

Válasz Dr. Jacsó Ádám opponensi bírálatára

Tisztelt Jacsó Ádám!

Hálásan köszönöm a Bíráló alapos és részletekbe menő munkáját, lényeglátását, és számos felvetését, amelyek nemcsak a dolgozat jobbá tételét eredményezik, hanem a téma további kutatása során is inspirálni fognak.

A felvetett észrevételek jogosságát a legtöbb esetben nem vitatom. Ezek megszívlelésével a kutatási téma színvonalasabb folytatása válik lehetővé a későbbiekben.

A 2. pontban felvetett észrevételekre az alábbiakban röviden reagálok.

- A 13. oldalon csak két pont van kiemelve, ám a gyárszerkezet (architektúra) szervesen összefügg a folyamattal.
- A legyártott darabszámok összevetése reálisabb képet adhat a valós gyártásról, viszont a többi összefüggés is „idő” dimenzióban van megadva, ezért adtam meg a (7) képletben is „időben”. Jellemzően a ciklusidő a berendezések ütemidejével függ össze, ami egy automata gép esetében állandó érték.
- A (15) és (16) képletek arra utalnak, hogy a jelenleg használt mérőszámok mellett szükséges újabbak bevezetése is, ahogyan ezt a hivatkozott cikk is teszi.
- Az ábránál törekedtem a jó olvashatóságra, ám voltak olyan esetek amikor az ábra mérete és elrendezése adott volt, ahogy a 15. ábra esetében is, viszont ez fejezi ki a Petri-háló analizisét, ezért tartottam fontosnak elhelyezni a dolgozatban. Mellékletként elfordítva berakni jobb lett volna.
- A robot sebesség- és gyorsulásképeit a modellkörnyezetben nem kívántam figyelembe venni, ugyanakkor lehetséges egy arányossági számmal ezt kifejezni, indulásnál és megállásnál. Akár oly módon, hogy az útvonal hosszától függetlenül egy konstans értéket adok hozzá.
- A termelékenység maximalizálása valóban a legfontosabb célfüggvény lehet. Ezt úgy is el lehet érni, hogy az egyes elemek kihasználtságát maximalizáljuk, minden egyes elemre. Ezzel az egész rendszer kihasználtságát maximalizáltuk. Ugyanakkor ezek az elemek egymásra is hatással vannak, és más szempontokat is figyelembe kell venni, ezért komplex szimulációs eszközökkel lehet az egész rendszerre is maximalizálni a termelékenységet.
- Az egyezményes terminológia használtára törekedtem, ugyanakkor sajnos maradtak a dolgozatban nem konzekvens részek is, mint a „python” esetében.
- Az A és B pontok távolsága és a B és A pontok távolsága valóban különböző útvonalat jelöl, érdemes lett volna ezt a szövegben is megemlítenem. A sebességfüggvény egyszerűsített, a mozgás folyamán végig egyenletes sebességgel számol.
- A 21. ábrába túl sok információt akartam belesűriteni. A nagy akkumulátor kapacitás hosszabb töltési ideje miatt teljesítenek ezek rosszabbul az alacsonyabb csomagszámoknál, ugyanakkor a szimuláció nem töltéssel indul, arra csak az első szükséges időpontban kerül sor.

- A szimulációt így építettem fel, mert így tűnt logikusnak és megvalósíthatónak. Más megközelítést alkalmazva valószínűleg más következtetéseket vonhattam volna le. A kutatás további szakaszában érdemes lesz egy ilyen modellt is felépíteni.
- Mindenképpen érdemes megfontolni a rövidebb töltési ciklusok alkalmazását. Jelenleg ezt a kisebb akkumulátor kapacitásokkal vettem figyelembe, és minden esetben teljes töltést alkalmaztam, ami arányos a kapacitás értékével.
- A 77. oldalon az optimalizációs feladat specifikációit elnagyoltam, minden esetben volt minimális és maximális távolság is.
- Az RS-A távolság valóban nem szerepel a CD termék célfüggvényében, a leírt mondat csak példa volt.
- A 24. ábrán is túl sok információt akartam közölni. Így viszont a grafikonok egymással jól összehasonlíthatók.
- A 78. oldalon a „kiszámoltatás” és „optimalizálás” kifejezésekkel arra utaltam, hogy az én módszerem alapján számolom a célfüggvényt, majd a genetikus algoritmus ezek alapján elvégzi az optimalizációt.
- A genetikus algoritmus finomhangolása szükséges lett volna, így csak az alapbeállításokat használtam fel.
- A 79. oldal alján a „nagyobb populáció méretet és kisebb generáció számot választva” lenne a helyes.
- 81. oldal: magát a genetikus algoritmust számos alkalommal futtattam le, minimálisan 30-szor, és ezekből vontam le azt a következtetést, hogy kevesebb mint 10 generáció alatt jutott el az optimális megoldáshoz az én kiértékelő függvényemet felhasználva.
- 81. oldal: az egy darabra eső szimulációs idő szórását feltételezésem szerint a program elején és végén található program részek okozzák.
- 89. oldal: Amennyiben „halmazokban” gondolkodunk úgy egyszerűsíti a kiválasztást, és ezen egyszerűsítés keretén belül ad egy optimális megoldást.
- 91. oldal: a komplexitás kérdése valóban kritikussá teszi a teljes igazság táblázatos kiértékelést. Ugyanakkor az elmélet helyességét, hogy felírható a kiválasztás egy tudásbázissal, sikerrel igazoltam. A komplexitás csökkentésére pedig a szabály alapú következtetésre való áttérés adott megfelelő válaszokat, vagyis gyorsította fel a kiértékelést.
- 23. táblázat: a mértékegységek:
 - a terhelhetőség kg-ban van,
 - az akkumulátor kapacitás percben,
 - a töltési idő szintén percben,
 - a megbízhatóság %-ban,
 - a teljesítmény szintén %-ban.
 - A 22. táblázatban szerepelnek a mértékegységek, azért nem tüntettem fel a 23. táblázatban.

