

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
GÉPÉSZMÉRNÖKI KAR
GYÁRTÁSTUDOMÁNY ÉS -TECHNOLÓGIA TANSZÉK



TDK DOLGOZAT

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
GÉPÉSZMÉRNÖKI KAR
GYÁRTÁSTUDOMÁNY ÉS -TECHNOLÓGIA TANSZÉK

KIS-TAMÁS LEVENTE
AGV flotta irányításának modellezése Siemens Tecnomatix
Plant Simulation szoftverben

Konzulens:

Dr. Németh István

Tanszékvezető-helyettes, egyetemi docens

Budapest, 2024

TARTALOMJEGYZÉK

1. Bevezetés.....	1
1.1. Absztrakt.....	1
1.2. Áttekintés	2
2. AGV-k.....	3
2.1. Anyagmozgatás automatizálása.....	3
2.2. AGV-k jellemzői.....	3
2.3. AMR-ek	4
2.4. AGV flotta irányítása.....	5
3. Szimulációk	9
3.1. Bevezetés	9
3.2. Szimulációk használata	10
3.3. Csoportosítás	11
3.4. Plant Simulation.....	12
3.5. Optimalizálás Plant Simulation-ben	15
4. A mintagyár bemutatása	17
4.1. A mintagyár felépítése és működése	17
4.2. Inicializálás.....	18
4.3. AGV-k irányítása	19
4.4. AGV irányítási stratégiák	21
4.5. Ütközések kezelése	23
4.5.1. Ütközés kereszteződésben.....	24
4.5.2. Hátról belemenős ütközés.....	24
5. Kísérletek és eredmények.....	25
5.1. Teljesítménymutatók.....	25
5.2. AGV-k számának meghatározása	26
5.2.1. Kihozatal szerint	26
5.2.2. Több kimenetei változó figyelembevétele.....	28
5.2.3. Optimalizálás genetikus algoritmussal	31
5.3. AGV irányítási modellek összehasonlítása.....	32
5.4. Legjobb szimulációs paraméterek	34
6. Összefoglalás.....	36
6.1. Továbbfejlesztési lehetőségek	36
6.2. Összefoglalás	37
7. Felhasznált források	38
Jelölések jegyzéke	

A táblázatban a többször előforduló jelölések magyar és *angol* nyelvű elnevezése, valamint a fizikai mennyiségek esetén annak mértékegysége található. Az egyes mennyiségek jelölése – ahol lehetséges – megegyezik hazai és a nemzetközi szakirodalomban elfogadott jelölésekkel. A ritkán alkalmazott jelölések magyarázata első előfordulási helyüknél található.

Rövidítések

Jelölés	Megnevezés, értelmezés
AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robots
DES	Diszkrét Esemény-vezérelt Szimuláció
f_v	függvény
GA	Genetikus Algoritmus
i, j	általános futóindexek (egész szám)
KPI	Key Performance Indicator: fő teljesítmény mutató
MU	Mobil Unit (mozgó egység): a Plant Simulation szoftverben egy mozgó objektumot jelöl, lehet például munkadarab vagy a szállítóeszköz is
nom	névleges (nominális) érték
opt	legkedvezőbb (optimális) érték
PLM	Product Lifecycle Management

1. BEVEZETÉS

1.1. Absztrakt

Az AGV-k („Automated Guided Vehicle”) vezető nélküli járművek, amelyek egyre fontosabb szerepet töltenek be a logisztikai és a modern gyártási folyamatokban. Az AGV rendszerek lehetővé teszik a szállítási feladatok automatizálását, ezáltal növelve az anyagáramlás hatékonyságát és csökkentve a költségeket.

A dolgozat célja egy AGV flotta irányítási stratégiáinak kidolgozása és ezek megvalósítása a Siemens Tecnomatix Plant Simulation szoftverben. A szimulációs szoftver támogatja az AGV-k modellezését, viszont komplexebb rendszerekre, mint az AGV csoport irányítására nem kínál közvetlen megoldást. A feladathoz egy tesztlő gyártási folyamatot készítettem el a Plant Simulation segítségével, amelyben az AGV flotta különböző szállítási feladatokat lát el. Az AGV-k irányításához többféle vezérlési stratégiát dolgoztam ki, beleértve az egy, és a több AGV csoportot használó megoldásokat. A szimulációk során meghatároztam a feladatok ellátásához szükséges minimális AGV-k számát, vizsgáltam a parkolóhelyek kialakítását, valamint a modell figyelembe veszi a járművek ütközéseit és ezek elkerülését is.

Az irányítási modellek értékeléséhez különböző teljesítménymutatókat vezettem be, mint például a termelési kihozatal, az AGV-k száma, az általuk megtett úthossz. A teljesítménymutatók segítségével lehetőség nyílt az AGV flotta irányítási modelljeinek összehasonlítására és a különböző vezérlési stratégiák hatékonyságának értékelésére.

A dolgozatban a maximális kihozatal eléréséhez szükséges járműszám meghatározása után az ideális számú AGV-k számának megtalálását kétféle módszerrel is megközelítettem: először szimulációs kísérletek útján, majd genetikusan alkalmazásával. Ezt követően a GA segítségével a szimuláció optimális paraméterbeállítási variációi is meghatározásra kerülnek.

A szimulációk és az eredmények alapján lehetőség van a gyártási folyamatok további optimalizálására és az AGV flotta hatékonyabb alkalmazására, ezzel támogatva a logisztikai rendszerek költséghatékonyabb működését.

1.2. Áttekintés

A dolgozat első, bevezető fejezetében a dolgozat rövid tartalma és célkitűzései kerülnek bemutatásra. Az absztrakt összefoglalja a dolgozat főbb eredményeit.

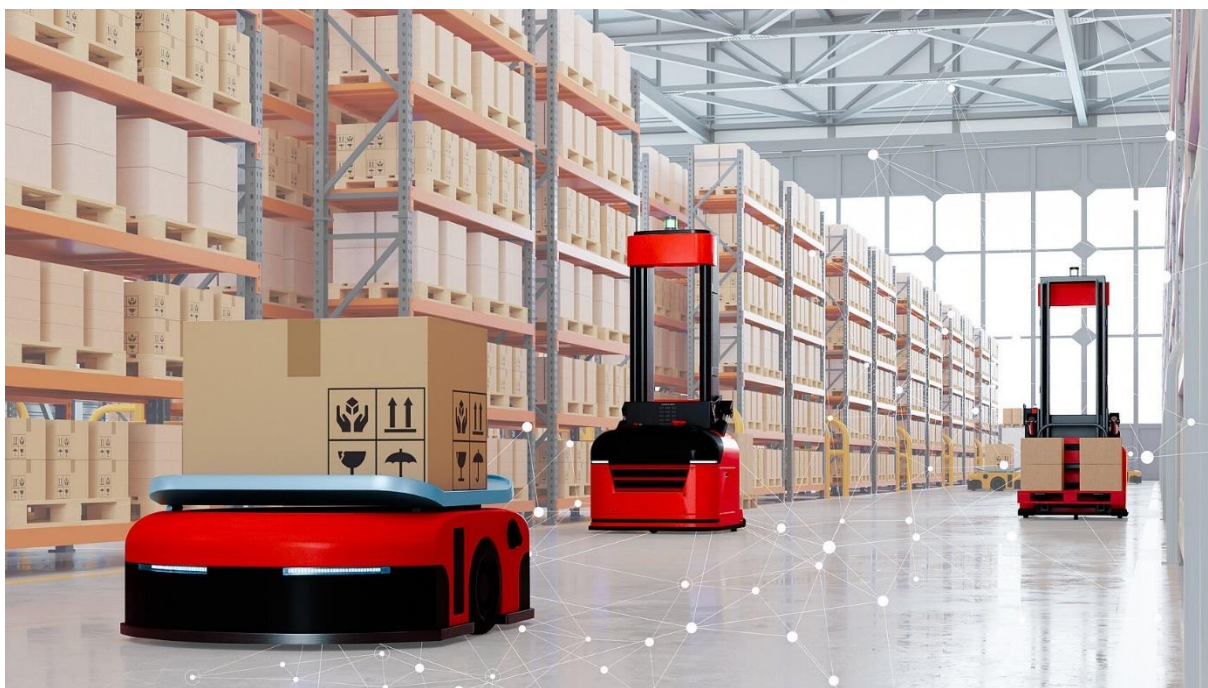
A második fejezet az AGV-k alapjait ismerteti. Bemutatom az anyagmozgatás automatizálásának módjait, az AGV-k és AMR-ek főbb jellemzőit. A fejezetben kitérek az AGV flották irányításának lehetőségeire is.

A harmadik fejezet a szimulációk témakörét tárgyalja, amelyben ismertetem a szimuláció definícióját, előnyeit, alkalmazását, csoportosítási lehetőségeit. A Plant Simulation szoftver külön bemutatásra került, részletezve az AGV rendszerek tervezésének opcióit, valamint az optimalizálási feladatok megtalálásának lehetőségeit a szoftver segítségével.

A negyedik fejezetben kerül bemutatásra a mintagyár felépítése és működése, amely a dolgozat gyakorlati részének dokumentálását tartalmazza. A fejezetben ismertetem az AGV-k irányításának működését és az ütközések kezelését a szimulációs modellben.

Az ötödik fejezet tárgyalja az eredmények kinyeréséhez szükséges kísérletek elvégzését és az optimalizációs feladatok definiálását. Összehasonlításra kerülnek a különböző irányítási modellek, az optimum keresési feladat során pedig meghatározom a legjobb szimulációs paraméterbeállításokat.

A hatodik, záró fejezet összefoglalja a dolgozat tartalmát, eredményeit, valamint tekintést tesz a továbbfejlesztési lehetőségek felé.



1-1. ábra. Példa AGV-k alkalmazására [19]

2. AGV-K

2.1. Anyagmozgatás automatizálása

A modern gyártási és logisztikai feladatok esetében kiemelt cél, hogy az alacsonyabb hozzáadott értékű feladatoknál az emberi munkaerőt automatizált megoldásokkal váltsák ki. Ezáltal nemcsak a munkaerőköltségek csökkenthetők, hanem a folyamatok hatékonysága és pontossága is növelhető, hozzájárulva a versenyképes és fenntartható termeléshez.

Az egyik leggyakrabban automatizált folyamat az üzemben belüli anyagmozgatás. Az anyagmozgató berendezések két csoportra oszthatók: szakaszos és folyamatos gépekre. A folyamatos szállítóeszközök meghatározott irányban mozgatják az anyagot, és állandó helyen működnek, csak a gép bizonyos részei mozognak. Előnyük, hogy magas hozamot képesek mozgatni, ilyenek például a konvektorok és elevátorok.

A szakaszos szállítógépek ezzel szemben megszakításokkal működnek, és egyszerre csak meghatározott mennyiségű anyagot képesek mozgatni. Előnyük azonban, hogy nagyobb rugalmasságot biztosítanak a felhasználás során. Ide tartoznak a rakodógépek, daruk, targoncák, robotok és az AGV-k, AMR-ek is, az 1-1. ábra az utóbbiak alkalmazására mutat példát. [7]

2.2. AGV-k jellemzői

Az AGV-k („Automated Guided Vehicles”) és AMR-ek („Autonomous Mobile Robots”) önvezető járművek, melyek a gyártási és logisztikai területén felmerülő anyagmozgatási feladatok automatizálására alkalmasak. Az AGV-k fő jellemzője, hogy egy fix pályán haladnak végig, a gép egy előre elkészített nyomvonalat tud csak követni. A pályakövetés elve többféle lehet, a legnépszerűbbek az indukzív, fotoszenzoros, mechanikus és a lézeres pásztázó.

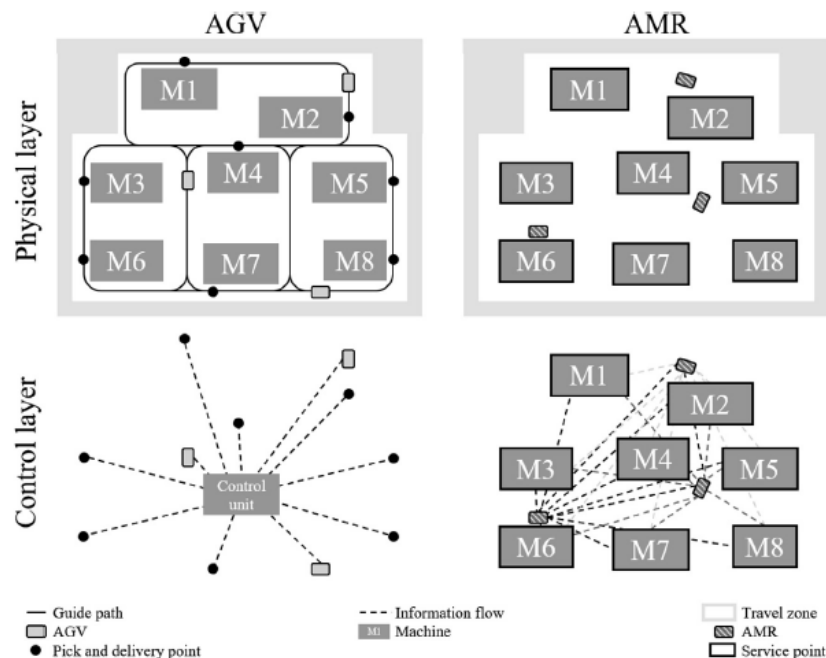
AGV-k esetén azonban mindig szükséges kiépíteni az úthálózatot, amely költséges lehet, valamint a rendszer rugalmasságát is csökkenti. Amennyiben a gyárelrendezést módosítani kell, az utak újra tervezésére és átalakítására is szükség van. A fix útvonal további következménye, hogy ha akadály kerül az AGV elé – bár képes azt érzékelni, kikerülni nem tudja. Ezek miatt az AGV-k olyan területeken használhatók a leghatékonyabban, ahol nem várhatóak gyakori változások a termelési elrendezésben vagy a munkafolyamatokban. [2]

2.3. AMR-ek

Az AMR-ek („Autonomous Mobile Robots”) használata során, ellentétben az AGV-vel, nem szükséges előre telepített útvonalhálózat. (Érdemes megjegyezni, hogy a gyakorlatban az AMR-ek számára is kijelölnek egy pályahálózatot, amely azonban nem feltétlenül konkrét fizikai vonalakkól áll [4]). Az AMR-ek képesek érzékelni és értelmezni környezetüket, így autonóm módon, szabadon navigálhatnak benne. Az útvonaltervezéshez különböző szenzorokat, kamerákat, mesterséges intelligenciát, gépi tanulást és SLAM („Simultaneous Localization and Mapping”) technológiát alkalmaznak. [3]

Amennyiben a robot váratlan akadállyal találkozik, képes lelassítani, megállni és újra tervezni az útvonalát. Az AGV rendszerekben egy központi egység irányítja a döntéseket, például, hogy hova menjen az adott jármű. Az AMR-ek ezzel ellentétben képesek önállóan kommunikálni más gépekkel és rendszerekkel, és egyéni döntéseket hozni. A decentralizált irányítás célja, hogy a rendszer dinamikusan reagálhasson a változó igényekre és környezeti változásokra, ezt szemlélteti az alábbi, 2-1. ábra. Ennek köszönhetően az AMR-ek rugalmasan alkalmazkodnak a környezetükben bekövetkező állapot módosulásokra.

