

## Városi buszok útvonaltervezése Hangyakolónia módszer segítségével

Dimitriu Adonisz\* Harmati István\*  
Kővári Bálint\*\*

\**Irányítástechnika és Informatika tanszék, Budapesti Műszaki és Gazdaságtudományi Egyetem  
(e-mail: d.adonisz96@gmail.com, harmati@iit.bme.hu)*

\*\* *Közlekedés és- Járműirányítási Tanszék, Budapesti Műszaki és Gazdaságtudományi Egyetem  
(email: kovari.balint@kjk.bme.hu)*

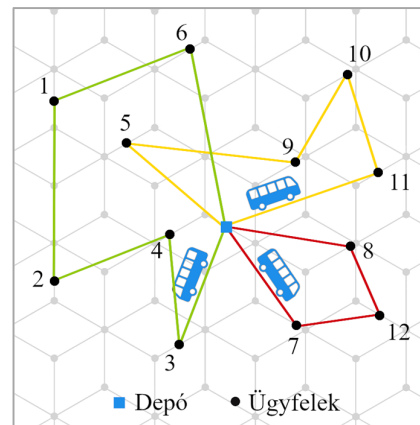
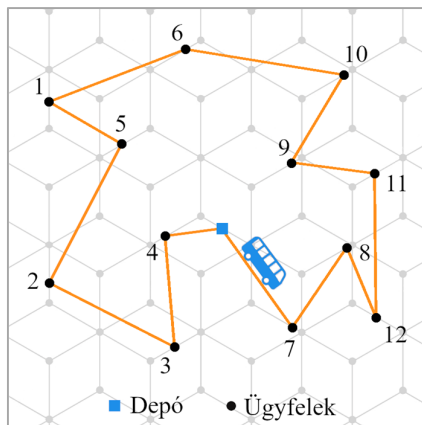
Az intelligens közlekedési rendszerek (ITS: Intelligent Transportation systems) egyik fő feladata a földfelszíni közlekedés fejlesztése. A földfelszíni tömegközlekedés jellemzően kötött vonalú utazást jelent fix útvonallal, megállókkal és menetrenddel, melyben a városi autóbuszok kulcsfontosságú szerepet töltenek be. Ebben a tanulmányban a tömegközlekedés egy új formáját vetjük fel, melyben a buszok szabadon, az utasokhoz alkalmazkodva közlekednek (és nem fordítva) úgy, hogy felveszik őket az aktuális pozíciójukon és elszállítják őket a céljukhoz. A feladat olyan útvonalak keresése a hálózaton, amelyek mentén a buszok el tudják szállítani az összes utast úgy, hogy az átlagos utazási idő minimális legyen. Ebben a tanulmányban bemutatjuk a probléma formális leírását és egy hangyakolónia algoritmust (Max-Min Ant System: MMAS), mellyel közel optimális megoldást kapunk.

### 1. BEVEZETÉS

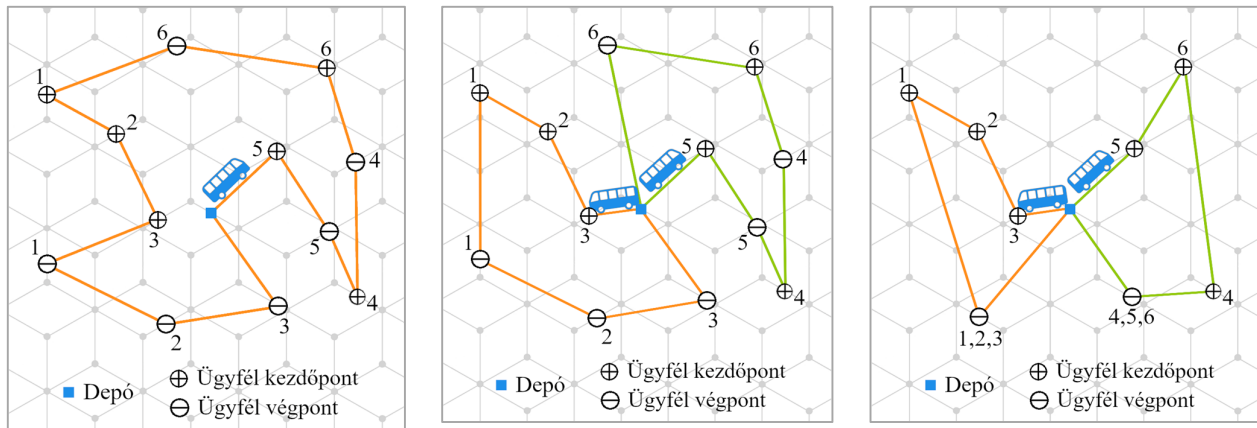
A jól ismert utazóügynök probléma (TSP: Traveling Salesman Problem) általánosan egy vagy több ország városainak az optimális bejárásáról szól, melyben az utazóügynök egy várost csak egyszer érinthet. A cél a legrövidebb útvonal megtalálása. Ennek egy ekvivalens gráfelméleti megfogalmazása: egy élsúlyozott teljes gráfban keressük a legrövidebb Hamilton-utat. A TSP bizonyítottan az NP-nehéz problémaosztályba tartozik (Garey & Johnson, 1979), ezért egzakt megoldást kereső algoritmusok helyett heurisztikus módszereket alkalmaznak az optimális megoldás közelítésére. A jármű útvonaltervezési probléma (VRP: Vehicle Routing Problem) a TSP egy általánosítása, melyben egy olyan úthalmazt keresünk

járművek számára, mely optimálisan szállítja ki az ügyfelek megrendeléseit egy adott telephelyről (a továbbiakban „depó”). A klasszikus VRP – más néven kapacitált VRP (CVRP: Capacitated Vehicle Routing Problem) – egy depót feltételez, ahol egy bizonyos számú (áru-kapacitással ellátott) jármű tartózkodik, hogy kiszolgálják több ügyfél szállítási igényét.

