PAPP ÁDÁM[1]

Ipari robotkarok ütemezése kombinatorikus eszközökkel

1. BEVEZETÉS

Napjainkban a technológia fejlődésének és az Ipar 4.0 térhódításának köszönhetően egyre nagyobb szerepet játszik az automatizálás és az informatika az ipari gyártórendszerekben. A vállalatok számítógép-vezérelt rendszereket alkalmaznak a termelékenység növelése érdekében, így próbálva lépést tartani a piaci igényekkel. Következésképpen, a hasonló rendszerek megfelelő modellezése, ütemezése és így a termelékenység optimalizálása kulcskérdés, ezért azt a szakirodalom számos területen vizsgálja úgy, mint: festés/fényezés, félvezetőlapkák megmunkálása, összeszerelés, anyagmozgatás. Utóbbi kategóriába tartoznak a robotikus gyártócellák, amelyeket szintén széles körben alkalmaznak.

A robotikus gyártócellák a nyersanyagot vagy félkész terméket megmunkáló berendezésekből, a munkadarabokat mozgató ipari robotkarokból, valamint bementi és kimeneti tárolókból állnak. Ezek a gyártócellák legtöbbször olyan környezetben használatosak, ahol folytonos tömegtermelés zajlik, így a legyártandó mennyiségek állandók. Az ilyen esetekben legtöbbször a hosszútávú termelés ütemezése történik ciklikus módszerrel, azaz egy rövidtávú ütemezés – egy ciklus – meghatározása a feladat, amely ezután periodikusan ismételhető. A robotikus gyártócellák ciklikus ütemezése során a cél egy cikluson belül meghatározni a robot mozgásának egy olyan ütemezését, amely esetén a gyártásra vonatkozó korlátozások nem sérülnek, és az elért profit, valamint a ciklusidő hányadosa maximális.

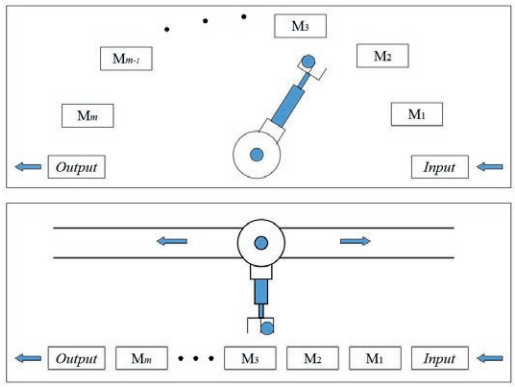
Munkám során a fenti ütemezési feladatosztályt elemeztem, és különböző paraméterek figyelembevételével azonosítottam a legfontosabb alprobléma-osztályokat. Ezek modellezésre kerültek az S-gráf módszertannal, mint többféle feladatosztályra sikeresen alkalmazott kombinatorikus módszerrel. Emellett több közelmúltbéli, szakirodalomban fellelhető MILP modell került implementálásra ugyanezen feladatosztályhoz. Végül a különböző módszerek és modellek összehasonlítása következett tudásuk és hatékonyságuk alapján.

[1] SZE alapképzés, mérnökinformatikus. Témavezető: Dr. Hegyháti Máté, SZE Gépészmérnöki Informatikai és Villamosmérnöki Kar, Informatika Tanszék, tudományos főmunkatárs.

2. IRODALMI ÁTTEKINTÉS

A robotikus gyártócellák ütemezési feladatosztálya először Sethi és társai által 1992-ben jelent meg a szakirodalomban.[2] Azonban ütemezési szempontból nagy hasonlóságot mutat Phillips és Unger által 1972-ben publikált ún. Hoist Scheduling Problem-mel,[3] ahol az anyagmozgatást nem egy robotkar végzi, hanem egy sínen mozgó daru. A két problémaosztály szemléletes összehasonlítása látható az 1. ábrán. Az évek során több különböző módszeren alapuló megoldást publikáltak mindkét feladatosztályra. Abd és társai (2016)[4] Fuzzy logikán alapuló megközelítést használtak, Zhou és Li (2003)[5] pedig Tabu keresési heurisztikára fókuszáltak. Batur és társai (2016)[6] az utazóügynök probléma alapján modellezték, majd szimulált hűtésen alapuló algoritmust publikáltak. Al-Ahmari (2016)[7] pedig Petri hálók segítségével oldotta meg a problémát. A leginkább széleskörben publikált módszer viszont a vegyes egészértékű lineáris programozás (Mixed Integer Linear Programming) volt, például Feng és társai (2018).[8]

1. ábra: A robotikus gyártócellák és a Hoist Scheduling problem felépítése



[2] Sethi, S. P. – Sriskandarajah, C. – Sorger, G. – Blazewicz, J. – Kubiak, W.: Sequencing of parts and robot moves in a robotic cell, in *International Journal of Flexible Manufacturing Systems*, 1994/4. no., pp. 331–358.