Ezek miatt az AMR-eket olyan területeken is érdemes lehet alkalmazni, ahol gyakran előfordulnak változások a termelési elrendezésben vagy a munkafolyamatokban.

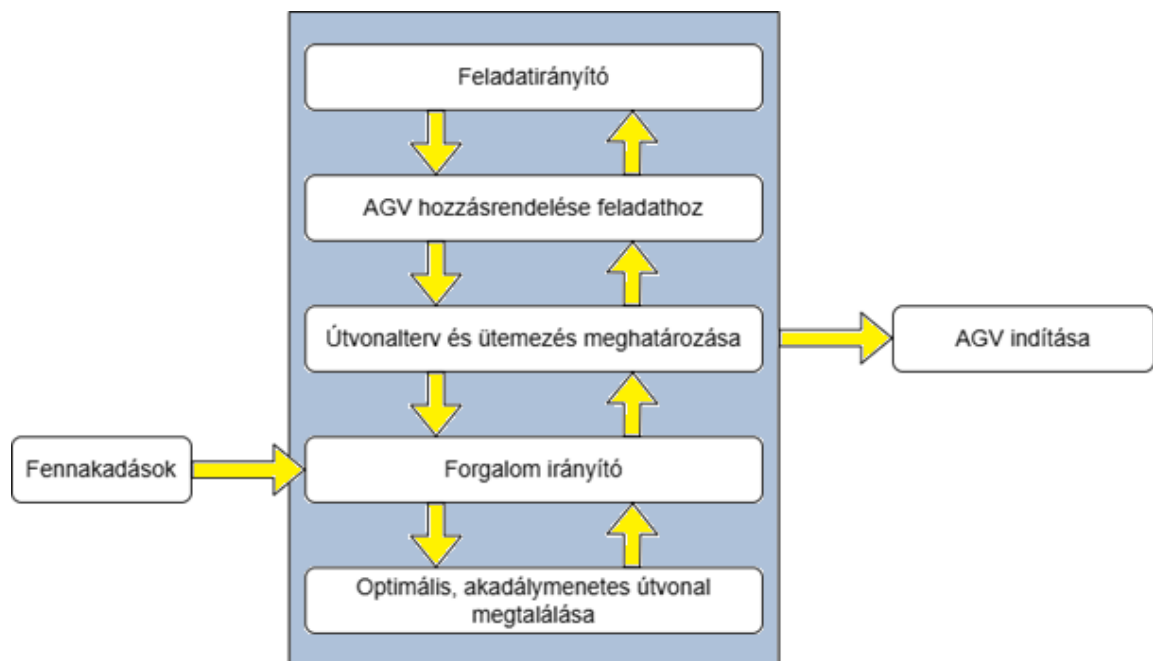


2-1. ábra. Az AGV és AMR vezérlés közötti különbség [3]

2.4. AGV flotta irányítása

Számos szakirodalmi forrás foglalkozik a több AGV-t tartalmazó rendszerek irányításának lehetőségeivel és alkalmazásával. Ebben a fejezetben a legfontosabb irányítási stratégiák kerülnek bemutatásra, amelyek lehetővé teszik az AGV flották optimális működését és integrációját a különböző munkafolyamatokba.

Az AGV flotta irányítása alapvetően két szintre osztható: a feladat- és a forgalom irányításra [4]. A feladatirányítás az AGV-k által elvégzendő szállítási feladatok ütemezését és útvonalának meghatározását foglalja magában, és célja az üres utazások minimalizálása, valamint a várakozási idők csökkentése. A forgalomirányítás az úthálózat állapotát figyeli, beleértve az utakon kialakuló esetleges torlódásokat, illetve az útvonalon megjelenő akadályokat. Ennek a szintnek az a feladata, hogy valós időben optimalizálja az AGV-k mozgását, elkerülve az ütközéseket és biztosítva a zökkenőmentes anyagmozgatást. A 2-2.-es ábrán látható folyamatábra az irányítási lépések logikai sorrendjét szemlélteti.

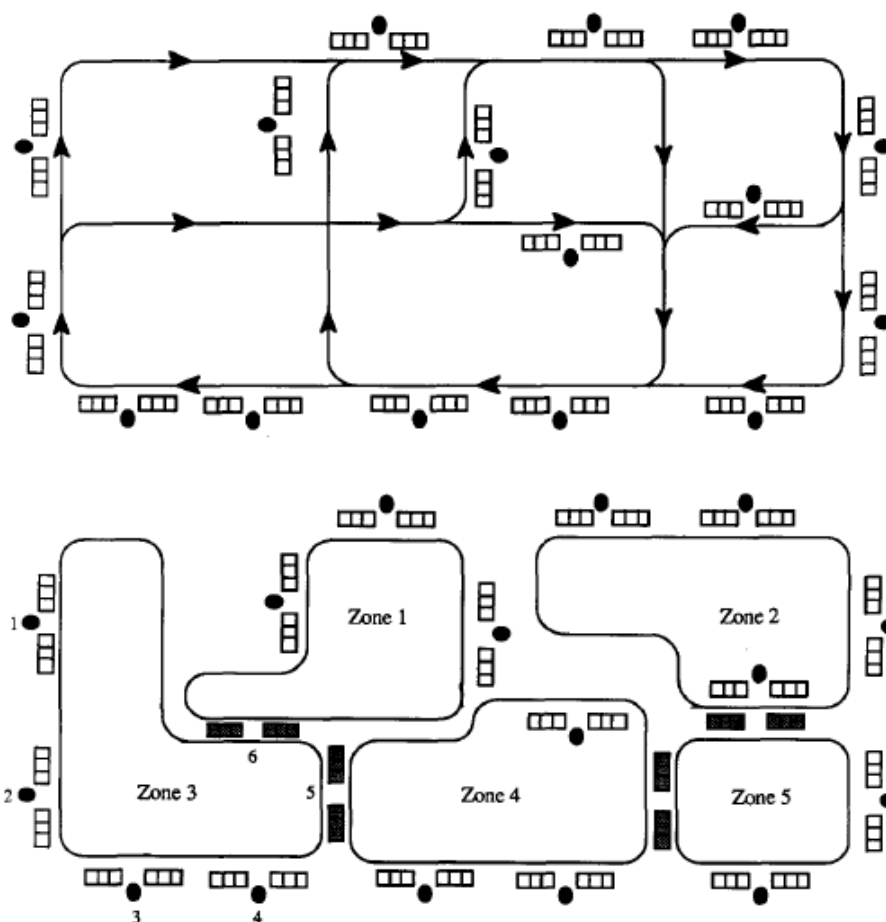


2-2. ábra. AGV vezérlő működése [4]

Az AGV csoportok feladatirányítására számos megoldási opció alkalmazható. Az egyik fő megközelítés a központi vagy az elosztott vezérlés használata. A központi irányításnál egyetlen központi rendszer felügyeli a teljes AGV-flottát, végzi a feladatok hozzárendelését és az útvonalak optimalizálását. Előnye, hogy képes átfogóan figyelembe venni az összes jármű állapotát és az aktuális gyártási igényeket és ezek alapján irányítani az AGV-ket. Hátránya, hogy bonyolultabb, nagyobb teljesítményű központi vezérlőegységet igényel, valamint kevésbé rugalmas és meghibásodás esetén a teljes flotta működése leállhat. [5]

Az elosztott irányítási stratégia ezzel szemben kisebb, önállóan működő csoportokra osztja a gyárat, ahol minden egyes csoport saját irányítórendszerrel rendelkezik. A csoportok közötti koordináció csak szükség esetén valósul meg, a járművek alapból végig a flottához tartozó gyárrészlegen dolgoznak. Az elosztott rendszer rugalmasabb, mivel könnyebben alkalmazkodik a helyi változásokhoz. Az elosztott irányítás egyszerűbb programozási struktúrát igényel, ami csökkentheti a rendszer összetettségét. Emellett előnye, hogy a forgalmi torlódások észlelése és kezelése helyi szinten könnyebbé válik, mivel a csoporton belüli járművek interakciói egyszerűbben kezelhetők. Hátránya, hogy nehezebb a teljes gyártási rendszer globális optimalizálása, mivel a helyi döntések nem feltétlenül veszik figyelembe a gyár teljes területén fellépő igényeket. Emellett kihívást jelent a megfelelő csoportok kialakítása és az ezekhez szükséges erőforrások meghatározása. [5, 6]

Az elosztott irányítási stratégián belül létezik ún. tandem konfiguráció, amelyben az útvonalhálózatot egyszerű, független hurkokra osztják, és mindegyik hurokhoz hozzárendelnek egy AGV-t, ezzel a járművek közötti interakció, a különböző ütközések kizárható. A tandem struktúra felépítésére mutat példát a 2-3. ábra. A hurkok átadó állomásokkal vannak összekapcsolva, ahol a körök közötti anyagáram mozgás valósítható meg. A tandem konfiguráció nagyon egyszerű programot igényel és nem lesznek torlódások a járművek között, azonban az egész üzem működése több AGV-t igényel, valamint lassabb termelés valósítható meg a hurkok közötti anyagátadási idő miatt. [7] [8]



2-3. ábra. Példa hagyományos (felül) és tandem (alul) AGV vezérlés működésére [8]

Létezik olyan vezérlés, amikor az AGV folyamatosan mozog a gyár területén, és csak akkor áll meg, amikor éppen szükség van rá egy adott feladat végrehajtásához, tipikusan ilyen szokott lenni a tandem konfigurációkor. Ez a megközelítés egyszerűbb vezérlést és ütemezést tesz lehetővé, hátránya azonban, hogy ez a stratégia nagyobb energiafogyasztással jár, mivel az AGV-k gyakran üresen mozognak. Elterjedtebb módszer, mikor az érkező feladatok indítják el az AGV-ket és küldik a megfelelő állomásra. Ebben az esetben, ha a járműnek nincs feladata egy parkolóhelybe megy és megáll. Mivel a szállítóeszközök nem mozognak folyamatosan, kevesebb energiafogyasztás érhető el és hatékonyabb termelés valósítható meg, azonban összetettebb irányítást igényel.

A munkadarabot kísérő irányítás egy olyan stratégia, amelyben minden érkező alapanyaghoz hozzárendelnek egy AGV-t, amely végigkíséri a gyártási folyamat összes lépésén. Ezt a megközelítést gyakran dedikált AGV-k alkalmazásának is nevezik, mivel a járművek egy-egy munkadarabhoz kapcsolódnak a teljes folyamat alatt. Előnye, hogy alacsony a szállítóeszközre várakozási idő, valamint, hogy az AGV-k kevés utat tesznek meg. A munkadarab kísérő irányítást főként kis hozamú, alacsony darabszámú termelésekben alkalmazzák. Hátránya, hogy nagyobb kihozatalú gyártások esetén több jármű szükséges, mint más stratégiáknál. [9]

Az AGV-k irányításakor megkülönböztethetünk előrelátó („think-ahead”) és előre nem látó („no think-ahead”) irányításokat [4]. Az előre nem látó irányítás valós időben dolgozik, az AGV az egyes feladatokat a rendszer aktuális állapota szerint kapja meg. Ez a módszer egyszerűbb szabályokat követ, gyorsan reagál a változásokra, viszont a szállítóeszközökre várakozási idő hosszabb lehet. Az előrelátó irányítás ezzel szemben egy, vagy több lépéssel előre megtervezi az AGV-k ütemezését és feladatát, ezzel lehetővé teszi a várakozási idők minimalizálását és a termelékenység növelését. Hátránya, hogy jóval összetettebb vezérlést igényel, nagyobb számítási kapacitással, valamint érzékenyebb a váratlan eseményekre; például egy nem tervezett leállás vagy akadály zavart okozhat, mivel felboríthatja az előre megtervezett optimális útvonalakat és ütemezést.

3. SZIMULÁCIÓK

3.1. Bevezetés

A gyártási folyamatok fejlődésével és az Ipar 4.0 terjedésével a szimulációk szerepe is megnőtt. A szimulációk fontos részét képezik a kiberfizikai rendszerek megvalósításának, mivel lehetővé teszik a valós környezet lemásolását és ezzel összekapcsolva a fizikai és a digitális világot. A szimulációk számos mérnöki problémára képesek segítséget nyújtani, legyen szó kísérletezésről, új rendszer tervezéséről, vagy meglévő üzem optimalizálásáról. Különösen hasznosak, mikor egy problémát már nem lehet, vagy nagyon bonyolult lenne analitikusan megoldani annak bonyolultsága miatt.

VDI („Verein Deutscher Ingenieure, Association of German Engineers”) 3633-as definíciója szerint „a szimuláció egy rendszert és annak dinamikus tulajdonságait képezi le, utánozza le egy olyan modellbe, amivel kísérleteket lehet végezni. Célja a valós rendszerbe átvihető információk szerzése” [10]; tehát a szimuláció egy rendszer reprezentációja, mely segíti a rendszer viselkedésének megértését és működésére vonatkozó jellemzők meghatározását anélkül, hogy a valós rendszeren kísérleteket kellene végezni.

Szimulációk használatának számos előnye van; az egyik legfontosabb, hogy információt tudunk nyerni a rendszerről, segíti annak megértését és optimalizálását, mindent a fizikai rendszeren való vizsgálatok nélkül, gyorsan és kockázatmentesen lehet különböző kísérleteket végezni. AGV rendszerek esetén például a szimuláció lehetővé teszi az irányítási stratégiák tesztelését, valamint vizsgálni lehet a járművek mozgását és állapotát. Szimulációk segítségével optimalizálható a rendszer működése, kihasználtsága, növelhető a termelékenység ezekkel csökkentve a költségeket és növelve a bevételt. A szimuláció során a rendszer vizualizációja nemcsak a rendszer működésének jobb megértését teszi lehetővé, hanem növelheti a termék vonzerejét is, ami fontos tényező lehet az értékesítés során. Továbbá, a szimuláció futási ideje eltérhet a valóstól, így lehetséges a folyamatok felgyorsítása vagy lelassítása: például egy műszak teljesítményének elemzése percek alatt elvégezhető, nem pedig órák alatt.