Az 1. ábrán a TSP és VRP útvonalainak illusztrációi láthatók. Kényszereket csatolva a VRP-hoz, lehetőségünk van a standard modellt a különböző alkalmazásokhoz illeszteni. P. Toth és D. Vigo egy tekintélyes terjedelmű és kiemelkedő jelentőségű munkája a témában ismerteti az addig megjelent variánsokat és a megoldásukra irányuló különböző megközelítéseket is (Toth & Vigo, 2002). Az árufelvevő-átadó



1. ábra: TSP (bal oldal) és VRP (jobb oldal)



2. ábra: egy-járműves DARP (bal oldal), több járműves DARP (középen) és SBRP (jobb oldal)

VRP (PDVRP: Pickup and Delivery Vehicle Routing Problem) vagy még általánosabban árufelvevő-átadó probléma (PDP: Pickup and Delivery Problem) egy speciális variánsa a VRP-nak, ahol a járművek szállítási kéréseket teljesítenek. Az 1-1 PDP-ben a járművek árucikket vagy embereket szállítanak egy kezdő és végpozícióból álló lokáció-pár között. Ha a szállítás emberekre vonatkozik, akkor a probléma neve hívj-egy-fuvar (DARP: Dial-a-Ride Problem), melyben az ügyfeleket a céljukhoz kell szállítani. Egy átfogó ismertető a problémáról megtalálható a (Cordeau & Laporte, 2003b) hivatkozásban.

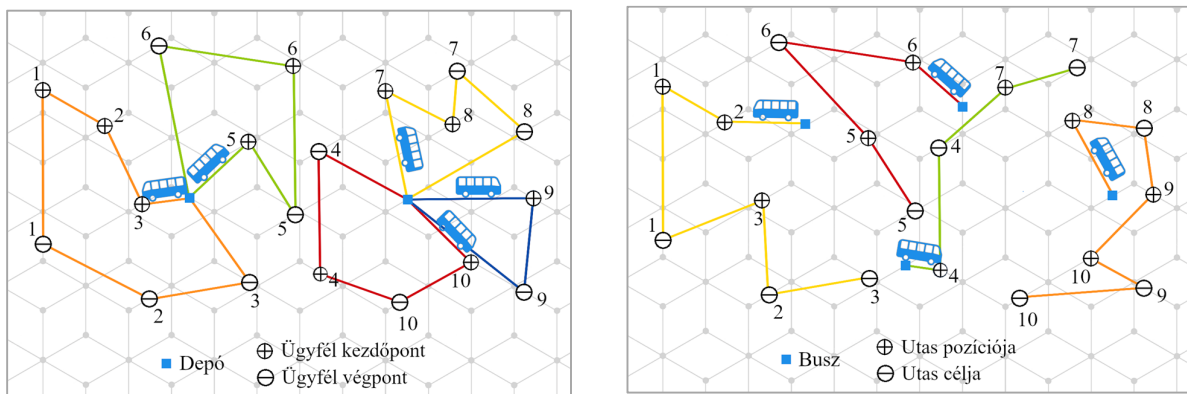
A DARP problémacsaládot az ügyfélközpontú perspektíva lényegesen megkülönbözteti a többi VRP variánstól, lévén az emberi elégedettség – vagy éppen elégedetlenség – nem egy matematikailag egzakt jelenség. Létezik egy-járműves (Psaraftis, 1980) és több-járműves (Cordeau & Laporte, 2003a) DARP, az előbbiben egyetlen jármű, az utóbbiban több jármű is rendelkezésre áll a kiszolgálásra. Egy gyakorlati alkalmazási területe a több-járműves DARP-nak az úgynevezett iskolabusz probléma (SBRP: School Bus Routing Problem), mely az amerikai iskolabuszok útvonaltervezésével foglalkozik. Az iskolabuszok egyetlen depóban helyezkednek el, innen indulnak a diákokért, akiket a környező iskolákba

szállítanak, mintegy háztól-házig való közlekedést megvalósítva.

Egy figyelemre méltó összefoglaló a témáról megtalálható a (Park & Kim, 2010)-es hivatkozásban. A 2. ábrán ennek a három variánsnak a különböző jellegzetességű útvonalai láthatók. A több-depós DARP-ban (Braekers et al., 2014), (Detti et al., 2017) a járművek különböző helyekről indulhatnak az ügyfelekért. Ez az általánosítás az, amely a legtöbb fejtörést okozza a problémával foglalkozóknak. A 3. ábrán a több-járműves DARP és a dolgozatban szereplő útvonaltervezés összevetése látható ugyanabban a környezetben.

