemezések gantt diagramjai egy 4 hetes tervezési időszakkal rendelkező példára	49
5.12. C-EK és $D[2, 4, 8]^+$, $D[2, 4, 8]^-$ -EK+ modellek futási idejei (sec) 4 hetes tervezési időszakokra nézve nagyszámú véletlenszerűen generált példák esetén dobozdiagramon ábrázolva	50

Táblázatok jegyzéke

5.1. Az üzem alkalmazottainak a rendelkezésre állása	39
5.2. Optimum (m^3) és futási idő (sec) eredmények C-EK, D[2, 4, 8], D[2, 4, 8]-EK modellek esetén az extra munkaerő növelésnek változ- tatásával	42
5.3. Optimum (m^3) és futási idő (sec) eredmények C-EK, D[2, 4, 8]+, D[2, 4, 8]-EK+ modellek esetén az extra munkaerő növelésnek változ- tatásával	43