Bár a szimulációk alkalmazása számos előnnyel jár, vannak hátrányai is. Egy jó szimuláció elkészítése megfelelő képzettséget és tapasztalatot igényel. Fizikai rendszer szimulálása esetén fontos ismerni a modell tulajdonságait, adatokat gyűjteni róla, melyek megszerzése hosszabb idő is lehet. Ha a digitális modell nincs összekapcsolva a fizikai rendszerrel megfelelően, ha nincsen validálva a szimuláció, akkor az így kapott eredmények félrevezetőek lehetnek és hamis eredményeket mutathatnak a valós rendszerről. [11] [12]

3.2. Szimulációk használata

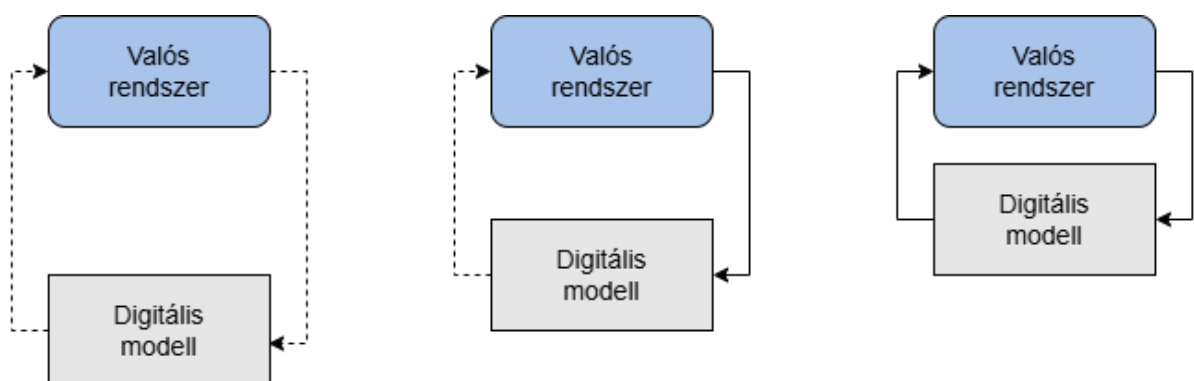
Szimulációkat leggyakrabban három esetben alkalmaznak: új létesítmény tervezésekor, meglévő optimalizálásakor vagy kész tervek bevezetésekor.

Új üzem tervezése előfordulhat például egy zöldmezős beruházás során. Ebben az esetben számos területen használható a szimuláció, ilyenek például a gyárelrendezés és rendszertervezés. Segíthet felderíteni a különböző problémákat még a fizikai létesítmény elkészülte előtt. A gyár jövőbeli viselkedéséről adhat becsléseket, így lehet vizsgálni, hogy az elvárt igényeknek megfelel-e a terv. A szimulációval meghatározhatók a szükséges erőforrások, valamint tesztelhetők vezérlési stratégiák.

Meglévő létesítmény esetén egy meglévő termelést kell optimalizálni, viszont már vannak korlátok, például nem lehet a teljes gyárelrendezést átrendezni. Valós termelési adatok szimulációba integrálásával a digitális modell pontosabb mása lesz a valós rendszernek, megbízhatóbb információkat fog tudni szolgáltatni annak viselkedéséről. A szimulációkkal felderíthetők a különböző hibaforrások, valamint segíthet az összefüggések megértésében.

Szimulációk használhatók kész tervek bevezetésekor, például mikor egy új áru gyártása előtt szimuláció segítségével meghatározzák a várható termelékenységet. Továbbá alkalmazhatók virtuális beüzemeléshez, vagy az operátorok betanítására.

Az Ipar 4.0-val egyre népszerűbb eljárás a digitális iker használata és a virtuális üzembe helyezés, mellyel csökkenteni lehet a kockázatokat és költségeket, valamint felgyorsítani a tényleges üzembe helyezés folyamatát. Napjainkban egy valós rendszernek három különböző szintű szimulációs modelljét lehet elkészíteni: a digitális modell, a digitális árnyék és a digitális iker modelljét.



3-1. ábra. Digitális modell, digitális árnyék és digitális iker

Digitális modellnek tekinthetők a hagyományos szimulációk, mikor nincs folyamatos beérkező jel a valós környezetből a szimulációba. A digitális árnyéknál már érkeznek a rendszer fontos jellemzőiről információk a fizikai rendszerről. A digitális iker egy valós rendszer pontos virtuális másolata, ahol a fizikai környezetből gyűjtött jelek automatikusan feldolgozásra és beépítésre kerülnek a szimulációba, ezen felelül a modellből is be lehet avatkozni a fizikai rendszerbe, ezzel a valós és a digitális világ szorosan összekapcsolódik. [13]

3.3. Csoportosítás

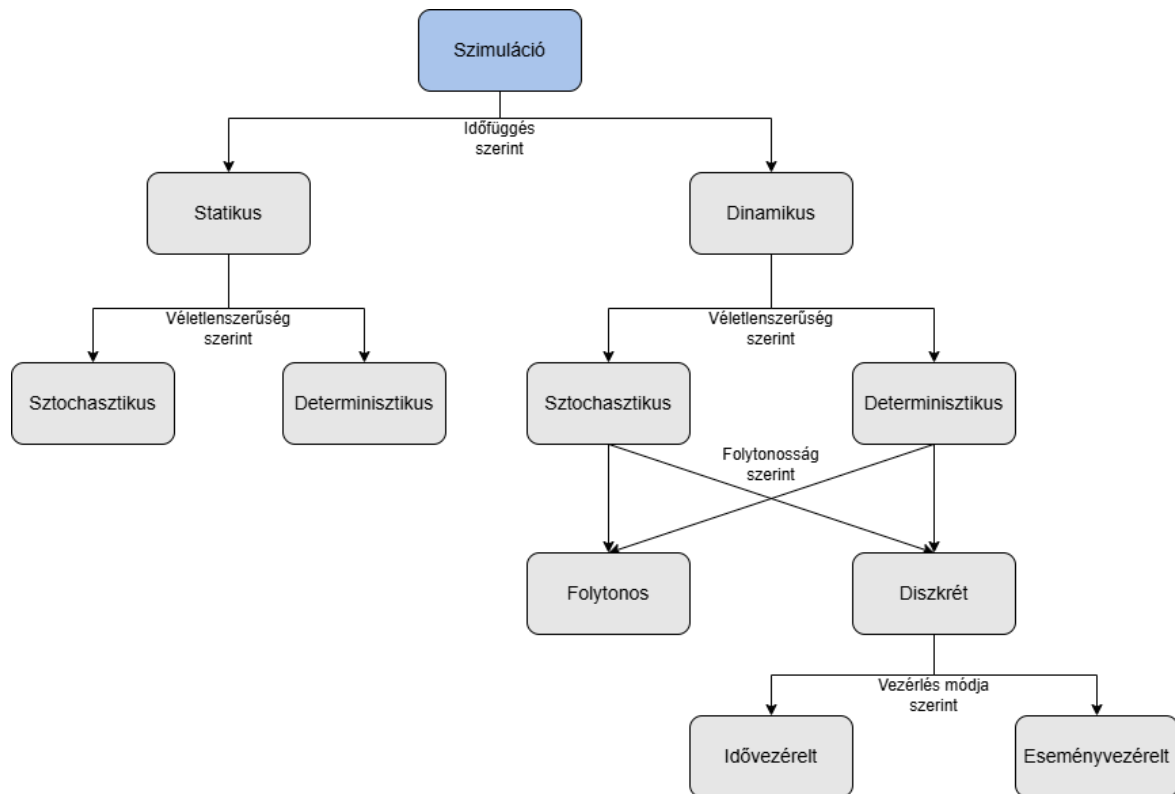
A szimulációk többféleképpen is csoportosíthatók, különböző szempontok alapján.

A véletlenszerűség alapján a szimuláció lehet determinisztikus vagy sztochasztikus. A determinisztikus modellek nem tartalmaznak valószínűségi elemeket, így a kimenet egyértelműen meghatározható, mindig ugyanazt az értéket veszi fel azonos bemeneti paraméterek mellett. Ezzel szemben a sztochasztikus szimuláció tartalmaz véletlenszerű elemeket is, ezért a kimenet nem lesz mindig azonos ugyanazokra a paraméterekre történő futásra.

Időfüggés alapján megkülönböztethetünk statikus és dinamikus szimulációkat. A statikus szimulációk az időtől függetlenek, a rendszernek csak egy adott időpillanatban felvett állapotát mutatják be. Ilyen például a szerkezetek statikus deformációjának vizsgálata. A gyártás területén azonban inkább dinamikus, időtől függő szimulációkat alkalmazunk, amelyekben a rendszer állapota változik az idő múlásával, ilyen például az anyagáram modellezése. Ezekre jellemző, hogy van egy belső memóriájuk, ami a korábbi bemeneteket, változókat és kimeneteket tárolja.

Folytonossága szerint a dinamikus modelleknél megkülönböztethetünk folytonos és diszkrét szimulációkat. A folytonos modellben a rendszer változói időben folyamatosan változhatnak, míg diszkrét szimulációk esetében a változás csak meghatározott időpontokban történnek.

Vezérlés módja szerint a diszkrét modellek lehetnek idő-vezérelt vagy esemény-vezérelt szimulációk. Az idő-vezérelt szimulációkban a rendszer változásai meghatározott időközönként történnek, míg az esemény-vezérelt szimuláció csak bizonyos események hatására fog változni. Amíg az állapotváltozást kiváltó esemény nem következik be, addig a modellen nem történik lényegi változás. Például, ha egy konveijoron haladó termék a szalag végére ér, akkor ez lehet egy ilyen kiváltó esemény. Az esemény-vezérelt szimulációk általában gyorsabban futnak le, mint az idő-vezérelt modellek, mert csak a releváns eseményeknél frissítik állapotukat. [14]

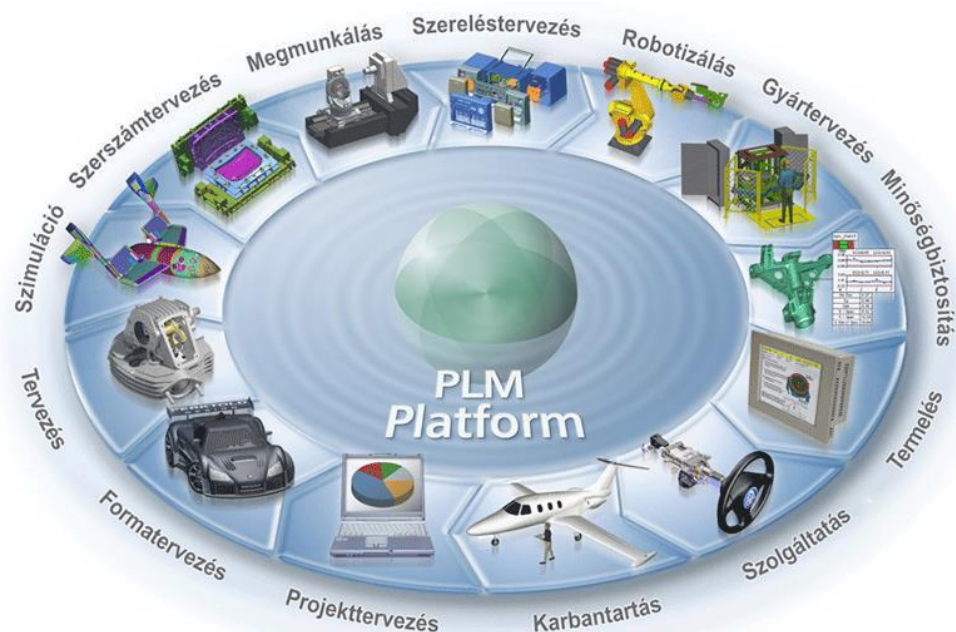


3-2. ábra. Szimulációk csoportosítása

3.4. Plant Simulation

A Siemens számos olyan szoftvert és terméket kínál, melyek lefedik a gyártással kapcsolatos teljes palettát, a fizikai megvalósítástól a digitális tervezésig. Ennek köszönhetően könnyen megvalósítható a kommunikáció a gyártás során használt alkalmazások között és összekapcsolhatók a valós és virtuális rendszerek.

Míg a Siemens TIA Portal („Totally Integrated Automation”) a fizikai rendszerek megvalósításához és irányításához nyújt megoldásokat (pl. PLC-k), addig a PLM („Product Lifecycle Management”) a digitális gyár területét foglalják magukba, mint például a szimulációkat és digitális tervezést. A PLM, magyarul termék életciklus menedzsment három fő komponensből áll, ezek az NX, a Teamcenter és a Tecnomatix, melybe a Plant Simulation szoftver is tartozik. A gyártóeszközök, gépek tervezéséhez kapcsolódó feladatokat az NX látja el, a projekttervezés és karbantartást a Teamcenter. A Tecnomatix a gyártástervezés, a robotizálás és a termelési folyamatok szimulálására nyújt megoldásokat, például robotos cella esetén a Process Simulate alkalmazást, míg gyártósor modellezése esetén a Plant Simulation szoftvert kínálja. [15] [16]



3-3. ábra. Siemens PLM komponensei [16]

A Siemens Tecnomatix Plant Simulation egy diszkrét esemény-vezérelt szimulációs szoftver. A gyártásban a gyártórendszerek, az anyagáram és a logisztikai műveletek modellezésére, szimulálására egyik leggyakrabban használt, legerterjedtebb alkalmazás. Használatával javítható a termelés, optimalizálható az erőforrás kihasználtság és így csökkenthetők az üzemeltetési vagy beruházási költségek.

A Plant Simulation szoftver objektum orientált felépítésű, így fellelhető az OOP (objektum orientált programozás) ismert tulajdonságai, mint az öröklés, egységbezárás és a polimorfizmus. A szimuláció objektumokból épül fel, melyek szülő-gyermek hierarchiából állnak, a szülő objektum példányai (gyerekei) megkapják az előd tulajdonságait. Ez a struktúra megkönnyíti a modellek építését, programozását; a bonyolultabb rendszerek átláthatóbbá válnak, valamint a szimulációk és az objektumok tulajdonságai könnyen módosíthatók.