A több-járműves több-depós DARP áll legközelebb a városi buszok útvonaltervezéséhez, az összehasonlítás végett a főbb tulajdonságai alább látható:

- (i) A problémát egy teljes gráf reprezentálja
- (ii) Minden jármű a neki kijelölt depóból indul
- (iii) Minden jármű valamelyik depóban fejezi be az útját
- (iv) Egy ügyfelt csak egy jármű szolgálhatja ki
- (v) Az ügyfelek felvétele időben megelőzi a szállításukat



3. ábra: több-járműves DARP (bal oldal) és buszok útvonaltervezése (jobb oldal)

- (vi) A megoldás útvonalak olyan halmaza, mely minden csomópontot (kivéve esetleg a depókat) csakis egyszer tartalmaz
- (vii) A cél az úthalmaz minimalizálása az ügyfél-elégedetlenség szempontjából

A (iv)-es, (v)-ös és (vii)-es tulajdonság igaz a jelen problémára is. A (ii) tulajdonságban a depók értelmezhetők úgy, mint a buszok kiindulási pozíciója, ahol minden depóban egyetlen busz helyezkedik el. A (iii)-as tulajdonság nem áll fenn, mivel a buszok az utolsó letett utasnál fejezik be az útjukat, ezzel a depó fogalom elveszti jelentőségét. Az (i) tulajdonságban említett teljes gráf pontjai az ügyfelek kezdő- és végpontjaiból és a depókból áll, minden pontpár között fut egy él, melyen egy költségfüggvény értelmezett, ami általában távolság vagy utazási idő alapú, mint a (Cordeau & Laporte, 2003a), (Ropke & Pisinger, 2006) hivatkozásokban. Fontos megjegyezni, hogy valójában az összes VRP variáns teljes gráfot használ, melyet az ügyfelekből (a továbbiakban „utasok”) és a depókból generálnak. Az ilyen típusú gráfok leírására a ügyfél-alapú gráf (customer-based graph) terminust használjuk, melyet Huang és munkatársaitól adaptáltunk (Huang et al., 2017), másrésztől az úthálózati gráf kifejezés alatt olyan gráfokat értünk, melyek a valós úthálózatot reprezentálják. Mint említettük, a teljes gráf pontjai a depók és az utasok kezdő- és végpontjai, az ezeket összekötő élek súlya a valós úthálózati elrendezést veszi figyelembe. Ez kétféleképpen történhet:

- a) Az egyik eset, hogy az élsúlyok az egyes utasok és depók közötti légvonalbeli távolságot jelölik az úthálózatban. Ilyenkor a DARP megoldása egy olyan pontsorozatot ad, mely csak szűk keretek között tekinthető érvényesnek, például akkor, ha a feladat több város bejárása, mert ekkor a városok meglátogatásának sorrendje fontosabb lehet magánál az útvonaltól. Azonban ez hibára vezet, ha például a valóságban folyó fut a városok között.
- b) A másik mód, hogy az élsúlyok a legrövidebb távolságot tartalmazzák két pont között. Ebben az esetben a DARP megoldásaként kapott pontsorozat érvényesnek tekinthető, viszont egy  $n$  csomópontú ( $n$ =depók száma+2-utasok száma) ügyfél-alapú gráfban  $(n(n-1))/2$  számú útvonalat kell előre meghatározni, valamilyen útkereső algoritmus (például A\*) segítségével.

Az a) esetben a légvonalbeli távolság használata legtöbb esetben megbízhatatlan eredményre vezet és további számításokat igényel, hogy tényleges útvonalat kapjunk. A b) esetben előfeldolgozást kell végezni, amely során végeredményben csökken a gráf pontjainak a száma, viszont nő az élek száma.

Ebből a perspektívából nézve vitatható az az állítás, hogy az ügyfél-alapú gráf adaptálása teljesítményben és minőségben felülmúlja az úthálózati gráfok használatát. Ennélfogva a jelen tanulmányban úthálózati gráfot alkalmazunk, ügyfél-alapú gráf helyett. Következésképp a megoldásunk olyan úthalmaz, mely – (vi) ellenére – nem feltétlen tartalmazza az összes csomópontot a gráfban.

A dolgozatnak nem célja kimeríteni a feladat kapcsán felmerülő összes problémát, így például nem foglalkozik a következő tulajdonságokkal:

- Buszok kapacitása
- Idő-ablakok (Ropke et al., 2007)
- Jármű heterogenitás (Parragh, 2011)
- A probléma dinamikus változata (Berbeglia et al., 2010)

Mivel a DARP a TSP egy általánosítása, ez is az NP-nehéz problémaosztályba tartozik, így többen metaheurisztikus algoritmusokat fejlesztettek ki ennek a megoldására. J.-F. Cordeau és G. Laporte egy tabu-search algoritmust dolgozott ki az egy-depós DARP-re (Cordeau & Laporte, 2003a), ezt később egy branch-and-cut algoritmus követte (Cordeau, 2006), majd mások genetikus algoritmust (Jorgensen et al., 2007) és variable neighborhood search metaheurisztikát (Parragh et al., 2010) alkalmaztak. A több-depós DARP-vel kevés tanulmány foglalkozott eddig, a probléma komplex természete miatt. A (Braekers et al., 2014) szerzői sikeresen adaptálták a branch-and-cut algoritmust a több-depós szcenárióhoz, majd P. Detti és munkatársai egy tabu-search és egy variable neighborhood search algoritmust (Detti et al., 2017) publikáltak. A hangyakolónia algoritmus (ACO: Ant Colony Optimization) eddig elkerülte a kutatók figyelmét, összesen egy példát találtunk a szakirodalomban, ahol Tripathy és a munkatársai egy hangyakolónia rendszert (ACS: Ant Colony System) használtak az egy-depós DARP-hoz (Tripathy et al., 2017). A multi-depós variánsra nem találtunk hangyakolóniás megoldást, valamint olyan publikációt sem, amelyben úthálózati gráfot használnának a DARP problémaosztály bármelyik variánsára.

