

## KERESÉSI ALGORITMUSOK HEGESZTÉSI PONTSORREND OPTIMALIZÁLÁSHOZ

**Ternay Máté**

fejlesztő, Bay Zoltán Alkalmazott Kutatási Közhasznú Nonprofit Kft.  
3519 Miskolc, Iglói utca 2, e-mail: [mate.ternay@bayzoltan.hu](mailto:mate.ternay@bayzoltan.hu)

**Soltész Zsolt**

szenior fejlesztő, Bay Zoltán Alkalmazott Kutatási Közhasznú Nonprofit Kft.  
3519 Miskolc, Iglói utca 2, e-mail: [zsolt.soltesz@bayzoltan.hu](mailto:zsolt.soltesz@bayzoltan.hu)

**Árvai László**

osztályvezető, Rendszerfejlesztés osztály, Bay Zoltán Alkalmazott Kutatási Közhasznú Nonprofit Kft.  
3519 Miskolc, Iglói utca 2, e-mail: [laszlo.arvai@bayzoltan.hu](mailto:laszlo.arvai@bayzoltan.hu)

### **Absztrakt**

*A kutatás célja, hogy különböző útvonaltervező algoritmusok segítségével meghatározásra kerüljenek a hibás hegesztési pontkoordináták, ezáltal megkönnyíteni a hegesztési pontsorrend optimalizálást, mint folyamatot. Első lépésként feltérképezésre kerülnek a jelenleg széles körben alkalmazott útvonaltervező algoritmusok. Ebben a cikkben két útvonaltervező algoritmus kerül összehasonlításra, amik megkönnyíthetik egy pontsorrend optimalizálás során felmerülő hegesztési hibapontok megkeresését és meghatározását.*

**Kulcsszavak:** pontsorrend, ponthegesztés, optimalizálás, robottechnológia, autóipar

### **Abstract**

*The aim of the research is to determine the erroneous welding point coordinates using different route planning algorithms, thus facilitating the welding point sequence optimization as a process. As a first step, the currently widely used route planning algorithms will be mapped. This paper compares two route planning algorithms that may facilitate the search for and determination of welding error points during point sequence optimization.*

**Keywords:** point order, spot welding, optimization, robotics, automotive

### **1. Bevezetés**

A keresési algoritmusoknak számos fajtája és változata létezik, amivel optimalizálni tudunk egy-egy folyamatot. A kutatás során több cikk és publikáció is feldolgozásra került, amiket mérlegelve kerültek meghatározásra a számba vehető algoritmusok. A két kiválasztott algoritmus az ICP mátrix és a hangya kolónia algoritmus. A cikk a következőképp épül fel. A bevezetés után a 2. részben röviden áttekintjük a pontsorrend optimalizálást és néhány létező problémáját. A 3. fejezetben az algoritmusok bemutatására kerül sor és arra, hogy hogyan tudjuk őket alkalmazni egy hegesztési pontsorrend optimalizálási folyamat során. A 4. fejezet szemlélteti az algoritmusok összehasonlítását, majd a 5. fejezetben áttekinthetők az eredmények.

A hegesztési pontoptimalizálás, a XXI. századi gépgyártásban, azon belül pedig legfőképp az autógyártásban jelentős szereppel bír. Napjainkban egy autó legyártását főképp robotok végzik. A robotok hegesztik össze az elemeket, rakják össze az alkatrészeket és végeznek el minden olyan feladatot, amivel megkönnyítik és felgyorsítják a gyártási folyamatot. Azáltal, hogy a munka nagy részét robotok végzik, elengedhetetlen feladat a pontosság. A robotokat a lehető legpontosabban kell beprogramozni ahhoz, hogy egy autó az elkészülése után az emberek biztonságosan tudjanak használni.

Egy autógyártás során a hegesztési pontokat sok szempont szerint lehet optimalizálni. Lehet optimalizálni a hegesztési pontok minimális számára szerkezeti szilárdság megtartásával [1]. De ugyanilyen fontos a hegesztési paraméterek optimális értékének megtalálása [2].

A gépi tanulás gyors fejlődése forradalmasítja a robotikát. Az alkalmazásából eredő nyereséget és előnyt elsősorban az automatizált folyamatok és a magasabban képzett munkaerőnek köszönhető nagyobb rugalmasság biztosítja majd. Nőni fog a fogyasztás is, mivel a vevők több testreszabott, egyedi és magas minőségű terméket fognak vásárolni. A gépi tanulás módszerei felhasználhatók a hegesztési technológia optimalizálásához is [3].