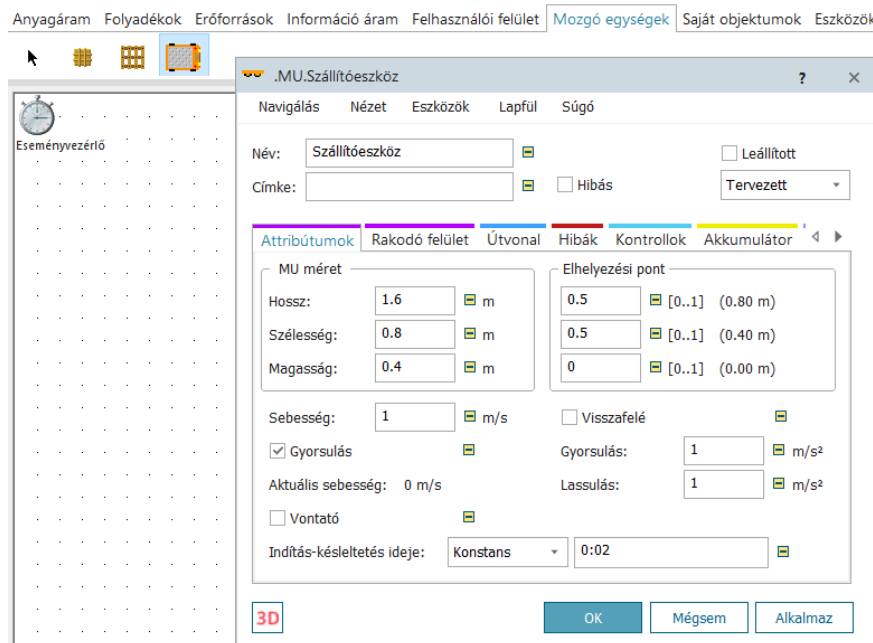
A Plant Simulation-ben az egyszerű modellek létrehozása hasonlóan történik, mint más szimulációs szoftverekben: a felhasználó kiválasztja az objektumokat az osztálykönyvtárból, majd azokat behúzza a modellbe. A szoftver könyvtárai tartalmazzák az alapvető eszközöket, amelyek a különböző gyártási és logisztikai folyamatok modellezéséhez szükségesek. A legfontosabb osztályok közé tartoznak az anyagáram, az erőforrások, az információs áram, a felhasználói felület és a mozgó egységek. Ezekben találhatóak a leggyakrabban használt objektumok, amelyek segítségével a modell gyorsan és hatékonyan felépíthető. Ezekre felül további könyvtárak is importálhatók, illetve a felhasználók saját objektumokat is létrehozhatnak a speciálisabb feladatok megoldására.

Azonban pusztán a beépített objektumokkal általában a komplexebb feladatok nem megoldhatóak. A szoftverben emiatt lehetőség van a programozásra, mely növeli az alkalmazás rugalmasságát és lehetővé teszi az egyedi viselkedések és logikák megvalósítását. A programot úgynevezett metódusokban lehet megírni, amelyek bizonyos események bekövetkezésekor futnak le. A szoftverbe integrált SimTalk programozási nyelv révén elérhetőek és módosíthatók az objektumok attribútumai, segítségével definiálhatók a modell működését meghatározó feladatok.

A Plant Simulation alkalmas statisztikák és kimutatások készítésére. Könnyen előállíthatók különféle teljesítménymutatók (KPI-k), erőforrás- és rendelkezésre állási statisztikák, amelyek támogatják a szűk keresztmetszetek és optimalizálási lehetőségek azonosítását. A felhasználók diagramokat, grafikonokat és jelentéseket készíthetnek, amelyek részletes információkat nyújtanak a termelés hatékonyságáról.

A szoftver továbbá képes kettő- és háromdimenziós animációk megjelenítésére, és támogatja a CAD modellek importálását, ezzel látványosabbá téve a szimulációkat. Bár ezek az elemek legtöbbször nem befolyásolják a szimulációs eredményeket, segítik a rendszer működésének megértését.

A Plant Simulation az AGV-k modellezésére a mozgó egységek (MU) könyvtárában található szállítóeszköz objektumot kínálja megoldásként (3-4. ábra). Ennek számos paramétere beállítható, például a méretek, a sebesség, a gyorsulása és a szállítási kapacitása, amelyek lehetővé teszik a jármű részletes konfigurálását. Lehetőség van meghibásodások, leállások definiálására, valamint az AGV-k energiafelhasználásra vonatkozó adatait is meg lehet adni. [17]



3-4. ábra. AGV objektum Plant Simulation-ben

Az AGV-k útjának modellezésére a szoftver két fő megoldást kínál: a nyomvonalakkal, valamint a jelölőpontokkal definiált útvonalak használatát. Nyomvonalak használata esetén először ki kell alakítani az útvonalhálózatot, amely mentén a járművek közlekedhetnek. Az útvonalak lehetnek egy- és kétirányúak, modellezésükben hasonlítanak a konvektorok felépítéséhez és működéséhez. AGV-k célobjektumra küldésének két alapvető módja van: a szállítóeszköz „targetPosition” vagy „destination” attribútumának beállítása. A „targetPosition”-ben egy adott útszakasz konkrét pontját kell megadni (ez lehet például egy szenzor helye). A „destination” attribútum objektum típusú, ahova az elérni kívánt útszakaszt lehet megadni. Ha több módon is el lehet jutni a célobjektumhoz, akkor a Plant Simulation beépített útkereső algoritmus megtalálja a két pont között a legrövidebb útvonalat.

A másik módszer a jelölőket („marker”) használó megoldás, ahol pontok segítségével lehet megadni az útvonal kezdeti, köztes és vég pontjait. A célobjektumra küldés a legkönnyebb módon a „setRoute” paranccsal valósítható meg, amivel megadható a jelölőpontok sorrendje, melyeken keresztül haladnia kell az AGV-nek. Ennek a módszernek az előnye a nagyobb szabadság, amely révén többféle útvonal bejárható, azonban akadályokkal, például gépekkel teli modellek esetében a „setRoute” parancsnál sok pontot kell megadni, hogy az AGV kikerülje az objektumokat. Az ütközések kezelése a szoftver korábbi verzióiban nem állt rendelkezésre a jelölőpontok használata esetén, azonban a Plant Simulation 2201 (v17) verziótól kezdve már beépített ütközéskezelési funkciókat is kínál [18]. [17]

3.5. Optimalizálás Plant Simulation-ben

A Plant Simulation számos lehetőséget kínál a termelési folyamatok optimalizálására. Ebben segíthetnek a különböző statisztikai elemzési eszközök, a vizualizációs eszközök, a diagrammok készítése, a kísérletkezelő objektum, valamint a genetikus algoritmusok használatát támogató GAWizard. [17]

A kísérletkezelő („Experiment Manager”) eszköz lehetővé teszi a különböző paraméterkombinációk lefuttatását és automatikus összehasonlítását, ezzel segítve a legjobb konfiguráció megtalálását. A felhasználók megadhatják a vizsgálni kívánt paramétereket és tartományaikat, majd a szoftver automatikusan végrehajtja a szükséges kísérleteket. Azonban sok paraméterkombináció esetén a kísérletek futási ideje nagyon hosszúra nyúlhat, ekkor másféle eljárással érdemes az optimumot megtalálni.

Az optimum megtalálása lehetséges különböző algoritmusokkal is, amelyek egy célfüggvény szélsőértékét keresik. Az algoritmusok általában három elv alapján működhetnek: kimerítő keresés („brute force”), gradiens-alapú keresés és véletlen alapú algoritmusok. A keresés során fontos szempont, hogy az algoritmus képes legyen megtalálni a globális optimumot, és ne ragadjon be egy lokális szélsőértéknél.

Az egyik legnépszerűbb optimum-kereső algoritmus a genetikus algoritmus. A GA evolúciós algoritmus, ahol a generációk során a legjobb fitness értékű egyedek maradnak bent. Az egyedek szelekcióval, mutációval és keresztezéssel javulnak a generációk haladásával [18]. A Plant Simulation támogatja a genetikus algoritmus használatát, amelyhez a GAWizard nevű objektum áll rendelkezésre. A GAWizard nagyban leegyszerűsíti a GA használatát, alkalmazásához meg kell adni a fitness függvényt, az optimalizálandó paramétereket, valamint a generációk számát és méretét. A mutáció értéke és más változók alapértelmezett beállításokkal rendelkeznek a Plant Simulation-ben. Ezzel a GAWizard jelentősen leegyszerűsíti az optimumkeresési feladatot, lehetővé téve a genetikus algoritmus hatékony alkalmazását a szimulációs modellben.

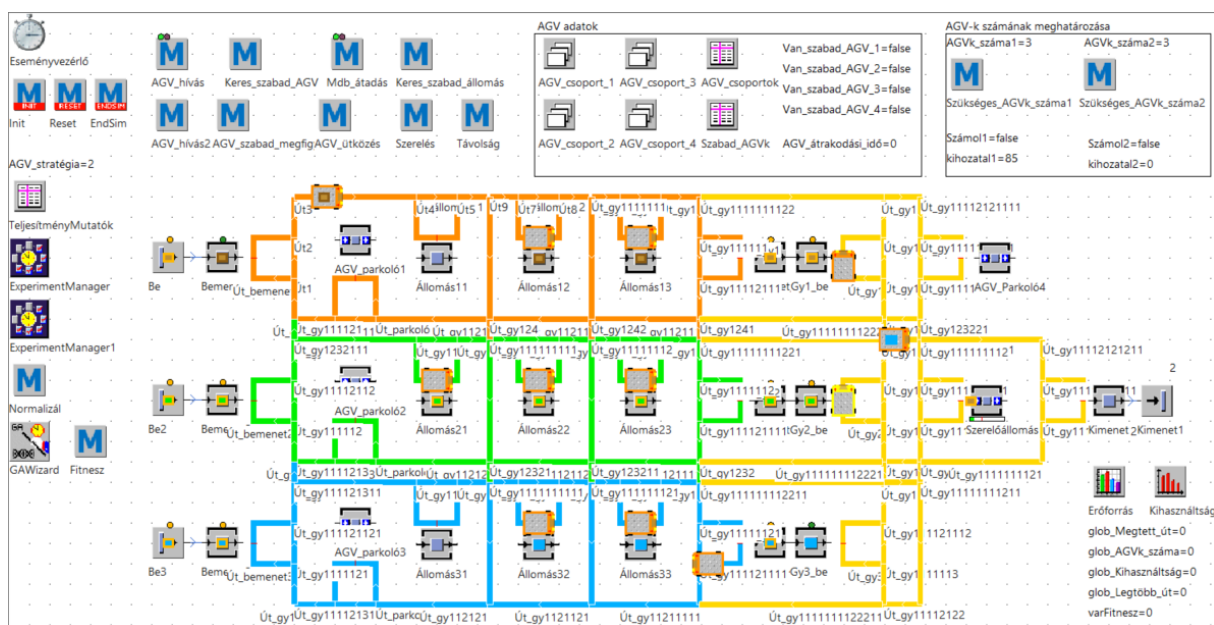
4. A MINTAGYÁR BEMUTATÁSA

Az AGV flotta irányításának vizsgálatához egy mintagyár modellt készítettem a Tecnomatix Plant Simulation (2302) szoftverben. Ebben a fejezetben bemutatásra kerül a szimulációs modell felépítése és működése, valamint az AGV csoportok irányításának megvalósítása.

4.1. A mintagyár felépítése és működése

A mintagyár egy tetszőleges gyártási folyamatot modellez, ahol a munkadarabok szállítását az AGV-k végzik. Az üzembe háromfajta alapanyag érkezik három bemenetről. A különböző munkadarabok párhuzamos gyártósorokon, azonos munkafolyamaton mennek végig. Egy gyártósor három eltérő megmunkálóállomásból áll, melyeken sorban végig kell haladnia az alapanyagoknak. A gyártósorok végén elkészülő munkadarabok egy következő állomáson összeszerelésre kerülnek és végül az elkészült terméket az AGV-k a kimenetre szállítják.

A mintagyár elrendezése a 4-1. ábrán látható. A különböző alapanyagok és a gyártósorokat behálózó AGV útvonalak eltérő színekkel vannak modellezve. Például egy zöld színű munkadarab a zöld színű soron megy keresztül. A sárga színű részen a három gyártósorton elkészült gyártmányokat elszállítják a szerelőállomásra, ahonnan később az összeszerelt terméket a kimenetre viszik az AGV-k.



4-1. ábra. A mintagyár

A szimulációs modellben található hasonló tulajdonságú objektumok egy szülőobjektum leszármazottjai, melynek jellemzőit az utódobjektumok örökölik. Az állomások a Plant Simulation anyagáram-eszköztárában található „Station” objektumokkal lettek modellezve, a feldolgozási idejük alapértelmezetten 30 másodpercesre lett beállítva, ez a „szülő” állomáson később könnyen módosítható. Vannak azonban 0 másodperc feldolgozási idővel rendelkező állomások, ilyen található például a kimeneten. Ezek a helyeken a munkadarab nem tartózkodik, hanem azonnal továbbhalad, mivel nem végez rajta semmilyen műveletet. Funkciója csupán a programozás és a modell vizsgálatának egyszerűsítését szolgálja.

Az AGV-k a Plant Simulation mozgó egységek könyvtárában található szállítóeszköz objektummal lettek modellezve. A járművek szállítási kapacitása 1, tehát egyszerre csak egy munkadarabot képes szállítani. A sebessége 1 [m/s] és gyorsulása pedig 1 [m/s²]. A járművek alaphelyzetben egy AGV parkolóból indulnak és oda térnek vissza, ha nem érkezik új megrendelés. Az AGV foglaltságát egy „szabad” nevű, bool típusú felhasználói attribútum tárolja el. Ha egy feladatra lefoglaljuk az adott járművet, a szabad attribútuma hamis értéket vesz fel a feladat teljesítéséig. Az AGV szabad, amikor nincsen megadva neki célállomás, vagy a célállomása a parkolóhely.

Az AGV-k útjának modellezése a 2.2.4. *fejezetben* bemutatott két módon lehetséges: nyomvonalakkal és jelölők használatával. Én előbbit választottam, mivel ebben az esetben egyszerűbb a jármű céljának eléréséhez definiálni a pályát, ezenfelül az ütközéseket is könnyebb figyelembe venni és elkerülni. Az úthálózat egyirányú utakból épül fel, több hurkot is tartalmaz és úgy lett kialakítva, hogy bármelyik helyről bárhova el lehessen jutni. A Plant Simulation beépített útkereső algoritmus megtalálja két pont között a legrövidebb útvonalat. Minden állomás előtt található egy külön leálló útszakasz, ahol a munkadarab átrakodás zajlik az AGV és az állomás között, ezzel a jármű nem tartóztatja fel a forgalmat a főútvonalon.

4.2. Inicializálás

A szimuláció indításakor először az „Init” nevű inicializáló metódus fut le. Ez a metódus készíti elő a modellt a kiindulási állapotba, létrehoz objektumokat, beállítja azok attribútumait, továbbá változókat és táblázatokat definiál.

A szimulációban paraméterként meg lehet adni, hogy hány AGV legyen alapállapotban az egyes parkolóhelyeken, valamint, hogy milyen irányítási stratégia vezérelje a járművek mozgását.