[3] Phillips, L. W. – Unger, P. S.: Mathematical programming solution of a hoist scheduling program, in *AIIE transactions*, 1976/8. no., pp. 219–225.

[4] Adb, K. – Abhary, K. – Marian, R.: Multi-objective optimisation of dynamic scheduling in robotic flexible assembly cells via fuzzy-based Taguchi approach, in *Computers & Industrial Engineering*, 2016/99. no., pp. 250–259.

[5] Zhou, Z. – Li, L.: Single hoist cyclic scheduling with multiple tanks: a material handling solution, in *Computers & Operations Research*, 2003/6. no., pp. 811–819.

[6] Batur, G. D. – Erol, S. – Karasan, O. E.: Robot move sequence determining and multiple part-type scheduling in hybrid flexible flow shop robotic cells, in *Computers & Industrial Engineering*, 2016/100. no., pp. 72–87.

[7] Al-Ahmari, A.: Optimal robotic cell scheduling with controllers using mathematically based timed Petri nets, in *Information Sciences*, 2016/329. no., pp. 638–648.

[8] Feng, J. – Chu, C. – Che, A.: Cyclic jobshop hoist scheduling with multi-capacity reentrant tanks and time-window constraints, in *Computers & Industrial Engineering*, 2018/120. no., pp. 382–391.

3. ROBOTIKUS GYÁRTÓCELLÁK OSZTÁLYOZÁSA ÜTEMEZÉSI SZEMPONTBÓL

A robotikus gyártócellák nagyon sokfélék lehetnek, így kifejezetten sok alprobléma-osztály azonosítható közöttük. Ebben a fejezetben az ütemezésük szempontjából legfontosabb csoportosítási szempontok és paramétereik kerülnek ismertetésre.

Robotkar típusa:

- Single gripper: Egy időben csak egy munkadarab mozgatására képesek, azaz egy munkadarab megmarkolására alkalmas fogószerkezettel rendelkeznek.
- Dual gripper: Egy időben akár két munkadarab mozgatására is képesek, azaz két termék megmarkolására alkalmas fogószerkezettel rendelkeznek.

Termékek száma:

- Single part type cell: Olyan gyártócella, amelyben csak egyféle terméket gyártanak, egyféle recept szerint.
- Multiple part type cell: Olyan gyártócella, amelyben többféle terméket gyártanak, többféle recept szerint.

Gyártási útvonal:

- Flow-shop: Az összes terméknek azonos sorrendben kell végigmenni az összes berendezésen, mindegyiket pontosan egyszer érintve.
- Job-shop: A különböző termékek útvonala eltérhet egymástól, illetve megengedett, hogy útjuk során egy berendezést akár többször vagy egyszer sem látogassanak meg.
- Open-shop: A receptben a termékek útvonala nincs előre meghatározva, hanem az ütemezése során eldöntendő.

Várakozási rend:

- No-wait: A terméket a feldolgozásának befejezésekor azonnal el kell szállítania a robotkarnak, egyébként a termék károsodik.
- Interval: Ekkor a termék a feldolgozásának befejezése után csak egy korlátozott ideig várakozhat a berendezésben.
- Free-pickup: A robotkarnak nem muszáj a terméket a feldolgozásának befejezésekor azonnal elszállítani, hanem ezt a későbbiekben bármikor megteheti.

Szállítási idők:

- Konstans: A gyártócellában található összes berendezés között a szállítási idők azonosak, függetlenül a berendezések és a robotkar elhelyezkedésétől.
- Additív: A gyártócellában lévő berendezések és tárolók elhelyezkedését úgy egyszerűsíti, mintha ezek egy egyenes mentén, egymástól egyenlő távolságra helyezkednének el. Így bármely két egymást követő berendezés között a szállítási idő azonos (leggyakrabban δ-val jelölik), két nem egymást követő berendezés között pedig ennek egész számú többszöröse.
- Euklideszi: Három tetszőleges berendezés közötti szállítási idők teljesítik a háromszög egyenlőtlenséget. Azaz ha például *i, j, k* ebben a sorrendben három berendezés, δ pedig a szállítási idő (amely minden esetben feltételezhető, hogy szimmetrikus), akkor $\delta_{ij} + \delta_{jk} \geq \delta_{ik.}$