Az ipari robotika egyre jobban élre tör Magyarországon is. A mesterséges intelligencia ötvözése nyomán megjelennek a hálózatba kapcsolt robotok. Ez a fejlesztés óriási lehetőséget rejt magában, ezért egyre több helyen alkalmazzák ezeket optimalizáció és robotvezérlés területén.

Az internetre kapcsolt szenzoroknak és a felhő alapú számítástechnika alkalmazásának köszönhetően hatalmas mennyiségű adatot képesek gyűjteni, elemezni, majd az eredményeket célzott cselekvésekké, beavatkozásokká alakítani. Ezek eredményeként kiküszöbölik a hiányosságokat, és a gyárak olyan teljesítményszinteket érnek el, amelyeket a múltban csak elméletben tartottak megvalósíthatónak. Jelenleg az ipari robotok csupán 2%-a kapcsolódik távoli monitoring központokhoz.

A forradalmi változások túlnyomórésze pozitív. A Negyedik Ipari Forradalom megnyitja az utat a robotika új aranykora felé, amelyben lehetővé válik majd a termelékenység javítása számos iparágban, és átalakul, könnyebbé válik a munkahelyi életünk.

## 2. Hegesztési pontsorrend optimalizálás

Ahhoz, hogy egy hegesztési folyamatot optimalizálni tudjunk, szükségünk van először magát a teljes hegesztési folyamatot megvizsgálni és kiszűrni azon tényezőket, amik lassíthatják az egyes folyamatokat. Több rendszer átvizsgálásával kialakítottunk olyan folyamatot (lépéseket), amik hatékonyan segítik a hegesztési pontsorrend optimalizálást. A lépések a következők:

1. **Mérnöki koordináta rendszer adatainak feldolgozása külön adatbázisban:** A bemeneti (input) adatokat, amelyek a mérnöki koordináta rendszerből származnak általában excel, pdf, CAD illetve egyéb állományokban történnek tárolásra és vannak jelen. Egy automatikus konvertálást lehetővé program segítségével, viszont fel tudjuk ezen adatokat egyszerre dolgozni és importáljuk egy adatbázisba a későbbi könnyebb és egyszerűbb adatfeldolgozás érdekében.
2. **Robotprogram / robot koordináta rendszer:** Robotprogramok kinyerése VPN kapcsolaton keresztül a próbacella robotokból, valamint robot koordináták előkészítése és matematikai algoritmus kidolgozás ezek számára. Robotprogramok letöltése a robotokról. Az itt kapott információkat ugyanazon az elvre alapozva, mint a mérnöki koordináta rendszer adatainak az összegyűjtését ellenőrizzük, feldolgozzuk és egyszerűsítjük későbbi adathasználatra. Az itt kinyert robotkoordinátákat JSON formátumban rendezzük és átkonvertáljuk a szükséges formátumra.
3. **Koordináta rendszerek közötti transzformáció:** Egy ICP / Hangya kolónia algoritmus használata, amely lehetővé teszi a mérnöki- és a robot- koordináta rendszerbeli pontfelhők összehasonlítását

és vizsgálatát. A két különböző koordináta rendszerben található ponthalmaz megfeleltetéséhez szükséges transzformáció meghatározása egy mátrix-al történik.

4. **CAD állományokban történő adatmegjelenítés és adat lekérdezés:** Minden információ egy az adott alkatrészekről, valamint robotokról nem csak egy adatbázisban van letárolva, hanem egy CAD állományban is. Ezen állománykezelő programok számára kifejlesztett belső modul lehetővé teszi, hogy egy adott alkatrésztől és a hozzá tartozó elemekről részletesebb információkat kapjunk vizuálisan is.
5. **Hegesztési pontok, valamint hibás hegesztési pontok keresése és hibajelölése:** Egy API segítségével a CAD állományok feldolgozása, hegesztési pontok azonosítása, attribútumaik kezelése. Az alkalmazás célja a hibás hegesztési pontok keresésének megoldása, elősegítése a mérnökök és az operátorok számára. A hibakereső alkalmazás egy adott modellekhez tartozó adatokat összehasonlítja az adatbázisban lévő adatokkal, majd betölti ezeket. Vizualizálva láthatjuk és követhetjük nyomon, hogy hol tart a hegesztés és mely alkatrészeknél hol fordulhatnak elő hibák.