A mintagyár üzemben a gyártási folyamat négy részre osztható, melyek mindegyikéhez tartozik egy-egy AGV parkolóhely. A különböző csoportokban az útvonalak eltérő színnel vannak megjelenítve. Az Init metódus a szimuláció indulása előtt létrehozza a megadott számú AGV-t a megfelelő parkolóhelyeken. Az AGV-ket táblázatokba elmenti, külön oszlopokba a külön parkolóhelyen található gépeket. A szabad

járműveket a szabad_AGVk nevű táblázatba menti (4-2. ábra), ami kiindulási állapotban megegyezik az összes szállítóeszközt tartalmazó táblázattal (mivel alapállapotban az összes jármű szabad), viszont a szimuláció futása során folyamatosan változni fog. (4-3. ábra)

	object 1	object 2	object 3	object 4
1	*.SajátObjektumok.AGV:1	*.SajátObjektumok.AGV:6	*.SajátObjektumok.AGV:11	*.SajátObjektumok.AGV:16
2	*.SajátObjektumok.AGV:2	*.SajátObjektumok.AGV:7	*.SajátObjektumok.AGV:12	*.SajátObjektumok.AGV:17
3	*.SajátObjektumok.AGV:3	*.SajátObjektumok.AGV:8	*.SajátObjektumok.AGV:13	*.SajátObjektumok.AGV:18
4	*.SajátObjektumok.AGV:4	*.SajátObjektumok.AGV:9	*.SajátObjektumok.AGV:14	*.SajátObjektumok.AGV:19
5	*.SajátObjektumok.AGV:5	*.SajátObjektumok.AGV:10	*.SajátObjektumok.AGV:15	
6				

4-2. ábra. A szabad AGV-ket tartalmazó táblázat inicializáláskor.

	object 1	object 2	object 3	object 4
1	*.SajátObjektumok.AGV:1	*.SajátObjektumok.AGV:6	*.SajátObjektumok.AGV:11	
2				
3		*.SajátObjektumok.AGV:8	*.SajátObjektumok.AGV:13	
4				
5	*.SajátObjektumok.AGV:5			
6				

4-3. ábra. A szabad AGV-ket tartalmazó táblázat a szimuláció futása közben.

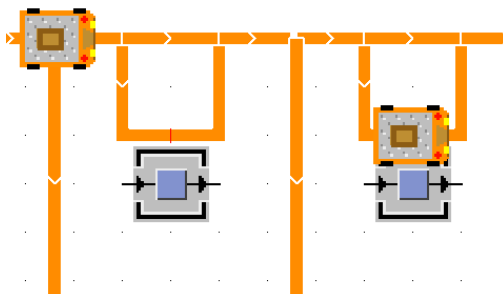
Minden állomás előtt található egy útszakasz, ahol a munkadarab átadásra kerül, ezt össze kell kapcsolni az adott állomással: az állomásokhoz hozzárendeljük az útszakaszt, valamint visszafele is, az útszakaszhoz hozzárendeljük a megfelelő állomást, így később programozás szempontjából könnyebb lesz hivatkozni rájuk. Ezeket a hozzárendeléseket az objektumok egy felhasználói attribútuma tárolja el. Az állomásoknak ezen felül további két jellemzője lett deklarálva: a következő állomás és a szabad attribútum. A következő állomás a gyártás során az állomást követő folyamat neve, lényegében az AGV-nek ezzel lehet megadni majd az új célállomást. A szabad attribútum pedig egy logikai típusú jellemző, az állomás elérhetőségét és foglaltságát figyeli. Értéke igaz, ha az állomáson nem folyik művelet, illetve, ha egyik AGV sem tart oda (az AGV-k célobjektuma nem egyezik meg az állomással).

4.3. AGV-k irányítása

Az AGV-k különböző stratégiák szerint irányíthatóak, de a működés elve, a főbb lépések és függvények minden esetben azonosak. A két fő metódus, ahol az AGV-k irányítása zajlik, az „AGV_hívás” és a „Mdb_átadás” nevű objektumok. Az járműveket elsősorban az AGV „destination” nevű attribútum megadásával küldik el a kívánt célállomáshoz.

Az „AGV_hívás” metódus az állomások kimeneti kontrollja, akkor fut le, mikor az adott állomáson elkészült a munkadarab. Először megvárja, míg a következő állomás üres és szabad, hogy tudja majd hova szállítani az terméket és feleslegesen ne foglaljon le erőforrást. Ezt követően megvárja míg az adott csoportban van szabad jármű, ezt csoportonként egy logikai típusú globális változóban tárolom (mivel Plant Simulation-ben nem lehet „waituntil” után táblázatot vagy annak celláját megadni feltételként). Ha van szabad szállítóeszköz, akkor lefut a „Keres_szabad_AGV” függvény. A szabad AGV keresés során a „Szabad_AGVk” táblázat elemeit vizsgálja, és kiszámolja a hívó-állomás (pontosabban a hívóállomás előtti útszakasz) és a jármű közötti távolságot, és a legközelebbi AGV-t kiválasztja. Ha az így megtalált AGV parkolóban van, akkor először felhelyezi az úthálózatra. A jármű célobjektumát beállítja a hívóállomáshoz tartozó útszakaszra, és lefoglalja az adott AGV-t és állomást (a szabad jellemzőit hamisra állítja). Ekkor a szállítóeszköz elindul a megadott helyre.

Minden állomás előtt található egy hozzátartozó útszakasz, amin egy szenzor található, ez jelöli az MU átadásának helyét. Ennek a szenzornak a kontrollja a „Mdb_átadás” metódus, ami akkor fut le, ha a szenzorra érkezik egy szállítóeszköz. Az érzékelőhöz érve az AGV először megáll; ekkor, ha parkoló előtti szenzorhoz ért és nincs új megrendelés, akkor bemegy a parkolóba, ha pedig állomáshoz ért, akkor két opció lehetséges: az AGV üres és fel kell tölteni termékkel az állomásról, vagy az AGV szállította a munkadarabot és át kell helyezni azt az állomásra. Amikor a jármű üres, akkor az állomásról átveszi a terméket, és elszállítja a következő állomásra (aminek nevét az állomás egy felhasználói attribútumában tárolja). Az állomás ekkor felszabadul, viszont az AGV és a következő állomás szabad jellemzője hamis lesz, ezzel a következő feladatra lefoglalta a szükséges erőforrásokat és végül elindítja az AGV-t. Másik esetben a jármű nem üresen érkezik az érzékelőre, ekkor a rajta található munkadarabot áthelyezi az állomásra, és utána az új célobjektuma a parkolóhely lesz és az AGV szabadná válik.

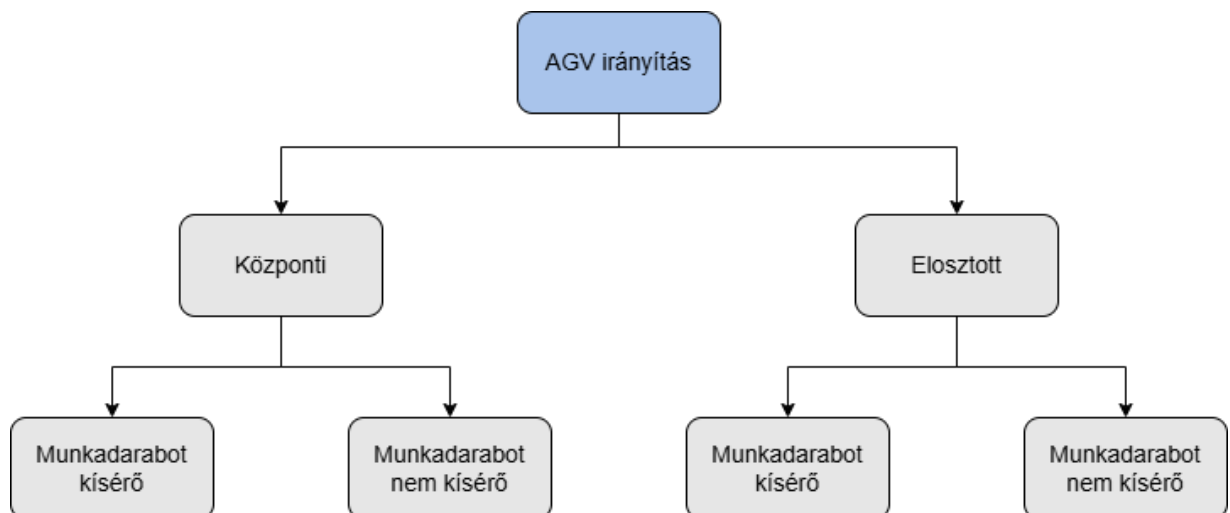


4-4. ábra. A munkadarab átadás helye. Látható az állomások előtt kialakított útszakasz rajta a szenzorral, aminek kontrollja a „Mdb_átadás” metódus.

A Plant Simulation diszkrét esemény-vezérelt szimuláció, ezért a metódusokat is valamilyen esemény bekövetkezése futtatja le. Például a szenzorra érkezik egy MU, ez az esemény indította az fent tárgyalt „Mdb_átadás” metódust. Ez a vezérlő esemény lehet azonban a felhasználó saját maga által definiált attribútumban történő változás is, ezekre az esetekre úgynevezett megfigyelőket lehet készíteni. A megfigyelő lényegében a vizsgált jellemző változásakor lefutó metódus. A mintagyárban az AGV-k „szabad” felhasználói tulajdonságára létre lett hozva egy megfigyelő-metódus, ami bármely szállítóeszköz szabad jellemzőjének állapotváltozásakor lefut. Ez a metódus frissíti a szabad AGV-k táblázatát: mikor a jármű foglalttá válik, akkor kitörli az adott cellából; ha viszont éppen felszabadult, akkor beírja a megfelelő helyre. Az AGV szabad állapotváltásakor továbbá megvizsgálja, hogy üres-e a táblázat adott oszlopa; ha igen, akkor nincs szabad jármű azon a gyárrészen (ezt a változót kell a szabad AGV kereső metódus elé „waituntil” feltételként megadni).

4.4. AGV irányítási stratégiák

A szimuláció futtatásakor paraméterként szerepel az irányítási stratégia, melynek megfelelően fog a szimuláció később működni. A mintagyárban számos stratégia került kidolgozásra, viszont ezek négy osztályba sorolhatók: az AGV kontrol lehet központi, vagy elosztott, ezeken belül pedig lehet munkadarabot kísérő, illetve nem kísérő stratégia.



4-5. ábra. A modellben megvalósított AGV irányítási stratégiák osztályozása

Központi irányítás esetén az egész gyárat egy csoportnak tekintjük és az AGV-ket az üzem bármely pontjáról bárhova lehet küldeni. Elosztott stratégia esetén a gyártási folyamatokat alcsoportokra lehet osztani, és a csoportokban történő anyagszállítási feladatokért külön AGV flották felelősek, ebben az esetben a járművek nem mehetnek bárhova.

A mintagyárban a csoportok egyértelműen elkülöníthetők; az útvonalszínezés megmutatja az elosztott irányítás esetén kialakuló négy külön csoportot. Mindegyik csoporthoz tartozik egy-egy parkolóhely, melyre a szimuláció indításakor az Init metódus létrehoz megadott számú AGV-t. Ha felszabadul egy szállítóeszköz, akkor a csoportjában található parkolóhelyre megy.

Központi irányítás esetén az Init függvény csak egy darab központi parkolóra hozza létre az AGV-ket. Ebben az esetben a járműveknek viszonylag sokat kell utaznia az egyes állomásokra, ezért létrehoztam vegyes központi-elosztott irányítást is. A vegyes stratégiánál kettő csoportra bontottam a gyártási folyamatot, a szerelés részlegre (sárga úthálózat) és a megmunkáló részlegre (narancs, zöld, kék együtt), ezeket a részeket egy-egy AGV flotta látja el.

A munkadarabot kísérő irányítás esetén minden alapanyaghoz hozzá rendel egy szabad AGV-t, amely lépésről lépésre szállítja el a következő állomásokra. Amíg a munkadarab megmunkálás alatt van, addig a hozzárendelt jármű az adott állomás előtt megvárja, amíg végez a művelettel és után viszi tovább a következő helyre. Programozást tekintve lényegében annyiban különbözik a nem-kísérő módszertől, hogy a Mdb_átadás metódusban, mikor az AGV leszállít egy munkadarabot és átrakja az állomásra, akkor a jármű ott marad a szenzoron és nem lesz szabad; megvárja, míg az állomás végez a művelettel. Mikor elkészült a munkával, lefut az AGV_hívás metódus, melyben külön vizsgálom, hogy tartózkodik az állomás előtt jármű vagy sem.

AGV híváskor az állomásnak mindig meg kell várnia, míg a jármű oda nem ér, ez függ a jármű sebességétől és a tartózkodási helyétől. Az állomás AGV-re várakozási idejének a csökkentésére bevezettem egy előrelátó stratégiát, melynek lényege, hogy figyelembe veszi a jármű utazási idejét és ez alapján úgy hívja oda az állomásra, hogy a várakozási idő minimális legyen. A mintagyárban szereplő előrelátó modell az osztályozását tekintve munkadarabot nem kísérő elosztott stratégia csoportba sorolható. A szimulációs modellben ezt az AGV_hívás2 metódus végzi, ami az állomások bemeneti kontrollja. Az előrelátó stratégia esetén az állomásoknak nincs kimeneti kontrollja, a szállítójármű hívását csak a bemeneti kontroll végzi. Az AGV_hívás2 metódus részben megegyezik az eddig használt AGV_hívás metódussal, ugyanazokat a lépéseket hajtja végre az elején. A különbség annyi, hogy becslést végez a jármű várható utazási idejéről a sebességből és a távolságból. A metódus az állomás feldolgozási idejének vége előtt a szükséges utazási időnyivel korábban lefoglalja az AGV-t és elindítja azt az állomás felé (vagyis az állomás feldolgozási idejének és az AGV útidőnek különbségéből adódó idővel később foglalja le az adott járművet). Ekkor ideális esetben, ha

mindig van szabad szállítóeszköz az állomástól legfeljebb a feldolgozási időnyi távolságra, akkor az állomások járműre várakozási ideje 0-ra csökkenthető.