## 2. A PROBLÉMA FORMALIZÁCIÓJA

Legyen  $G = (V, E, c)$  egy összefüggő, irányítatlan és élsúlyozott gráf az úthálózat reprezentációja,  $V$  csúcshalmazzal,  $E = \{(i, j) \mid i, j \in V, i \neq j\}$  élhalmazzal és  $c(i, j)$  élköltséggel minden  $(i, j) \in E$  élen. Praktikus okokból  $V$  felbontható részhalmazokra:  $V = I \cup O \cup D \cup V'$ , ahol  $I = \{1, 2, \dots, k\}$  a buszok kiinduló pozícióinak a halmaza,  $O = \{k + 1, k + 2, \dots, k + n\}$  az utasok kezdeti pozícióinak a halmaza,  $D = \{k + n + 1, k + n + 2, \dots, k + 2n\}$  az utasok célpozícióinak a halmaza, és  $V' = \{k + 2n + 1, k + 2n + 2, \dots\}$  az úthálózat kereszteződéseinek a halmaza. Ettől fogva  $I, O, D$  halmazok elemei tekinthetők az úthálózat virtuális kereszteződéseinek, míg az élek a kereszteződéseket összekötő útszakaszoknak. Az így definiált gráf illusztrációja az 5. ábrán látható. Minden  $P_h = (p_o, p_d)$ ,  $h \in \{1, 2, \dots, n\}$ , utast egy  $p_o = orig(P_h) \in O$  kezdeti- és egy  $p_d = dest(P_h) \in D$  célpozícióból álló utas-pontpár határoz meg. A busz útvonala  $t_z = (v_1, v_2, \dots, v_m)$ ,  $(v_i, v_{i+1}) \in E, \forall i = 1, 2, \dots, m - 1$ , amely legalább egy utas-pontpárt tartalmaz  $v_l = orig(P_h) \in t_z, v_q = dest(P_h) \in t_z, l < q$  sorrendben, vagyis az útvonalnak hamarabb kell tartalmaznia az utas felvételét, mint a lerakását.

Megjegyzendő, hogy a cél az utazási idő (várakozási és buszon töltött idő) minimalizálása az utasok számára és ez különbözik a buszok utazási idejének a minimalizálásától. Emiatt a költségfüggvényt nem a buszokra, hanem az utasokra kell definiálni. Legyen a  $P_h$  utashoz rendelt útvonal a következő pontsorozat:

$$r_h = (v_1, v_2, \dots, v_q), (v_i, v_{i+1}) \in E, \forall i = 1, 2, \dots, q - 1$$

Minden ilyen  $r_h$  útra:

$$v_1 \in I, \quad (1)$$

$$v_q = \text{dest}(P_h) \in D, \quad (2)$$

$$v_l = \text{orig}(P_h) \in r_h, l < q. \quad (3)$$

A  $P_h$  utashoz rendelt út az öt felvevő busz útvonalának egy részeként (szakaszaként):  $r_h \subseteq t_z$  van értelmezve, amely a busztól (1) az utas céljáig (2) tart. A (3) -as kifejezés biztosítja, hogy a pontsorozatnak előbb kell tartalmaznia az utas felvételét, mint a lerakását. Továbbá megjegyzendő, hogy  $r_h$  tekinthető az alábbi két szakasz kompozíciójának:

- $(v_1, v_2, \dots, v_l)$  az utas eléréséig megtett útvonal
- $(v_l, v_{l+1}, \dots, v_q)$  az utas szállításához tartozó útvonal

Ezzel a felbontással a várakozási idő az előbbi, a buszon töltött idő az utóbbi szakasznak feleltethető meg. Legyen  $s_h = (e_1, e_2, \dots, e_{q-1})$ ,  $e_i = (v_i, v_{i+1}) \in E, \forall i = 1, 2, \dots, q - 1$  az utashoz rendelt  $r_h$  út élsorozata. A  $P_h$  utasra vonatkozó út-költség a következőképpen írható fel:

$$c(r_h) = c(s_h) = \sum_{e_i \in s_h} c(e_i) \quad (4)$$

mely azon élek súlyainak az összege, melyeket az utashoz rendelt út érint. A költségfüggvény, az összes utasra vonatkoztatott út-költségek átlaga:

$$\min \frac{1}{n} \sum_{h=1}^n \sum_{e_i \in s_h} c(e_i) \quad (5)$$

Ha szükséges, akkor ez a költségfüggvény kibővíthető további szempontokat is figyelembe vevő költségekkel, például a buszok útvonalának hosszával. Emellett több olyan alkalmazás is elképzelhető, ahol az utasokhoz prioritás rendelhető (például munkába induló utasok). Ez egyszerűen szimulálható azzal, ha minden utas út-költségét egy  $a_h$  skalárral súlyozzuk. A kibővített költségfüggvény a következő:

$$\min \left\{ \gamma \frac{1}{n} \sum_{h=1}^n a_h \sum_{e_i \in s_h} c(e_i) + \delta \frac{1}{k} \sum_{z=1}^k \sum_{e_i \in d_z} c(e_i) \right\} \quad (6)$$