4. CIKLIKUS ÜTEMEZÉS

A tömegtermelést végző rendszerekben a legyártandó mennyiségek állandók, ilyenkor célunk a hosszútávú profit maximalizálása. Ámde egy ilyen hosszútávú optimális ütemezés meghatározása a komplexitásából fakadóan nagy valószínűséggel kombinatorikus robbanást eredményezne, és nem lenne megoldható belátható időn belül. Erre a problémára jelenthet megoldást a ciklikus ütemezés használata. Lényege, hogy hosszútávú ütemezés helyett egy rövidtávú ütemezést, egy ciklust határozunk meg, amely ezután periodikusan ismételhető. Az ismételhetőség feltétele, hogy a robotkarnak a ciklus kezdetén és végén ugyanabban a pozícióban kell lennie.

Ciklikus ütemezés használata esetén a cél: egy cikluson belül meghatározni a robot mozgásának egy olyan ütemezését, amely esetén a gyártásra vonatkozó korlátozások nem sérülnek és az elért profit, valamint a ciklusidő hányadosa maximális. A nemlineáris optimalizálás nehézségeinek elkerüléséért sok módszerben ezt úgy kezelik, hogy egy rögzített gyártási mennyiséget feltételezve minimalizálják a ciklusidőt.

A rögzített mennyiség definiálására több fogalmat is bevezettek. Az egyik ilyen az ún. r-degree cycle, amely egy olyan speciális ciklus, amin belül pontosan r darab termék gyártása kezdődik el és fejeződik be. Emellett használatos még az ún. Minimal Part Set (MPS – minimális elemhalmaz) kifejezés, amely egy olyan minimális elemszámú halmaz, amelyben a különböző termékek aránya megegyezik a gyártási igényekben megjelenő arányokkal.

Egy másik fontos megemlítendő tulajdonsága a ciklikus ütemezésnek az ún. interlacing (átlapolódás). A fogalom azt takarja, hogy a tevékenységek végrehajtása nem feltétlenül fejeződik be ugyanazon cikluson belül, amelyikben elkezdődött, hanem átlapolódik egy vagy akár több rákövetkező ciklusba. Ezt a jelenséget meglehetősen nehézkes megfelelő módon lemodellezni, így majd látható lesz, hogy az egyes módszerek mennyire különbözően kezelik, már amennyiben képesek rá.

5. MILP MODELLEK ÖSSZEHASONLÍTÁSA

Az irodalmi áttekintés utolsó bekezdésében említett 4 MILP modell került összehasonlításra a robotikus gyártócellák ütemezése feladatosztályra. Az egyszerűsítés kedvéért az alábbi rövidítések kerülnek bevezetésre a továbbiakban:

- **Z03** Zhou és Li (2003), precedencia alapú modell[9]
- **L14** Li és Fung (2014), precedencia alapú modell[10]
- **G18** Gultekin és társai (2018), hibrid, időegység/prec. alapú modell[11]
- **F18** Feng és társai (2018), precedencia alapú modell[12]

Az 1. táblázat a modellek tudását hasonlítja össze a robotikus gyártócellák főbb, releváns ütemezési paraméterei alapján.

1. táblázat: A 4 MILP modell képességei

	G18	F18	L14	Z03
1 termék	★	★	✓	✓
Flow shop	✓	★	✓	✓
Interval várakozási rend	✗	✓	✓	✓
Eltérő üres és foglalt robot mozg.	✗	✓	✓	✓
r-degree ciklusok	✓	✓	✓	✗

Mindegyik modell az eredeti cikk alapján implementálásra, majd 2 teszteset alapján összehasonlításra került. A tesztpéldaként a sokat idézett, 12 állomást tartalmazó Phillips és Unger (1972) példa lett használva. Mivel az 1. táblázatból is látható, hogy a modellek eltérő tulajdonságú problémákat képesek kezelni, ezért az eredeti tesztpélda az adott tesztesetnek megfelelően lett módosítva. A teszteket futtató számítógép Intel i5-7200U 3GHz processzorral, 8GB memóriával rendelkezett, a megoldó szoftver pedig a Gurobi Optimizer 8.0.1-es verziója volt gyári beállításokkal.