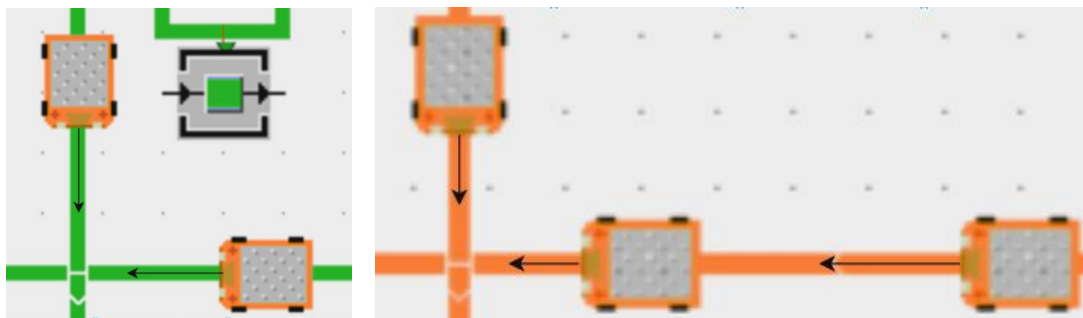
Észrevehető, hogy a Plant Simulation-ben ezeknek a modelleknek a megvalósítása programozástechnikai szempontból kissé eltérő a valós rendszerekétől, azonban működési elvüket nézve jól tükrözik a valós folyamatokat. A modell nem az irányítási programok tesztelésére készült, hanem annak működésének szimulálására.

4.5. Ütközések kezelése

A mintagyár modellben az AGV-k nyomvonalai úgy kerültek kialakításra, hogy a járművek közötti ütközések lehetőleg minél ritkábban forduljanak elő. Az úthálózat egyirányú útszakaszokból áll, ezért frontális ütközés nem fordulhat elő két AGV között. A munkadarabátadási helyekhez külön leálló sávok vannak kialakítva, hogy az ott várakozó járművek ne zavarják a forgalmat. A szimulációban minden AGV azonos sebességgel mozog, így normál körülmények között a járművek nem érik utol egymást.

Kettő esetben azonban előfordulhat ütközés a modellben (4-6. ábra)

1. Amikor több jármű egyszerre ér egy kereszteződéshez
2. Amikor egy jármű hátulról belefut az előtte levőbe, amely megállt egy kereszteződés előtt, hogy elengedjen egy másik irányból érkező AGV-t. A hátulról belemenés előfordulhatna több helyen is, ha a járművek eltérő sebességgel közlekednének.



4-6. ábra. AGV-k ütközésének esetei a modellben

4.5.1. ÜTKÖZÉS KERESZTEZŐDÉSSEN

Amikor egy kereszteződéshez egyszerre érkeznek két jármű, akkor valamilyen közlekedési szabály alapján döntenek, hogy melyik jármű haladhat át elsőként. Ilyen szabály lehet például a főút-mellékút vagy a jobbkézsabály.

A szimulációs modellben az utak egyirányúak és az AGV-k megadott sebességgel haladnak végig rajtuk az előírt irányban. Az útszakaszok ún. összekötőkkel kapcsolódnak egymáshoz, ezzel az egyes utaknak létrehozunk előd és utód útszakasz(oka)t. Az előd objektum az, ahonnan az AGV érkezik, az utód szakasz pedig az, amerre tovább halad. A kereszteződéseknél egy útnak több előd- és utódútja is lehet, melyeket sorszámok különböztetnek meg.

A mintagyár modellben amikor több AGV egyidőben érkezik egy kereszteződéshez és ütközésre kerülne sor, akkor elsőbbséget élvez az a jármű, amelyik a kisebb sorszámú elődágról érkezik. Amennyiben az AGV-k nem teljesen egyszerre érkeznek a kereszteződésbe, az érkezési sorrend dönt.

4.5.2. HÁTULRÓL BELEMENŐS ÜTKÖZÉS

A másik ütközési eset akkor fordul elő, amikor egy kereszteződés előtt a forgalom miatt álló AGV-be hátulról belefut egy másik jármű. A szállítóeszközöknek létrehozható egy ütközési kontroll, ezzel a módszerrel akadályozom meg, hogy teljesen egymásba menjenek. Az ütközéskezelő módszer megállítja a hátulról érkező járművet, vár egy másodpercet, és csak akkor indul újra, amikor az előtte lévő jármű elhagyja az adott útszakaszt.

A szimulációs modellbe (még) nem került beépítésre, de a Plant Simulation 2201 (v17) verziójától lehetőség van a szállítóeszközöknek távolság kontrollt is megadni, amivel meg lehetne akadályozni, hogy két jármű egy adott távolságnál közelebb kerüljön egymáshoz.

5. KÍSÉRLETEK ÉS EREDMÉNYEK

5.1. Teljesítménymutatók

A teljesítménymutatók (KPI-k) bevezetésével lehetőség nyílik az AGV flotta irányítási modelljeinek összehasonlítására és a különböző vezérlési stratégiák hatékonyságának értékelésére. A főbb mutatókat a modellben a „TeljesítményMutatók” nevű táblázatba gyűjtöm össze (5-1. ábra), amit a szimuláció lefutásának végén az EndSim metódus tölt ki. A táblázat automatikusan megjelenik a szimuláció végén, mutatva a beállított irányítási stratégiát és a különböző KPI-ket.

Gyártási folyamatok során az egyik legfontosabb teljesítménymutató a kihozatal, ami megegyezik az összes legyártott termékkel; ez a mintagyárban a szerelés művelet után a kimenetre szállított MU-k számával egyenlő. Másik fontos mutató az AGV-k száma, ami az Init metódus által létrehozott összes jármű számával egyezik meg. A teljesítmény szempontjából fontos, hogy minél nagyobb kihozatalt érjünk el, lehetőség szerint minél kevesebb AGV alkalmazásával.

	1	2
0	Teljesítmény mutató	Érték
1	AGV irányítási stratégia	
2	Kihozatal [db]	
3	AGV-k száma	
4	AGV-k által megtett össz.út [m]	
5	AGV kihasználtság [%]	
6	AGV össz.megtett útidő	
7	AGV össz.üres idő [time]	
8	AGV össz.szállítási idő [time]	
9	AGV össz.üres utazási idő [time]	
10	Állomás agv-re várakozási össz.idő [time]	
11	Legtöbb út [m]	

5-1. ábra. TeljesítményMutatók táblázat

A Plant Simulation szoftverben a szállítóeszköz objektumnak számos statisztikai attribútuma van, mint például az AGV által megtett távolság, vagy az üresen, munkadarab nélkül töltött idő. Azonban vannak mutatók, melyek saját számítás útján kerültek meghatározásra, mint például az AGV utazási vagy szállítási ideje (az utazási időn belül amikor van rajta munkadarab), illetve az üres utazási ideje (az utazási időn belül amikor nincs rajta munkadarab). A számításokat az EndSim metódus hajtja végre, ami a szimuláció végén fut le.

A táblázatban szereplő értékek többsége az összes AGV egy adott jellemzőjének összege. Az AGV-k által megtett össz. út az egyes járművek összesen megtett távolságának összege. A „legtöbb út” KPI a szimuláció során legtöbb távolságot megtévő jármű

útja. A távolságot figyelő mutatóknál a minél kisebb úthossz az előnyös, mivel ekkor lesz kisebb az AGV-k energiafogyasztása és amortizációja.

Az AGV-k kihasználtsága a járművek összes foglalt idejének és a szimuláció futási idejének hányadosa. A foglalt idő az az időmennyiség, amikor a szállító objektum szabad attribútuma hamis, ennek értékét az AGV szabad megfigyelő metódus folyamatosan számítja. A kihasználtság 0 és 1 közötti értékű és a minél nagyobb értékre kell törekedni.

A teljesítmény mutatók táblázatban az AGV össz. megtett útideje a járművek összes utazási idejével egyezik meg, ez az érték tovább van bontva más KPI-ként, szállítási és üres utazási időkre. Az AGV össz. üres idő tartalmazza az üres utazási időket, a jármű várakozási időket, valamint a parkolóhelyen eltöltött időket is.

5.2. AGV-k számának meghatározása

A szimuláció egyik fontos bemenő paramétere az AGV-k száma az egyes parkolóhelyeken. Ennek megadása két globális változó segítségével lehetséges, az AGVk_száma1 és az AGVk_száma2-vel. Központi irányítási stratégia esetén értelemszerűen csak egy változónak adunk értéket, viszont elosztott vagy vegyes stratégia esetén mindkettőt be kell állítani. Az AGVk_száma1 a megmunkáló részlegen lévő parkolókban lévő járművek számát határozza meg, központi vagy vegyes irányítás esetén csak az egyik parkolóba hoz létre AGV-ket (a középén található, mivel így érhető el, hogy a lehető legkevesebbet utazzanak az AGV-k). Elosztott stratégia esetén viszont az ott található mindhárom parkolóba a megadott számú szállítójárművet hozza létre. Az AGVk_száma2 paraméter a szerelés részlegen (sárga) működő járművek számát határozza meg, ehhez a részleghez csak egy darab parkoló tartozik.

A szükséges AGV-k számának meghatározását sok tényező befolyásolhatja, függhet többek között a termelékenység maximalizálásától, de számát korlátozhatja az energiahasználat, vagy a ráfordítható költségkeret.

5.2.1. KIHUZATAL SZERINT

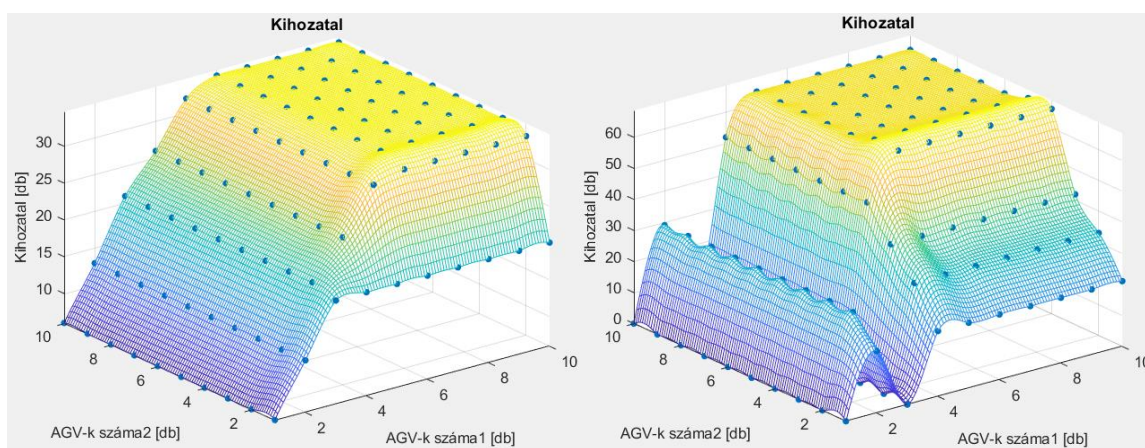
Gyártási folyamatoknál legtöbbször termelékenységet szeretnénk maximalizálni, így az AGV-k számának meghatározásánál fontos szempont, hogy mennyi jármű szükséges a legtöbb kihozatal eléréséhez.

A szimulációs modellben ennek meghatározásához kísérleteket végeztem a Plant Simulation-ben található „ExperimentManager” objektum segítségével. A kísérletek futtatásához definiálni kell a bemeneti és kimeneti változókat. A bemeneti paraméterek a modellben szereplő két, AGV-k számát megadó változó, a kimenet pedig a kihozatal.

A bemenő változókat 1-től indítja és 10-ig növeli, ezután minden kombinációra lefuttat egy szimulációt, így összesen 100 darab futást fog végrehajtani. A kísérletek végén a Plant Simulation automatikusan készít egy jelentést, megtalálva azokat a bemeneti kombinációkat, melyek esetén a legnagyobb a termelés.

A kísérletek eredményeit ezután ábrázoltam egy három-dimenziós grafikonon. Az ábrázoláshoz a kísérletek eredményeit a Plant Simulation lementette egy excel fájlba, majd azt beolvastam egy Matlab-ban megírt kódba és ott készítettem el a grafikonokat. A koordináta-rendszer x és y tengelyi az AGVk_száma1 és AGVk_száma2, a z tengely pedig a kihozatal.

Az alábbi, 4-2. ábrán látható két példa a kihozatal értékének szemléltetésére különböző irányítási modellek esetén. A baloldali grafikonon egy vegyes központi-elosztott irányítás eredményei láthatóak, a jobboldalin pedig az előrelátó irányítási stratégia eredményei. A pontok jelölik a ténylegesen felvett értékeket, a háló csak a könnyebb láthatóság miatt van megjelenítve.



5-2. ábra. Kihozatal

Észrevehető, hogy egy adott szám felett már nem nő tovább a kihozatal, a grafikonon egy x-y síkkal párhuzamos felület fogja jelölni, ahol maximális a kihozatal. Megfigyelhető, hogy az előrelátó modell esetén (jobb oldali) a kihozatal sokkal nagyobb, mint más modellek esetén, azonban az AGV-k számának növelésével nem nő végig monoton a kihozatal, található egy lokális minimum, ahol a kimenete 0. Ennek oka egy holtpont bekövetkezése, amikor minden szállítójármű egy állomás előtt vár, viszont nem tudják tovább vinni a munkadarabot, mert a következő állomás nem szabad. Ezeket a holtpontokat a modell nem kezeli (még).

5.2.2. TÖBB KIMENETEI VÁLTOZÓ FIGYELEMBEVÉTELE

A maximális kihozatal eléréséhez meghatározható a szükséges AGV-k száma, azonban több bemenő paraméter kombinációval is elérhető a legnagyobb kihozatal, ez látható a feljebbi 5-2.-es ábrán is. További keresési feltétellel lehet szűkíteni a keresést: például a maximális kihozatal, de a minimális számú járművel.

Az ideális mennyiség megtalálásához célszerű több kimeneti változót is figyelembe venni és így meghatározni az optimális számú AGV-t. Ehhez a többcélú optimalizálási feladathoz bevezetünk egy függvényt, ami a különböző kimeneti változók súlyozott összegét adja. Ezt a súlyfüggvényt fitness függvénynek fogom hívni a továbbiakban, mivel később ez lesz a genetikusan algoritmusnál a fitness függvény.

A fitness függvényben szerepel a kihozatal, az összes jármű száma, a kihasználtság, a megtett út összesen, illetve a legtöbbet utazó AGV által megtett távolság. A fitness függvény maximumának megtalálásával megkapható az optimális mennyiségű szállítójármű. Mindegyik tag egy beállított előjeles súllyal szerepel: pozitívvál, ha a nagyobb értékek a jók (pl. kihozatal); negatívvál, ha minél kisebb értékek a jók (pl. összes jármű száma). A fitness függvény alakja:

$$f(x) = \sum_{i=1}^n w_i \cdot g_i(x) \quad (5.1)$$

, ahol: f a fitness fv., w_i a súly, g_i -k pedig egy-egy célfüggvények (például a kihozatal), n pedig a célfüggvények száma. A függvény x paramétere pedig az AGV_k_száma1 és AGV_k_száma2, vagyis vektormennyiség. Egyváltozós optimalizálás során az i változó 1, ilyen eset volt az előző, 4.2.1.-es fejezetben, ahol a g célfüggvény a kihozatal volt, a súly pedig 1.