A (6)-os egyenletben a második összegzés a buszok útvonalának a hosszára vonatkozik, ahol  $d_z = (e_1, e_2, \dots, e_{m-1})$ ,  $e_i = (v_i, v_{i+1}) \in E, \forall i = 1, 2, \dots, m - 1$  a  $t_z$  busz útvonalának az élsorozata,  $\gamma$  és  $\delta$  az utasok út-költsége és a buszok útvonalhosszáinak a relatív fontossága. A jelen tanulmányban minden utas egyenlő prioritással rendelkezik:  $a_h = 1, \forall h = 1, 2, \dots, n$ , valamint csak az utasok út-költségeit vesszük figyelembe, tehát az (5)-ös számú egyenletet tekintjük a probléma költségfüggvényének, melyre a továbbiakban útvonal-költségment hivatkozunk.

### 3. Algoritmus

A rajtelligencia egy népszerű kutatási területe lett az utóbbi éveknek (Kennedy, 2006). A kutatók rengeteg energiát fektettek be olyan algoritmusok kifejlesztésébe, melyek bizonyos organizmusok – elsősorban egy fajhoz tartozó állatok – kollektív viselkedésén alapulnak. Ilyen viselkedés például a madarak „V” alakzatban való repülése, a méhek pollen és nektár gyűjtése, a tengeri halrajok örvénylő mozgása és a hangyakolónia táplálékszerzése. A szóban forgó rajok egyedei önmagukban nem rendelkeznek számottevő intelligenciával, viszont egymással és a környezetükkel való interakció során intelligens képességet mutatnak, úgynevezett kollektív intelligenciával rendelkeznek.

Az ACO egy optimalizációs módszer, melyet Marco Dorigo és társai fejlesztettek ki néhány különleges hangyafaj viselkedése alapján (Dorigo et al., 1991), (Dorigo et al., 2006). Ezek a hangyák feromonokat bocsátanak ki a táplálékkeresésük során, mely folyamatos párolgásnak van kitéve. Minden hangya egy úgynevezett feromonösvényt hagy hátra a hangyaboly és az élelemforrás között, amely előnyös útvonalat jelölhet ki a többi hangya számára.

Stütze és Hoos kifejlesztették a MMAS-et, mint az eredeti egy továbbfejlesztett változata, melyben csak a legjobb hangya frissíti a feromonokat. Ezek az algoritmusok effektív és robusztus metaheurisztikának bizonyultak több problémával – különösképpen a VRP variánsaival – szemben (Rizzoli et al., 2007).

A (Tripathy et al., 2017) szerzői 2 ACO-t használtak a DARP problémára: egyiket a járműflotta másikat az útköltség minimalizálására. Ebben a kutatási projektben egy általunk módosított MMAS algoritmust használtunk a buszok útvonaltervezésére, ezt az alaptípust találtuk legalkalmasabbnak az adott probléma megoldására. A matematikai formulák felírásához a (Dorigo et al., 2006)-es cikk jelölésrendszerét vettük alapul.

#### 3.1 Élválasztás

A hangyák iteratív módon, véletlenszerűen építik az útjukat, az élek feromonszintje és *kívánatossága* alapján. Az útépités közben a hangyák nem választhatják az útvonaluk utolsó éleit, vagyis közvetlenül nem fordulhatnak vissza (ezt a *tabu<sub>z</sub>* változó tartalmazza). Ez különbözik a korábban említett ügyfél-alapú gráfokon definiált megszorítástól, ahol a hangya az összes eddig meglátogatott pontra nem léphet (Toth & Vigo, 2002), (Tripathy et al., 2017), mely persze előfeltétele a Hamilton-utak keresésének.



Annak az esélye, hogy a  $z$ -edik hangya az  $(i, j)$  élt választja az adott iterációban, a következő kifejezés határozza meg:

$$p_{i,j}(z) = \begin{cases} \frac{\tau_{i,j}^\alpha \eta_{i,j}^\beta}{\sum_{x \notin tabu_z} \tau_{i,j}^\alpha \eta_{i,j}^\beta}, & \text{ha } j \notin tabu_z \\ 0, & \text{egyébként} \end{cases} \quad (7)$$

ahol  $\tau_{i,j}$  és  $\eta_{i,j}$  az  $(i, j)$  él feromonsűrűsége és kívánatossága,  $\alpha$  és  $\beta$  a feromonösvény és a kívánatosság relatív fontossága, a summázás pedig az összes engedélyezett élre vonatkozik, hogy normalizáljuk a valószínűséget. A kívánatosság egy heurisztikus információt tartalmazó paraméter, ebben a dolgozatban az élköltség reciprokja:  $\eta_{i,j} = \frac{1}{d_{i,j}}$ , ahol  $d_{i,j}$  az  $(i, j)$  él költsége (súlya).