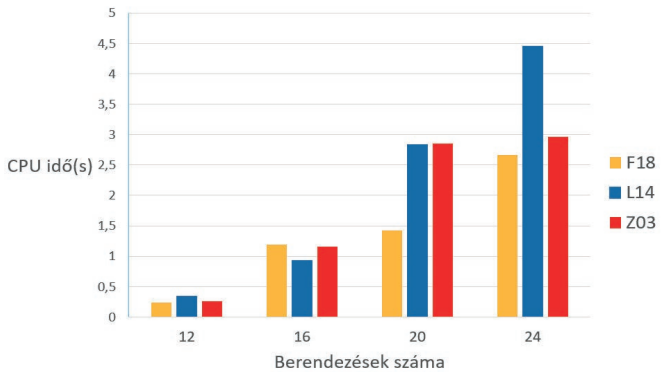
[9] Zhou, Z. – Li, L.: i. m. 2. o.
[10] Li, X. – Fung, R. Y.: A mixed integer linear programming solution for single hoist multi-degree cyclic scheduling with reentrance, in *Engineering Optimization*, 2014/5. no., pp. 704–723.
[11] Gultekin, H. – Coban, B. – Akhlaghi, V. E.: Cyclic scheduling of parts and robot moves in m-machine robotic cells, in *Computers & Operations Research*, 2018/90. no., pp. 161–172.
[12] Feng, J. – Chu, C. – Che, A.: i. m. 2. o.

Az első tesztesetben a következő tulajdonságokkal rendelkező tesztpéldán lettek összehasonlítva a modellek:

- 1 termék,
- flow shop,
- interval várakozási rend,
- különböző foglalt és üres robot mozgási idők,
- 1-degree ciklusok.

A skálázási paraméter ebben az esetben a berendezések száma volt, amely 12 és 24 között, 4-es lépésközzel változott. Mivel a G18 modell nem képes sem eltérő üres-foglalt robot mozgást, sem pedig interval várakozási rendet kezelni, ezért az első tesztesetből kihagyásra került. A stage-ek számának változására bekövetkező futásidőbéli eltérések a 3 modell esetén a 2. ábrán láthatóak.

2. ábra: Az első teszteset eredményei



Leolvasható, hogy mindhárom modell csak kis mértékű, alig néhány másodperces futásidő növekedéssel reagált a stage-ek számának növelésére, így elmondható, hogy ezen paraméter változása nincs szignifikáns hatással a modellek hatékonyságára.

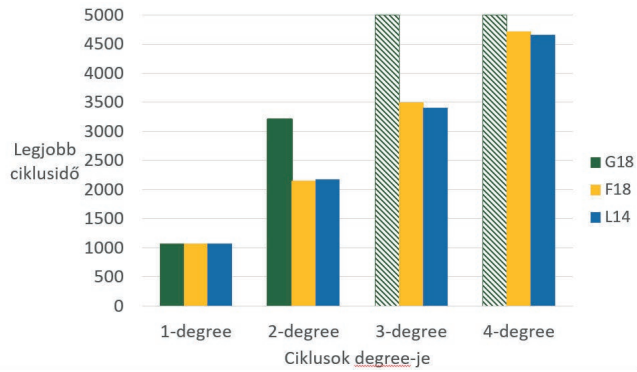
A második tesztesetben a következő tesztelési paraméterek lettek használva:

- 1 termék,
- flow shop,
- free-pickup várakozási rend,
- 12 különböző berendezés,
- azonos foglalt és üres robot mozgási idők.

A skálázási paraméter ebben az esetben ciklusok degree-je volt, azaz hogy egy cikluson belül mennyi terméket gyártunk. Mivel ebben az esetben a Z03 modell nem képes a teszteset paramétereit kezelni, ezért itt ez a modell került kihagyásra.

A modellek ennél a tesztesetnél kizárólag 1 degree ciklus esetén találták meg az optimális megoldást belátható időn belül, ezért egy 200 másodperces időlimit került beiktatásra. A 3. ábrán az látható, hogy az időkorlát lejártakor mi volt a legjobb megtalált megoldás (legkisebb ciklusidő) az egyes degree-k esetén. A grafikonról leolvasható, hogy az F18 és L14 modellek közel azonosan teljesítettek, viszont a G18 modell 3, valamint 4 degree esetén egyáltalán nem talált még megoldást sem az időkorláton belül. Megemlítendő még, hogy 5 degree esetén már egyik modell sem talált megoldást az időlimiten belül.