A g_i függvények alaphelyzetben nem összemérhetőek egymással, mert eltérő dimenziójúak, például lehet méter, idő vagy darab. Emiatt az értékeket először normalizálni kell. Többféleképpen is lehet skálázni, én az alábbi, min-max normalizálás képlettel számoltam:

függvény alakja:

$$x_{j,norm} = \frac{x_j - \min(A)}{\max(A) - \min(A)} \quad (5.2)$$

, ahol x_j az aktuális függvény érték, A pedig a felvett függvényértékek halmaza. A normalizálás után a célfüggvények értékei 0 és 1 közötti értékeket vesznek fel.

A fitness függvényben a súlyok megadása nagy szabadságot adhat, viszont a kimenet erősen fog függeni tőlük. A súlyok megválasztásánál több szempontot is figyelembe vettem:

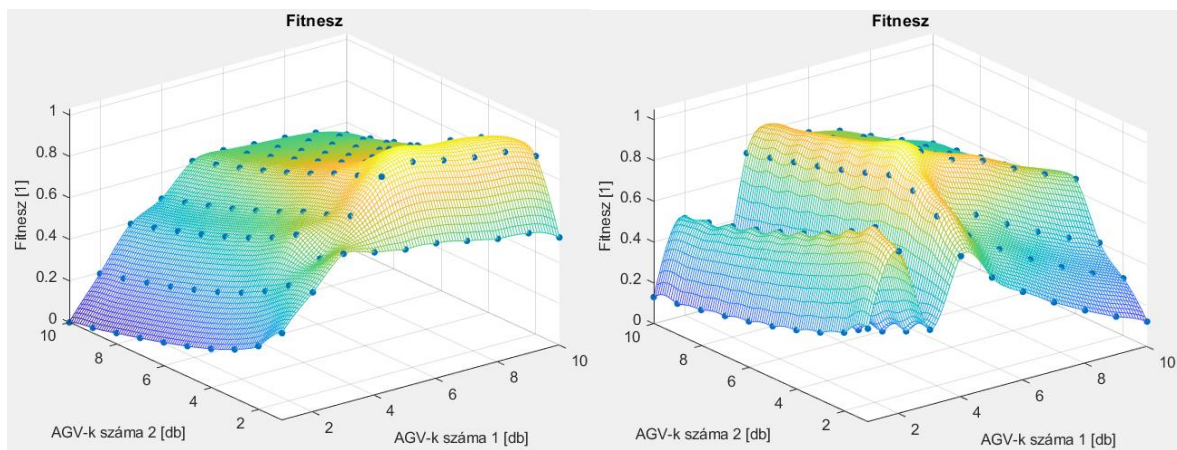
1. A kihozatal minél nagyobb legyen → ennek súlya legyen a legnagyobb és pozitív előjelű
2. A kihozatal után az AGV-k száma a legfontosabb (hogy a beruházási és a fenntartási költségek alacsonyak legyenek), ezért ennek súlya legyen a második legnagyobb. Mivel az AGV-k számának minél kisebbnek kell lennie, ezért a súlya negatív szám lesz.
3. A többi célfüggvények kisebb súllyal kerülnek beszámításra
4. A kihasználtság minél nagyobb legyen → pozitív súly
5. Az összes megtett út minél kisebb legyen, ezzel csökkentve az AGV-k energiafogyasztását → negatív súly
6. A legtöbbet utat megtevő jármű távolsága minél kisebb legyen → negatív súly

Az alábbi táblázat mutatja az egyes célfüggvényekhez megválasztott súlyokat:

5-1. táblázat. A súlyok értékei

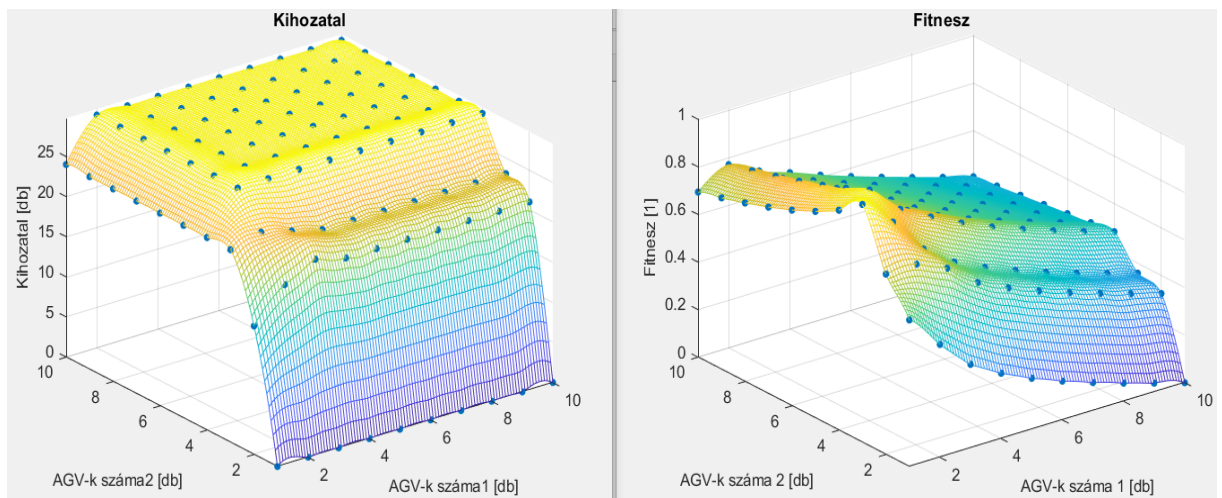
g_i	w_i
kihozatal	1
összes AGV száma	- 0,5
kihasználtság	0,1
összesen megtett út	-0,1
legtöbb út	-0,1

A fitness függvény maximumának megtalálására kísérleteket futtattam le a modellen a Plant Simulation „ExperimentManager” objektum használatával. A bemeneti paraméterek a modellben szereplő AGVk_száma1 és AGVk_száma2, ezeket a változókat 1-től indítja és 10-ig növeli a futások során. A kísérletek során megkapjuk a fitness értékeit a bemeneti kombinációk függvényében, a legjobb fitnessértékhez tartozó paraméterek határozzák meg az optimális számú AGV-t az üzem első, illetve a második részlegén. Az alábbi 5-3.-es ábrán a korábban (5-2.-es ábra) ábrázolt kihozatal függvényekhez tartozó irányítási modellek fitness függvényei láthatóak.



5-3. ábra. Fitnessz függvény

A legtöbb stratégiánál az optimális számú AGV-vel maximális kihozatal érhető el a lehető legkevesebb számú járművel. Azonban ez nem minden esetben teljesül, például mikor a kihozatal nem nő már szignifikánsan az AGVk-számának növelésével. Az alábbi grafikonon (5-4. ábra) erre az esetre látható egy példa; az ábra egy elosztott, munkadarabot nem kísérő irányítási modell kihozatali és fitnessz függvényét mutatja.



5-4. ábra. Kihozatal és Fitnessz függvény

5.2.3. OPTIMALIZÁLÁS GENETIKUS ALGORITMUSSAL

Az előző, 5.2.2. fejezetben az optimum megtalálásához egy fitness függvényt vezettünk be és ennek maximumát kerestük úgy, hogy a kísérlettervező objektum az összes lehetséges bemeneti kombinációra lefuttatott egy szimulációt és kiszámította a fitness fv. értékét. Ez a módszer azonban időigényes tud lenni, ha a bemeneti vektor kombinációinak száma nagyobb. Ilyen esetekre jó megoldás lehet a genetikusan algoritmus használata.

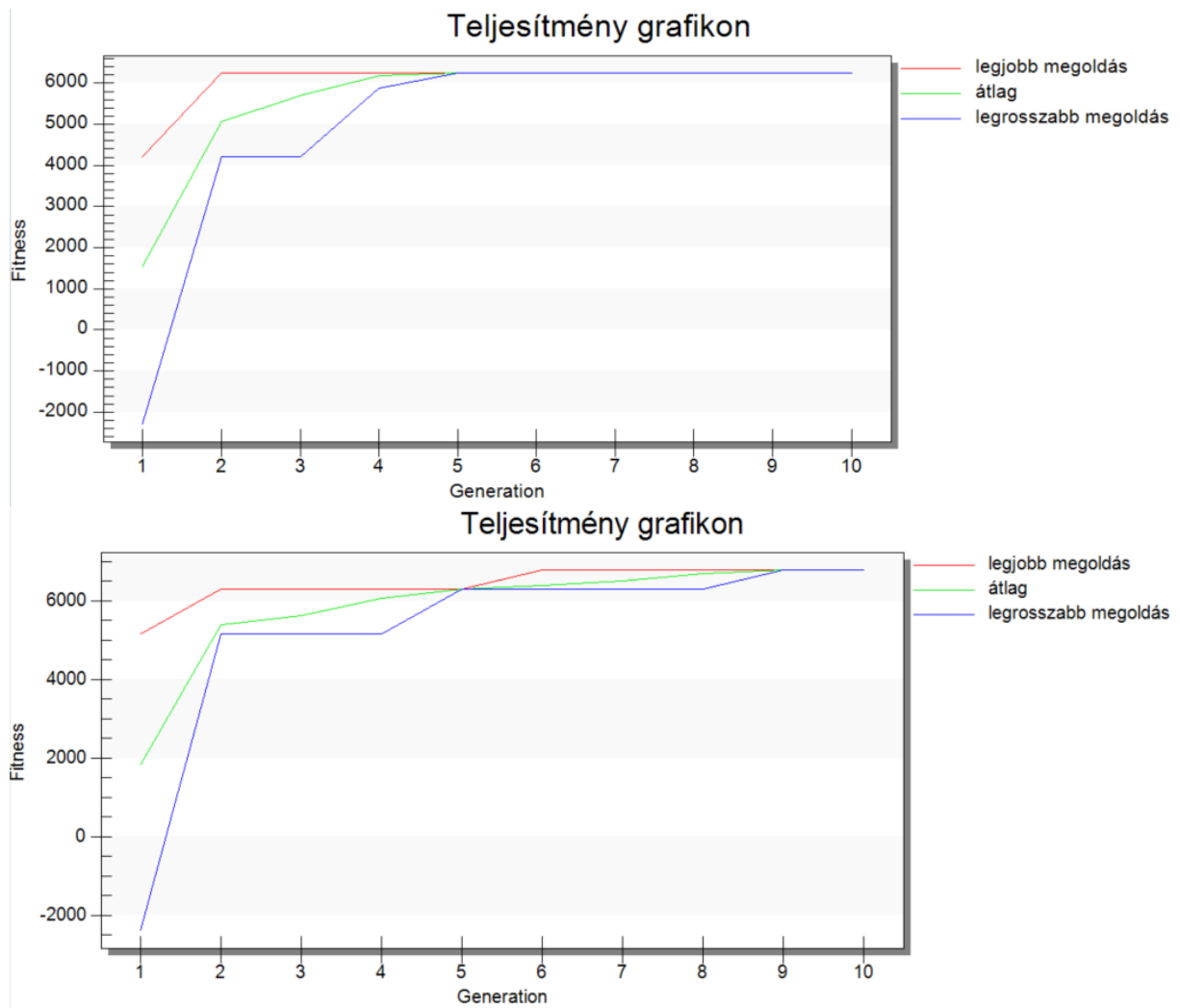
A Plant Simulation szoftverben található a GAWizard nevű objektum, ami a genetikusan algoritmus használatát segíti és nagyban leegyszerűsíti. A fitness függvény számítása megegyezik a már korábban bemutatott módon (5.2.2. fejezet). Az optimalizációs paraméterek az AGVk_száma1 és 2.

A genetikusan algoritmusokat kétféle képpen szokás leállítani, vagy ha már tartósan nem javul a fitness érték, vagy egy adott generáció számig. A GAWizard-ban az utóbbit lehet megadni, ennek értékét 10-re állítottam. A másik fontos paramétere az algoritmusnak a populáció mérete, ennek értékét 5-re állítottam.

A most vizsgált optimalizálási feladatnál összesen 100 bemeneti kombináció volt, ezek mindegyikére lefuttatva egy szimulációt még nem túl hosszú idő. Azonban a kombinációk száma jelentősen meg tud ugrani például egy új paraméter vizsgálatával, és ezzel együtt a kísérletek futási ideje is megnő.

Általánosságban a genetikusan algoritmus nem feltétlenül találja meg a fitness függvény maximumát, vagyis a tényleges optimumot, azonban szimulációs modellben a beállított paraméterek mellett minden esetben sikerült megtalálnia. A legjobb egyed fitness értéke megegyezik a kísérletek során meghatározott optimummal.

Az evolúciós folyamat ábrázolásával szemléltethető, hogy milyen gyorsan találja meg az algoritmus az optimumot. A különböző irányítási modelleknél eltérő sebességgel érte el a maximális fitness értéket, ez látható az alábbi grafikonokon is, az első esetben már az 5. generációban eléri a legrosszabb megoldás is (a legjobb megoldás már a 2. generációra), viszont a második esetben csak a 9.-re (a legjobb megoldás pedig a 6. generációra). Az kijelenthető, hogy 10 generáció alatt minden esetben megtalálta az optimumot a genetikusan algoritmus.



5-5. ábra. Evolúció

5.3. AGV irányítási modellek összehasonlítása

Az AGV irányítási stratégiák összehasonlítása többféle szempont szerint is lehetséges, többek között ezek vizsgálatára kerültek bevezetésre a különböző KPI-k. Az irányítási modelleket a Mintagyár termelési folyamatán teszteltem, az itt gyűjtött információk és teljesítménymutatók alapján fogom most röviden jellemezni az egyes stratégiákat.

Központi irányítási modell esetén egy parkolóhely van csak használatban az egész üzemben, onnan a gyár bármely pontjára el lehet küldeni az AGV-ket. Nagy előnye, hogy bár több jármű tartozik egy parkolóhoz, mint az elosztott vezérlés esetén, de összességében kevesebb járművel meg lehet valósítani a gyártási folyamatot a mintagyár modellben. Hátránya, hogy az állomásoknak többet kell várakoznia a járműre és hogy

az AGV-knek többet kell utaznia, ezért több lesz az egyes járművek energiafogyasztása és amortizációja.