### 3. Algoritmusok

A mérnöki és a robot koordinátarendszer összekapcsolódásához szükséges volt egy olyan kereső algoritmus, amivel könnyen meg lehet határozni a két rendszer pontpozícióit. Egy útvonaltervezés problémájára számos algoritmus létezik, ezek közül az ICP mátrix és a Hangyakolónia algoritmus került kiválasztásra, mivel ezek tűntek számunkra a lehető legígéretesebbnek eredmények terén [4]. A következő rész ezen algoritmusok általános bemutatására kerül sor.

#### 3.1. ICP mátrix

Az ICP (Legközelebbi iteratív pont) algoritmus a „de facto” szabvány szerint háromdimenziós geometriai modellek összehasonlítása szolgál, ugyanis ennek segítségével meg tudjuk határozni a kezdeti becsült pozíciót.

Mindegyik ICP iterációban a transzformáció kiszámítható az alábbi négy módszer bármelyikével:

1. SVD és Arun metódus alapján,
2. Horn négyes metódus,
3. Horn - ortonormális mátrixot használó algoritmus,
4. Walker – féle duális négyeseken alapuló számítások.

Az ICP algoritmusokban általában hiba keletkezik, ha túl sok pontot választunk jellegtelen területről. Ebben az esetben az algoritmus lassan konvergál, rossz pozíciót talál, vagy pedig divergál.

Azonban ezek a korszerű ICP variánsok mind azt feltételezik, hogy a bemeneti adatokat hálóként adják. Sok alkalmazási forgatókönyvben nem áll rendelkezésre háló, például a 3D-s adatoknál a robotikában. Itt a mérések tartalmazzák a Gauss-zajt. Ráadásul a robotikában ezeket gyakran szétszórja a szenzor. Ebből a két okból nem lehet alkalmazni a megszerzett pontok topológiáján alapuló egyszerű hálórendszert.

Az ICP algoritmusok sebességének javítása az utóbbi időben nagy figyelmet kapott. A kidolgozott módszerek széles skálája arra törekszik, hogy növelje a pontszámítási teljesítményt. A jelenleg rendelkezésre álló módszerek közé tartozik például a háromszög egyenlet és a többfelbontáson alapuló heurisztika. Kutatásunk során az ICP által a k-d fákat használjuk a keresés felgyorsítására. Ez a tanulmány egy új keresési eljárást ismertet, nevezetesen a cachelt k-d fákat, ICP algoritmus iteratív viselkedését kihasználva. Ez jelentős optimalizálást eredményez (körülbelül 50%).

Ez az ICP változat pont-sík hibamérettel és vetítési alapú módszerrel a pont illesztésnek kiválasztásában segít. Továbbá arra a következtetésre jutottunk, hogy az ICP-folyamat többi szakasza kis hatással van a konvergenciaszintre, így amit számításba vettünk az a véletlenszerű mintavétel, az állandó súlyozás és a pontpárok távolságának küszöbértéke.

Minden egyes iterációs lépésnél az algoritmus kiválasztja a legközelebbi pontokat és kiszámítja a transzformációt, azaz forgatást és fordítást ( $\mathbf{R}$ ,  $\mathbf{t}$ ) az egyenlet minimalizálása céljából.

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{Nm} \sum_{j=1}^{Nd} w_{i,j} \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2, \quad (1)$$

ahol  $N_m$  és  $N_d$ , az  $M$  modellkészletben lévő pontok száma és  $D$  adatkészlete, és  $w_{ij}$  a súlyok egy pontegyezéshez. A súlyok a következőképpen kerülnek hozzárendelésre:  $w_{ij} = 1$ , ha  $m_i$  a  $d_j$ -hez legközelebbi pont,  $w_{ij} = 0$  egyébként. Eq. (1) csökkenthető a következőre:

$$E(\mathbf{R}, \mathbf{t}) \propto \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\|^2, \quad (2)$$

$$N = \sum_{i=1}^{Nm} \sum_{j=1}^{Nd} w_{i,j},$$

mivel a megfeleltetési mátrixot egy  $v$  vektor ábrázolja, amely a pontpárokat tartalmazza, azaz  $v = ((d_1, mf(d_1)), (d_2, mf(d_2)), \dots, (d_{Nd}, mf(d_{Nd})))$ , és az  $f(x)$  a keresési funkció visszatér a legközelebbi pontra. A feltételezés az, hogy az utolsó iterációs lépésben a pontok megfeleltetése, tehát a pontpárok vektora helyes.