A 3. pontban a tézisek értékelését készítette el a Bíráló, melyet ezúton is köszönök. A leírt hiányosságok, megjegyzések hozzájárulnak a további munka kérdéses és pontosítandó területeinek meghatározásához is.

Tézisenként haladva:

Az 1. tézishez: A „milyen útvonal” és a „mekkora távolság” hordozhat plusz információt is amennyiben az útvonalon egy kikerülendő akadály helyezkedik el. A felvétel és lerakás pedig kellőképpen általános megfogalmazások pont azért, hogy a módszer általánosan is érvényes legyen, akár targonca szerű raklapszállító, akár futószalagos pufferből felvett dobozokat szállító robotokra egyaránt.

A 2. tézishez: Jelen esetben a „valós gyártási folyamatok” arra utal, hogy a számoláshoz a korábban ismertetett modellt használja, mi végig számolja az egyes útvonalakhoz tartozó szállítási időket.

A 3. tézishez: További vizsgálatok szükségesek, amivel a módszert is lehet majd pontosítani, így a „több tucat jellemző” is elérhetővé válik. Jelen esetben talán a „több, mint egy tucat jellemző” helyesebb lett volna.

A 4. pontban feltett kérdésekre a válaszaimat az alábbiakban adom meg:

Hogyan vehető figyelembe a kifejlesztett Petri-hálós gyártórendszer modellnél a több robot mozgása közötti interferencia, valamint a meghibásodások és karbantartások miatti leállások hatása?

A Petri-hálós modellben több robot egyidejű mozgása jelenleg nem vehető figyelembe, mert minden állapot „robot”-ként van hivatkozva. A folyamat szerint a gépekből van kettő, a robotból azonban csak egy. A modell kifejezetten az egyszerűsített megoldáshoz készült, viszont az jogos felvetés, hogy a valós problémák gyakran több robottal oldhatóak meg az elvárt követelmények kielégíthetősége miatt. A modell kiterjeszthetőségét több irányban is el tudnám képzelni. Az egyik szerint több ilyen „egyrobotos” modell lenne összefűzve, mintegy „szinteket” megadva. Így a gépek lehetnének egy bázis szinten, míg a robotok több szinten.

A megoldandó feladat így kettős lenne, először a gépek interfészeinél kerüljék el egymást a robotok, másodszor az útvonalon kell kezelni a találkozást. A gépeknél kialakított puffer az alapanyagok és termékek ütközésének elkerülése mellett a gépek folyamatos működését is lehetővé teszik, így ennek mintájára kialakítható lenne egy „robot-puffer” rész, amelyben az egyik robot pakolási feladatot végez, míg a másik beérkező robot egy kialakított helyen várakozik. Az útvonalon történő találkozás esetét ennél bonyolultabb lenne kezelni, az egyik ötletem szerint minden robotnak van egy „prioritási száma”, ami alapján a találkozás esetén eldönti, hogy melyik robot áll félre a másik útjából. Ezt a prioritást lehetne a termékhez is rendelni, ám ebben az esetben előfordulhat két azonos prioritás találkozási is, ami egy „szekunder prioritás” bevezetését is indokoltá tehetné. Egyszerűbb megoldásnak tűnik, hogy a robotok „kő-papír-olló” alapján döntsék el kinek van elsőbbsége. Esetleg a termékek készültségi fokát is figyelembe lehetne venni. Ez a terület további kutatást igényel, amire reményeim szerint a fejlesztés alatt álló szimulációs rendszer alkalmas is lesz, és további publikációkat is szeretnék írni ezen a területen.

Milyen megoldásokat tudna javasolni a kifejlesztett szakértői rendszer számítási komplexitásának csökkentésére?

A bemutatott szakértői rendszer első kiértékelő változata (19. ábra) egy teljes igazságtáblázatot használt, ami számítási komplexitás szempontjából rendkívül

erőforrás igényes, mert minden egyes újabb feltétel megduplázza a táblázat méretét, és így a számításához szükséges időt. Így a futási idő igénye exponenciálisan növekszik. Ezen úgy sikerült javítanom, hogy a kiértékelést szabályok végzik. Míg az első verzióban használt Python könyvtárral a felvetés logikai helyessége és működőképessége tesztelhető, addig a CLIPS környezetben a valós méretű problémák tesztelése is eredményesen futtatható reális számítási idő alatt. Egy-egy szabály az összes tényre lefut, így a számítási kapacitás igénye arányos a szabályok száma szorozva a tények számával, így a növekedés lineáris. A tesztek eredménye alapján a kiértékelés másodpercek alatt lefut, tucatnyi szabályra és több száz tényre is. A számítási komplexitás tovább csökkenthető, ha a szabályokat rangsoroljuk, és így az első szabállyal a tények (a választható robotok) száma radikálisan csökkenthető.

Köszönettel és tisztelettel,



Boleraczki Miklós

Veszprém, 2025. 12. 03.