Elosztott irányítás esetén több szállítóeszköz szükséges a termelés ellátásához, viszont az AGV-k kevesebb utat tesznek meg és hamarabb odaérnek a hívóállomáshoz, mivel mindig csak a saját csoportjukban mozognak és nem kerülnek messze az adott állomástól. Ugyanígy a parkolóra is hamarabb visszaérnek, ahol a valós gyárakban általában töltőre is tudnak kapcsolódni. További előnye, hogy az egész gyártási és szállítási folyamatok könnyebben átláthatóak.

Vannak vegyes központi-elosztott stratégiák is, melyek ötvözik a kettő előnyeit és hátrányait.

A munkadarabot kísérő modell esetén a legkisebb az AGV-k által megtett út, mivel nem tér vissza egyfolytában a parkolóhelyre. Emiatt ugyanakkor a töltés lehetősége sincs meg, viszont összességében jóval kevesebb a fogyasztása is. Másik előnye, hogy magasabb kihozatal érhető el hasonló számú járművel, mint másik stratégia esetén. A kihasználtsága is magasabb a szállítóeszközöknek, mivel az AGV végig foglalt állapotban van, amíg a hozzárendelt munkadarab el nem hagyja az adott gyártórészleget. Az eredmények alapján a munkadarabot kísérő modell nem használ több AGV-t, azonban ez más gyárelrendezésekre már nem biztos, hogy igaz lenne. Ennek vizsgálatához érdemes lehet a jövőben más fajta mintagyárakat létrehozni.

Előrelátó stratégia esetén a legmagasabb a kihozatal, akár kétszerese is lehet, mint a többi modell esetén. Az állomások szállítóeszközre váró ideje minimális és az AGV-k kihasználtsága viszonylag magas. Hátránya, hogy a maximális kihozatal eléréséhez több jármű szükséges, mint más stratégiáknál, valamint sokkal összetettebb és bonyolultabb az irányítás, mint más modellek esetén.

5-2. táblázat. Irányítási modellek előnyei – hátrányai

Irányítási stratégia	központi	elosztott	mdb-t kísérő	mdb-t nem kísérő	előrelátó
előnyök	kevesebb AGV szükséges	kevesebb utat kell megtenni	legkevesebb út, magas kihasználtság	gyakrabban lehet tölteni az AGV-ket, egyszerűbb irányítás	maximális kihozatal
hátrányok	több utat kell megtenni	több AGV	kevesebb töltési lehetőség	több utat kell megtenni	legtöbb AGV szükséges, bonyolult irányítás

5.4. Legjobb szimulációs paraméterek

Az AGV-k optimális mennyisége már meghatározásra került az egyes stratégiákra. Azonban meg lehet keresni az összes szimulációs paraméter ideális beállítási variációját. A modellben három bemenő változó van, az AGV irányítási stratégia típusa, az AGVk_száma1 és az AGVk_száma2 paraméter.

Három változó esetén már túl sok bemeneti kombináció lehetséges, hogy az összesre lefusson egy szimuláció. Helyette érdemes genetikus algoritmussal megtalálni az optimumot, tekintettel arra, hogy a modell továbbfejlesztésével akár további bemeneti változó is bevezetésre kerülhet és még több kombináció állhat elő.

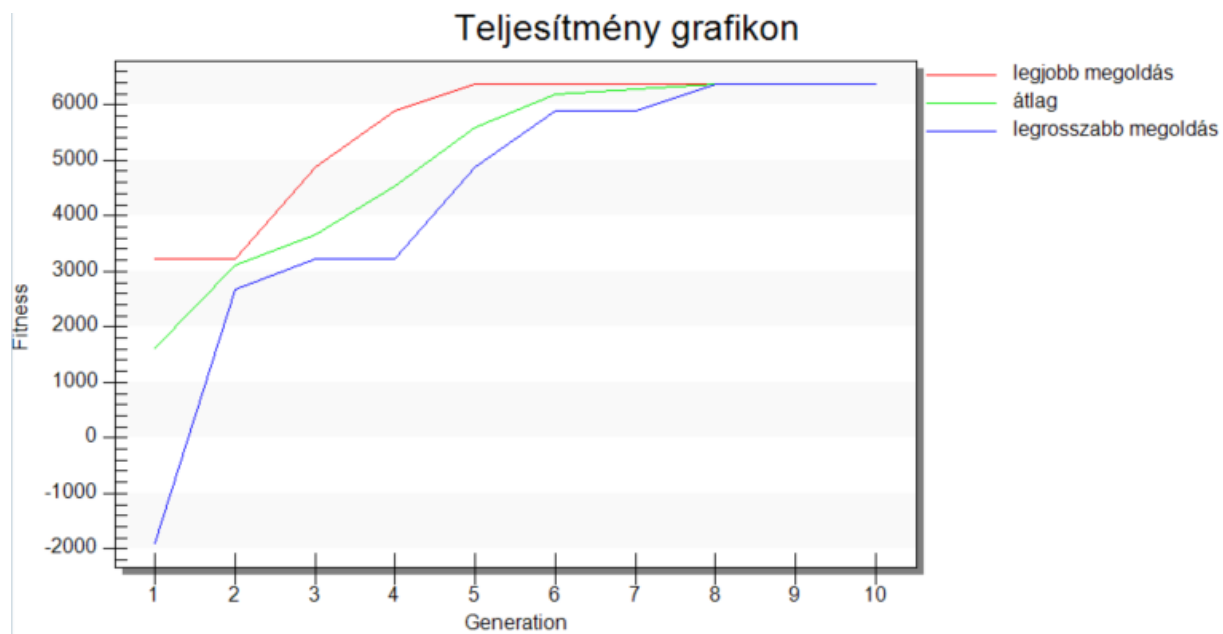
A fitness függvény a korábbi fejezetben definiált értékkel fog megegyezni, annyi eltéréssel, hogy a normalizálásnál az összes célfüggvény értékészletét kell figyelni, nem csak egy adott stratégiánál; így kell meghatározni a célfüggvények abszolút minimum- és maximumértékét. Ezzel a fitness függvény értéke összevethető az egyes irányítási stratégiák között.

A genetikus algoritmusban a korábbi optimalizálási feladatnál beállított paraméterek szerepelnek, kivéve az optimalizációs paraméterek, mert az kibővül az AGV stratégia típusával. Az algoritmus lefutásával megtalálható a fitness függvény maximuma, ami meghatározza az optimális szimulációs paraméter beállítást.

A genetikus algoritmussal kapott eredmény az alábbi 5-6. ábrán látható. A 6-os AGV stratégia az előrelátó irányítási modellt jelöli, a kapott AGV-k mennyisége pedig a legtöbb kihozatalt elérő kombinációk mellett a minimális számú. A 5-7. ábrán látható az evolúciós folyamat, a 8. generáció után már a legrosszabb megoldás esetén sem nő tovább a fitness függvény értéke.

table 0	real 1	integer 2	table 3	table 4		string 0	string 1	
string	Egyed	Fitness	ID	Kromoszómák ->	Próbálkozások	string	Készlet definiálás	Kiosztás
1	Gen 5 Ind 5	6384	40	Chrom 1	Fitness 40	1	root.AGVk_száma1	5
2	Gen 8 Ind 4	6163	69	Chrom 1	Fitness 69	2	root.AGVk_száma2	4
3	Gen 4 Ind 1	5888	26	Chrom 1	Fitness 26	3	root.AGV_stratégia	6
4	Gen 7 Ind 5	5471	60	Chrom 1	Fitness 60	4		
5	Gen 7 Ind 9	5017	64	Chrom 1	Fitness 64	5		
6						6		

5-6. ábra. Legjobb egyedek



5-7. ábra. Evolúció

6. ÖSSZEFOGLALÁS

6.1. Továbbfejlesztési lehetőségek

A szimulációs modell és az irányítási stratégiák továbbfejlesztésére számos lehetőség kínálkozik.

Jelenleg az AGV stratégiák egyetlen mintagyáron működnek, azonban az irányítási módszerek szélesebb körű tesztelésére érdemes lenne különböző gyármodelleket létrehozni, vagy valós termelési rendszerek digitális mását alkalmazni. Figyelembe lehetne venni a járművek és állomások meghibásodását is, amely közelebb hozná a modellt a valós működési környezethez.

A modell úthálózatának és parkolóhelyeinek kialakítása szintén továbbfejleszthető, hogy hatékonyabb járműmozgást tegyen lehetővé, ezzel gyorsítva az anyagáramlás sebességét és csökkentve a várakozási időket. Valós környezetben az AGV-k folyamatosan energiát fogyasztanak, és amikor parkolóhelyen tartózkodnak, töltés alatt állnak. Az energiahasználat figyelembevételével egy komplexebb modell alkotható, amelyben a töltési ciklusok összehangolhatók a gyártási folyamatokkal, így elkerülhetők a váratlan leállások.

Az AGV irányítási stratégiákat is tovább lehet javítani, például a holtpontok elkerülésének figyelése. Ezt talán legegyszerűbben az egyes feladatok prioritizálásával lehetne megoldani. A meglévő stratégiákon túl újakat is be lehetne vezetni a modellbe, például a tandem konfiguráció, vagy a folyamatosan mozgó AGV-ket.

Az optimalizálás során a modell bonyolultabbá válásával új bemenő paraméterek is megjelenhetnek, amelyekhez szintén meg lehet keresni a fitnessfüggvény maximumát. További kutatásokkal lehetőség nyílik új optimumkeresési módszerek tesztelésére, illetve saját fejlesztésű algoritmusok létrehozására, amelyekkel tovább lehetne javítani az optimalizálás pontosságát és hatékonyságát.

6.2. Összefoglalás

A dolgozat célja egy AGV flotta irányítási stratégiáinak kidolgozása és ezek megvalósítása volt szimulációs környezetben.

A dolgozat elején bemutatam az AGV/AMR rendszereket, valamint az AGV flották irányítására rendelkezésre álló vezérlési módszereket. Ezt követően a szimulációk téren végeztem kutatást, ismertettem azok csoportosítását, előnyeit, majd az iparban az egyik legnépszerűbb szoftvert, a Siemens Tecnomatix Plant Simulation-t mutattam be, kitérve a dolgozatban később használt funkcióira.

A flotta irányításának vizsgálatához egy mintagyár modellt készítettem a Plant Simulation szoftverben. Röviden bemutatam a modellt és annak működését, majd különböző AGV irányítási stratégiákat dolgoztam ki a mintagyárban az anyagszállítási feladatokra. A szimulációk futása és a teljesítménymutatók bevezetése lehetőséget nyújtott a vezérlési modellek összehasonlítására.

Meghatároztam a maximális kihozatal eléréséhez szükséges járművek számát, majd az optimális számú AGV-t két féle módszerrel kerestem meg: először kísérletek futtatásával, majd genetikus algoritmussal. Ezt követően genetikus algoritmus segítségével megkerestem az optimális szimulációs paraméterbeállítási variációt. Végül javaslatokat tettem a modell továbbfejlesztésére.

A dolgozat eredményei hozzájárulnak az AGV rendszerek tervezéséhez és irányításához, valamint a genetikus algoritmusok alkalmazásával a többcélú optimalizálás során megtalálhatók az ideális gyártási paraméterek. A szimuláció segítségével összehasonlíthatók az egyes irányítási stratégiák, növelhető a termelés hatékonysága, és csökkenthetők a költségek. A dolgozat támogatja a modern gyártási folyamatok, a digitális gyártás és az Ipar4.0 irányelveinek gyakorlati megvalósítását, valamint ezek előnyeinek szemléltetését és kihasználását.

7. FELHASZNÁLT FORRÁSOK

1. KULCSÁR BÉLA (1998): *Ipari logisztika*. LSI Oktatóközpont, Budapest, ISBN 963 577 242 4
2. REZA A. MALEKI: (1991): *Flexible Manufacturing Systems: The Technology and Management*. Prentice-Hall, Englewood Cliffs, ISBN 0-13-321761-2
3. GIUSEPPE FRAGAPANE, RENÉ DE KOSTER, FABIO SGARBOSSA, JAN OLA STRAND-HAGEN (2021): *Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda*. Elsevier.
4. RONALD J MANTEL, HENRI R.A. LANDEWEERD (1995): *Design and operational control of an AGV systems*. Elsevier.
5. IVICA DRAGANJAC, DAMJAN MIKLIČ, ZDENKO KOVAČIĆ, GORAN VASILJEVIĆ, STJEPAN BOGDAN (2016): *Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications*. IEEE Transactions on Automation Science and engineering, Vol. 13/No. 4, October 2016
6. KUN ZHENG, DUNBING TANG, WENBIN GU, MIN DAI (2013): *Distributed control of multi-AGV system based on regional control model*. German Academic Society for Production Engineering (WGP) 2013.
7. José A. Ventura, Chulung Lee (2001): *A Study of the Tandem Loop with Multiple Vehicles Configuration for Automated Guided Vehicle Systems*. Journal of Manufacturing Systems, Vol. 20/No.3, 2001
8. YAVUZ A. BOZER, MANDYAM M. SRINIVASAN (1992): *Tandem AGV systems: A partitioning algorithm and performance comparison with conventional AGV systems*. European Journal of Operational Research 63 (1992) 173-191
9. IRIS R.A. VIS (2006): *Survey of research in the design and control of automated guided vehicle systems*. European Journal of Operational Research 170 (2006) 677-709
10. VDI (VEREIN DEUTSCHER INGENIEURE) (2018): *VDI 3633, Blatt 8, Simulation von Logistik-, Materialfluss- und Produktionssystemen - Begriffe*. VDI-Gesellschaft Produktion und Logistik
11. JERRY BANKS (1999): *Introduction to simulation*. AutoSimulations, Atlanta
12. RICKI G. INGALLS (2011): *Introduction to Simulation*. 2011 Winter Simulation Conference, IEEE, Phoenix, AZ, USA, 11-14 December, 2011
13. W. KRITZINGER, M. KARNER, G. TRAAR, J. HENJES, W. SIHN (2018): *Digital Twin in manufacturing: A categorical literature review and classification*, IFAC Paper-OnLine
14. AVERILL M. LAW (2015): *Simulation Modelling and Analysis 5th Edition*. McGraw-Hill, New York

15. STEFFEN BANGSOW (2015): Tecnomatix Plant Simulation - Modeling and Programming by Means of Examples. Springer, Switzerland
16. Tecnomatix Plant Simulation Help, <https://docs.plm.automation.siemens.com/>, 2024
17. <https://digitalisgyar.com/2022/06/26/agv-kozelseg-szabalyozas-plant-simulation-gyakorlati-pelda/>
18. Csuta Ákos, Dobák Dávis (2023): *Genetikus algoritmus logisztikai alkalmazhatósága*. Mesterséges Intelligencia – interdiszciplináris folyóirat, V. évf. 2023/1. szám. 21-34. doi: 10.35406/MI.2023.1.21
19. <https://www.kendrion.com/en/industries/automated-guided-vehicles-agv>