### 3.2 Feromonfrissítés

MMAS algoritmusban minden iteráció végén egy kiválasztott hangya frissíti a feromonokat. Ebben a dolgozatban csak az adott iterációig futott legjobb hangya hagyhat feromonokat. A feromonfrissítés párolgásból és egy megerősítésből áll:

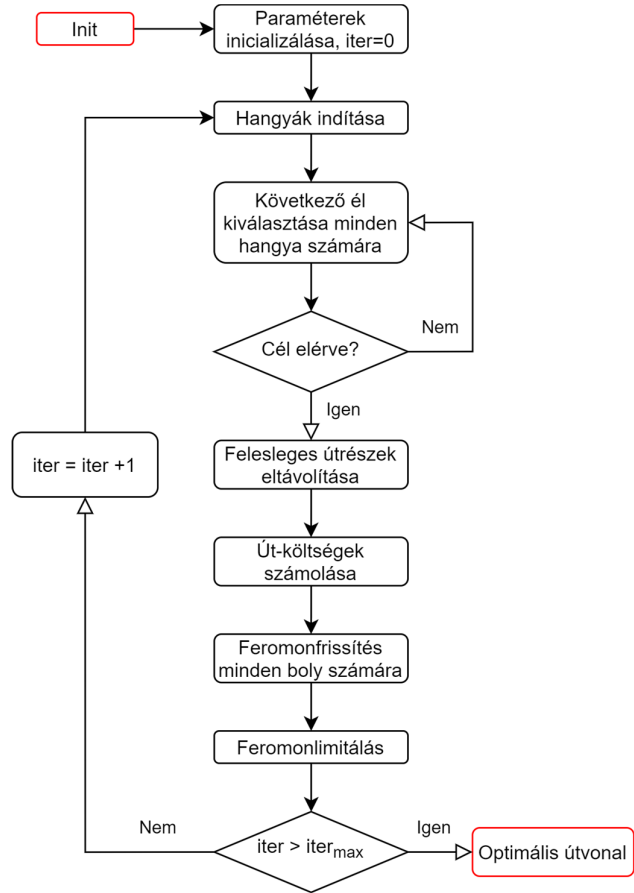
$$\tau_{i,j} \leftarrow [(1 - \rho) \cdot \tau_{i,j} + \Delta \tau_{i,j}^{best}]_{\tau_{min}}^{\tau_{max}} \quad (8)$$

ahol  $\rho \in [0, 1]$  a párolgás mértéke,  $\tau_{max}$  a felső,  $\tau_{min}$  az alsó határa a feromonértékeknek. A megerősítést az eddigi legjobb hangya útvonalán végezzük,  $\Delta \tau_{i,j}^{best}$  az eddigi legjobb útvonal feromonértéke, amely a következő formula szerint számítódik:

$$\Delta \tau_{i,j}^{best} = \begin{cases} \frac{Q}{L_{best}}, & \text{ha a legjobb útvonal tartalmazza } (i, j) \text{ élt} \\ 0, & \text{egyébként} \end{cases}$$

ahol  $L_{best}$  az algoritmus kezdete óta talált legjobb útvonal-költség (az utasok út-költségének átlaga, lásd: (5) egyenlet),  $Q$  pedig a hangyák kezdeti feromoncsomagjának értéke.

### 3.3 Párhuzamos Max-Min hangyarendszer



4. ábra: P-MMAS folyamatábrája

Minden busz kiindulási pozícióját külön hangyabolynak tekintjük. Az ezekből kilépő hangyák élről-élre lépkedve egymással párhuzamosan építik az útvonalakat, mellyel végeredményben közvetlenül a buszok útvonalát adják vissza. A hangyák csak a saját fajtáik (a saját bolyukból kilépő más hangyák) által hagyott feromonokra reagálnak, vagyis minden boly saját feromontérképpel rendelkezik. Egy iterációban a buszok számával megegyező  $k = |I|$  hangya indul útra, minden bolyból egy, melyek a saját feromontérképükön

	Úthálózat mérete	Buszok $\times$ utasok	Legjobb útvonal-költség	Átlagos útvonal-költség	Átlagos futási idő
1.	4 $\times$ 5	3 $\times$ 5	183	192.22	5.13 s
2.	4 $\times$ 5	3 $\times$ 6	201.2	226.4	6.17 s
3.	4 $\times$ 6	3 $\times$ 5	212	249.33	8.31 s
4.	4 $\times$ 6	3 $\times$ 6	231.6	272.04	9.55 s
5.	5 $\times$ 6	3 $\times$ 5	258.4	339.65	12.92 s
6.	5 $\times$ 6	3 $\times$ 6	288.2	350.12	13.74 s
7.	5 $\times$ 6	3 $\times$ 7	273	319.52	14.52 s

1. Táblázat: P-MMAS teljesítménye

$Iter_{max}$	$\alpha_{max}$	$\alpha_{min}$	$\beta_{max}$	$\beta_{min}$	$\rho$	$Q$	$\tau_{max}$	$\tau_{min}$
700	1.9	0.9	0.8	0.5	0.5	70	0.9	0.06