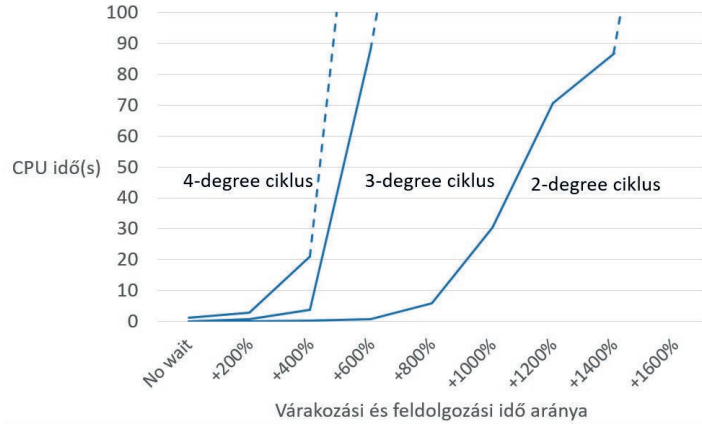
3. ábra: A második teszteset eredményei



Ebből következően merült fel a kérdés, hogy van-e kapcsolat, és ha igen, akkor milyen a ciklusok degree-je és a modellek hatékonysága között? Továbbá mivel az összes modell jóval gyengébben teljesített free-pickup várakozási rend esetén. Így arra is kíváncsi voltam, hogy van-e kapcsolat, és ha igen, akkor pontosan milyen a várakozási rend és a hatékonyság között?

Ennek kiderítésére egy harmadik tesztesetnél 2 skálázási paraméter is bevezetésre került, elsőként az ütemezés degree-je, másodikként pedig a várakozási rend. Utóbbinál egyre növelve lett az időablak, amely meghatározza, hogy egy termék meddig várakozhat egy berendezésben. Így a no-wait-től (amely esetén a megengedett várakozás értelemszerűen 0) az interval-on át a free-pickup várakozási rend is modellezhető (ahol már szignifikánsan nagy a várakozás, amely tekinthető végtelennek). A 4. ábra grafikonjának x-tengelye azt mutatja az F18 modell esetén, hogy a berendezéseknél a megengedett várakozás hányszorosa a feldolgozási időnek. A szaggatott vonal azt jelzi, hogy a modell elérte az időkorlátot, azaz nem találta meg az optimális megoldást 200 másodpercen belül.

4. ábra: A harmadik teszteset eredményei



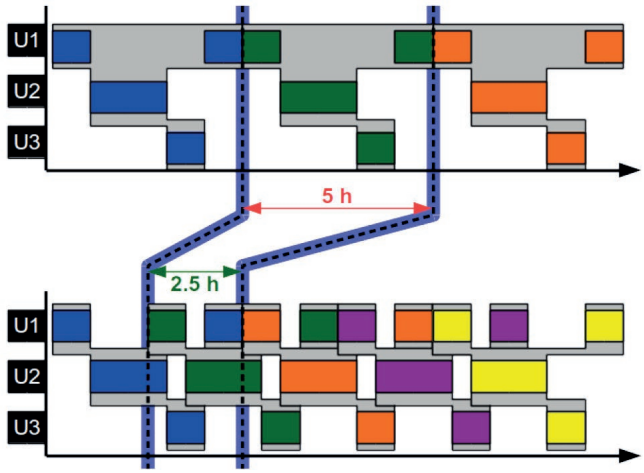
Összességében elmondható, hogy a free-pickup várakozási rend nagyban rontja a modellek hatékonyságát, illetve a ciklusok degree-jének növelésével is egyre hamarabb lépik túl az időkorlátot. Tehát szoros kapcsolat látható a várakozási rend és a hatékonyság, valamint aközött, hogy hány terméket gyártunk ciklusonként.

6. AZ S-GRÁF MÓDSZERTAN

Az S-gráf módszertan egy irányított gráfokon alapuló kombinatorikus ütemezési/optimalizálási keretrendszer. A robotikus gyártócellák feladatosztály a fent említett tesztesetek használatával S-gráf módszertannal szintén modellezésre került. Azonban az S-gráf másképp kezeli az interlacing-et, amely azt eredményezi, hogy nem hasonlítható teljes mértékben össze a MILP-pel, mivel elsősorban interval és free-pickup várakozási rend esetén más-más célfüggvény értékeket, így más optimális megoldást adnak. Az 5. ábrán szemléletes módon megfigyelhető a különbség az interlacing kezelése között. A felső diagram az S-gráf, az alsó a MILP által adott egy lehetséges ütemezés. A ciklushatárokat a kék vonal jelöli.