2. táblázat: P-MMAS paraméterkészlete

hagynak feromonokat a (8)-as egyenletnek megfelelően. Az éleket egymással párhuzamosan választják a (7)-es egyenletnek megfelelően, egészen a cél eléréséig, mely továbbra is az összes utas elszállítását jelenti. A cél elérésekor a  $z_i$ -edik hangya útvonal:

$$t_{z_i} \leftarrow (v_{1i}, v_{2i}, \dots, v_{q_i}), v_{j_i} \in t_{z_i}, \forall j \in \{1, 2, \dots, q\} \quad (9)$$

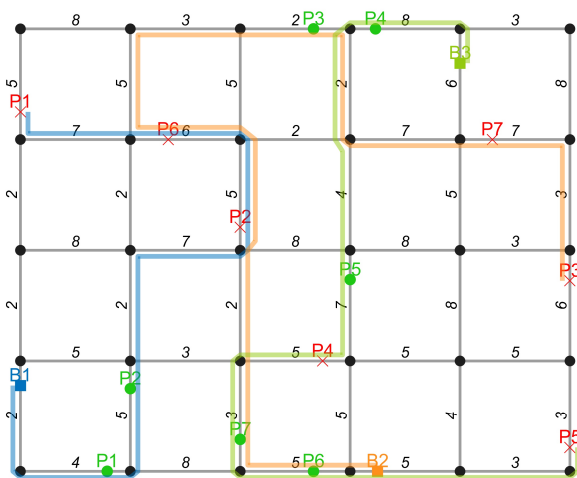
ahol  $v_{q_i} = dest(P_{last_i})$  a  $z_i$ -edik hangya utolsó utasának célpozíciója.

Az 4. ábra az algoritmus folyamatábráját mutatja.

Az egyéni feromontérkép lehetőséget ad arra, hogy a különböző buszpozíciókból induló feromonösvények ne befolyásolják egyidejűleg a hangyák élválasztását, így csak az adott buszhoz tartozó releváns információt dolgozzák fel. Ez gyorsítja az MMAS konvergálását az optimális megoldáshoz.

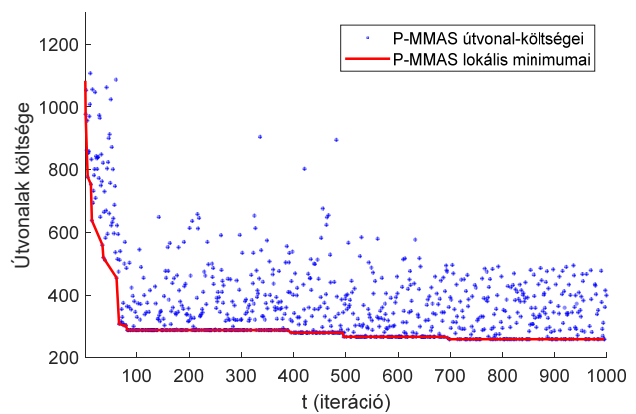
#### 4. TESZT EREDMÉNYEK

A P-MMAS algoritmust különböző komplexitású problémákon teszteltük. Az úthálózat költségeit, a buszok és utasok helyzetét random generáltuk. Az eredmények az 1. táblázatban láthatók, ahol közzétettük az adott problémán elért legjobb és az átlagos eredményt (lásd: (5) kifejezés), valamint az átlagos futási időt. A tesztet egy átlagos számítási kapacitással és teljesítménnyel bíró laptopon futtattuk a 2. táblázatban található paraméterekkel. A különböző problémákat sorszámmal láttuk el, megadtuk az úthálózat méretét, valamint a rajta elhelyezkedő buszok és utasok számát.



5. ábra: A 7-es problémára talált legjobb megoldás

Az 1. táblázatban különböző a sorszámmal ellátott problémák numerikus eredményei láthatók. Észrevehető, hogy a probléma komplexitásának növelése nagyobb futási idővel, valamint a legjobb útvonal és az átlagos útvonal költsége közötti különbség növekedésével jár.



6. ábra: P-MMAS konvergenciája

Az algoritmus konvergenciáját mutatja az 6. ábra a 7-es probléma esetében, ahol az összes iterációbeli hangya útvonal-költségét egy pont reprezentálja.

Láthatjuk, hogy a P-MMAS algoritmus konvergenciája exponenciális jelleget mutat. A paraméterek helyes megválasztása fontos, mert nagy mértékben befolyásolja a konvergenciát. Egész valószínű, hogy létezik a fenti paraméterkészlettel eltérő beállítás, mellyel az algoritmus gyorsabban konvergál, így javul az eredmény minősége és a futási ideje. Stütze és Hoos közreadtak egy irányelvet a cikkükben (Stütze & Hoos, mely segítõ kezét nyújt a paraméterek megválasztására, ám az útmutatás által nyújtott paraméterkészlet rosszabb teljesítménnyel rendelkezett, mint az általunk meghatározott paraméterszett.

#### 5. KONKLÚZIÓ

A dolgozat fő célja egy olyan közlekedési rendszer felvázolása, melyben a városi buszok kötött pályás közlekedés helyett szabadon mozognak az úthálózaton. A koncepció lényege, hogy a buszok közel optimális útvonalakon közlekedve szállítják el az utasokat a kezdőpontjuktól közvetlenül a céljukig, háztól-házig való szállítást megvalósítva. A feladat olyan úthalmaz megtalálása, mely mentén az utasok a legrövidebb idő alatt jutnak el a kívánt céljukhoz. Megadtuk a felvetés matematikai modelljét, melyben úthálózati gráfot feltételeztünk és egy módosított MMAS algoritmust, mely közel optimális megoldást nyújtanak. A P-MMAS algoritmus egy olyan általánosításai az eredetinek, melyek alkalmazhatók olyan környezetben, ahol a

probléma valamilyen több-kiindulópontú multi-ágens rendszerkoordinálás jelleget hordoz. Tipikusan ilyenek a DARP és VRP problémacsalád bonyolultabb variánsai. Az alacsony komplexitású esetekben kielégítő eredményeket kaptunk, azonban a magasabb dimenziójú problémák megoldásához további fejlesztés szükséges.

A P-MMAS legnagyobb hátránya a futási idő, melynek csökkentése a további fejlesztéseket tekintve központi jelentőségű. A paraméterek megfelelő megválasztásával egyértelműen javul az algoritmus teljesítménye, viszont ennek a meghatározása még további analíziseket igényel.

#### KÖSZÖNETNYÍLVÁNÍTÁS

Jelen BME által benyújtott tanulmány a Mesterséges Intelligencia Nemzeti Labor keretén belül az Innovációs és Technológiai Minisztérium támogatásával valósul meg.

EFOP-3.6.3-VEKOP-16-2017-00001: Tehetséggondozás és kutatói utánpótlás fejlesztése autonóm járműirányítási technológiák területén - A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

#### HIVATKOZÁSOK

Berbeglia, G., Cordeau, J.F. and Laporte, G., 2010. *Dynamic pickup and delivery problems*. European journal of operational research, 202(1), pp.8-15.

Braekers, K., Caris, A. and Janssens, G.K., 2014. *Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots*. Transportation Research Part B: Methodological, 67, pp.166-186.

Colomi, A., Dorigo, M. and Maniezzo, V., 1991, December. *Distributed optimization by ant colonies*. In Proceedings of the first European conference on artificial life (Vol. 142, pp. 134-142).

Cordeau, J.F. and Laporte, G., 2003. *A tabu search heuristic for the static multi-vehicle dial-a-ride problem*. Transportation Research Part B: Methodological, 37(6), pp.579-594.

Cordeau, J.F. and Laporte, G., 2003. *The dial-a-ride problem (DARP): Variants, modeling issues and algorithms*. Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 1(2), pp.89-101.

Cordeau, J.F., 2006. *A branch-and-cut algorithm for the dial-a-ride problem*. Operations Research, 54(3), pp.573-586.

Deti, P., Papalini, F. and de Lara, G.Z.M., 2017. *A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare*. Omega, 70, pp.1-14.

Dorigo, M., Birattari, M. and Stutzle, T., 2006. *Ant colony optimization*. IEEE computational intelligence magazine, 1(4), pp.28-39.

Garey, M.R. and Johnson, D.S., 1979. *Computers and intractability* (Vol. 174). San Francisco: freeman.

Huang, Y., Zhao, L., Van Woensel, T. and Gross, J.P., 2017. *Time-dependent vehicle routing problem with path flexibility*.

Transportation Research Part B: Methodological, 95, pp.169-195.

Jorgensen, R.M., Larsen, J. and Bergvinsdottir, K.B., 2007. *Solving the dial-a-ride problem using genetic algorithms*. Journal of the operational research society, 58(10), pp.1321-1331.

Kennedy, J., 2006. *Swarm intelligence*. In Handbook of nature-inspired and innovative computing (pp. 187-219). Springer, Boston, MA.

Park, J. and Kim, B.I., 2010. *The school bus routing problem: A review*. European Journal of operational research, 202(2), pp.311-319.

Parragh, S.N., Doerner, K.F. and Hartl, R.F., 2010. *Variable neighborhood search for the dial-a-ride problem*. Computers & Operations Research, 37(6), pp.1129-1138.

Parragh, S.N., 2011. *Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem*. Transportation Research Part C: Emerging Technologies, 19(5), pp.912-930.

Psaraftis, H.N., 1980. *A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem*. Transportation Science, 14(2), pp.130-154.

Rizzoli, A.E., Montemanni, R., Lucibello, E. and Gambardella, L.M., 2007. *Ant colony optimization for real-world vehicle routing problems*. Swarm Intelligence, 1(2), pp.135-151.

Ropke, S. and Pisinger, D., 2006. *An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows*. Transportation science, 40(4), pp.455-472.

Ropke, S., Cordeau, J.F. and Laporte, G., 2007. *Models and branch-and-cut algorithms for pickup and delivery problems with time windows*. Networks: An International Journal, 49(4), pp.258-272.

Stützle, T. and Hoos, H.H., 2000. *MAX-MIN ant system*. Future generation computer systems, 16(8), pp.889-914.

Tripathy, T., Nagavarapu, S.C., Azizian, K., Pandi, R.R. and Dauwels, J., 2017, September. *Solving dial-a-ride problems using multiple ant colony system with fleet size minimisation*. In UK Workshop on Computational Intelligence (pp. 325-336). Springer, Cham.

Toth, P. and Vigo, D. eds., 2002. *The vehicle routing problem*. Society for Industrial and Applied Mathematics.