Ellenben az eddigi tesztek alapján elmondható, hogy az S-gráf nem produkálja a MILP-eknél megfigyelt futásidőbéli kiugrást free-pickup várakozás esetén. Mint ahogy a ciklusok degree-jének növelése sem jár kiugró futásidő növekedéssel. Például a második tesztesetben 2-degree esetén, míg a MILP-ek még az optimumot sem találták meg, addig az S-gráf 0,26 s alatt megtalálta azt. Mindehhez interval várakozás esetén csupán 0,077 s-ra volt szüksége, 1 degree esetén pedig alig mérhető nagyságú (0,001 s) futásidővel oldotta meg a feladatot.

5. ábra: Az interlacing kezelésének összehasonlítása



7. ÖSSZEFOGLALÁS

A robotikus gyártócellák ütemezése feladatosztállyal, illetve a ciklikus ütemezéssel ismerkedtem meg munkám során. Azonosítottam a legfontosabb ütemezési paramétereket, majd ezek alapján osztályoztam a gyártócellákat. Implementáltam 4 szakirodalomban publikált MILP modellt, majd összehasonítottam őket több tényező alapján. Megállapítható volt, hogy a várakozási rend és az egy ciklusban gyártott termékek száma nagyban befolyásolta hatékonyságukat. Továbbá Feng és társai által 2018-ban publikált modell bizonyult a leghatékonyabbnak, és ez tudta kezelni a legtöbb típusú feladatot.

Az S-gráf módszerrel is megismerkedtem munkám során, amellyel szintén modellezésre került a feladatosztály. Az eddigi tesztek alapján ez a megoldó módszer sokkal kevésbé érzékeny az említett paraméterek megváltozására, továbbá az eddigiekben hatékonyabbnak is bizonyult a futásidőket tekintve. Ellenben a ciklikus ütemezés egy jelentős kihívását az interlacing-et nem tudja megfelelően kezelni, ezért a módszerek teljeskörű összehasonlítása még nem lehetséges.

A jövőbeli tervek között szerepel az S-gráf módszer mélyrehatóbb vizsgálata és továbbfejlesztése annak érdekében, hogy megfelelő módon tudja kezelni az interlacing-et, így még alkalmasabbá váljon a ciklikus ütemezésre.

FELHASZNÁLT IRODALOM

- Adb, K. – Abhary, K. – Marian, R.: Multi-objective optimisation of dynamic scheduling in robotic flexible assembly cells via fuzzy-based Taguchi approach, in *Computers & Industrial Engineering*, 2016/99. no., pp. 250–259.
- Al-Ahmari, A.: Optimal robotic cell scheduling with controllers using mathematically based timed Petri nets, in *Information Sciences*, 2016/329. no., pp. 638–648.
- Batur, G. D. – Erol, S. – Karasan, O. E: Robot move sequence determining and multiple part-type scheduling in hybrid flexible flow shop robotic cells, in *Computers & Industrial Engineering*, 2016/100. no., pp. 72–87.
- Feng, J. – Chu, C. – Che, A.: Cyclic jobshop hoist scheduling with multi-capacity reentrant tanks and time-window constraints, in *Computers & Industrial Engineering*, 2018/120. no., pp. 382–391.
- Gultekin, H. – Coban, B. – Akhlaghi, V. E.: Cyclic scheduling of parts and robot moves in m-machine robotic cells, in *Computers & Operations Research*, 2018/90. no., pp. 161–172.
- Li, X. – Fung, R. Y.: A mixed integer linear programming solution for single hoist multi-degree cyclic scheduling with reentrance, in *Engineering Optimization*, 2014/5. no., pp. 704–723.
- Phillips, L. W. – Unger, P. S.: Mathematical programming solution of a hoist scheduling program, in *AIIE transactions*, 1976/8. no., pp. 219–225.
- Sethi, S. P. – Sriskandarajah, C. – Sorger, G. – Blazewicz, J. – Kubiak, W.: Sequencing of parts and robot moves in a robotic cell, in *International Journal of Flexible Manufacturing Systems*, 1994/4. no., pp. 331–358.
- Zhou, Z. – Li, L.: Single hoist cyclic scheduling with multiple tanks: a material handling solution, in *Computers & Operations Research*, 2003/6. no., pp. 811–819.