A javasolt cachelt  $k$ -d fa keresésnek  $O((I + \log N_m) N_d)$  időre van szüksége a legjobb esetben. Ez a teljesítmény akkor érhető el, ha a konstans idő a visszakereséshez  $N_d \log N_m$  időt eredményez a fa létrehozásához, és  $I \cdot N_d$  a kereséshez abban az esetben, ha nincs szükség visszakeresésre. Nyilvánvaló, hogy a visszakeresési idő függ a számított ICP transzformációtól ( $\mathbf{R}$ ,  $\mathbf{t}$ ), ahol az idő közel állandó.

### 3.2. Hangyakolónia algoritmus

A hangyakolónia algoritmus alapjait Marco Dorigo fogalmazta meg [5]. Kutatásai során megfigyelte a hangyák élelemszerzési módszerét, melynek lényege, hogy a hangyák teljesen véletlenszerű úton indulnak el a bolyból élelmet keresni. Ahol élelmet találnak, a visszafele úton feromont bocsájtanak ki. A feromon a többi hangya számára vonzó tulajdonságú, ezért amikor egy újabb hangya indul élelemért, akkor nagyobb valószínűséggel indulnak el azon irányba, ahol előzőleg egy hangya feromont bocsájtott ki. Minél több hangya halad el az adott útvonalon, annál erősebb az útvonalon a feromon nyom, ezáltal több hangya választja majd ezt az útirányt. Emellett minél közelebb van az adott élelemforrás, annál többször tudnak megfordulni a hangyák, ami szintén a feromon nyom erősödését vonja maga után [6] [7]. Ennek mintájára nem csak egy hegesztési pontok megtalálását, hanem magát a hegesztési munkafolyamatot is optimalizálni lehet a hangyakolónia algoritmus segítségével [8].

Az ACS (Ant Colony System) [9] három fő szempontból különbözik az előző hangyarendszertől:

1. Állapotátmenet szabálya közvetlen utat biztosít az új szegélyek feltárása, valamint a problémára vonatkozó prioritás és felhalmozott tudás kiaknázása között,
2. Globális frissítési szabály csak azokra az élekre alkalmazták, amelyek a legjobb hangya-útvonalakhoz tartoznak,

3. Amíg a hangyák megoldást állítanak fel, egy helyi feromonfrissítési szabályt (röviden helyi frissítési szabályt) alkalmaznak.

Az antkolónia optimalizálása (ACO) az SI meta-heurisztika családja, amely utánozza a valódi hangyák kollektív viselkedését [10]. Kommunikációjukban a hangyák közötti kommunikáció a feromon nyomvonalak segítségével valósul meg, amíg a hangyák élelmet keresnek.

```
for (int index = 1; index <= graph.numberCities; index++)
{
    graph.edges[bestTour.data[index], bestTour.data[index + 1]].pheromone =
    (1.0 - GLOBAL_PHEROMONE_UPDATE) *
    graph.edges[bestTour.data[index],
    bestTour.data[index + 1]].pheromone +
    GLOBAL_PHEROMONE_UPDATE * 1.0 /
    bestTour.cost;
}
```

1. ábra: Hangyakolónia - legjobb útvonal számítási részlet

Mivel ezek a nyomok idővel elpárolognak, a kolóniától az élelmiszer-forrásig tartó legrövidebb út több hangyát vonz, mert nagyobb mennyiségű feromonnal rendelkezik. Mikor az Ant System-et kifejlesztették, akkor alkalmazták az utazó ügynök problémára is. Kezdetben három különböző verzió volt az AS-ből: ant-density, ant-quantity, és ant-cycle. Míg az ant-density és az ant-quantity esetében a hangyák egyből frissítik a feromont amint átmennek egy városból egy szomszédos városba, addig az ant-cycle esetében a feromonokat csak azután frissítik, hogy minden hangya megkonstruálta a saját útját és elhelyezte rajta a feromont az út minőségétől függően. Mivel az ant-cycle jobban teljesített, mint a másik kettő, ezért ezt mutatom be és a továbbiakban Ant System (AS) néven hivatkozok rá. Az AS-ben mindig m darab hangya létrehoz egy megoldást, egy utat. A helyi keresést nem alkalmazzuk [11][12]. Hegesztési pontsorrend optimalizálás során, amikor megtaláltuk a mérnöki koordináta rendszerhez tartozó robotkoordinátákat szelektálnunk kellett és azok közül is megtalálni a legközelebbi pontot

```
int shortestIndex = 0;
double shortest = Double.MaxValue;
for (int index = 1; index <= n; index++)
{
    if (distances[index] < shortest)
    {
        shortestIndex = index;
        shortest = distances[index];
    }
}
```

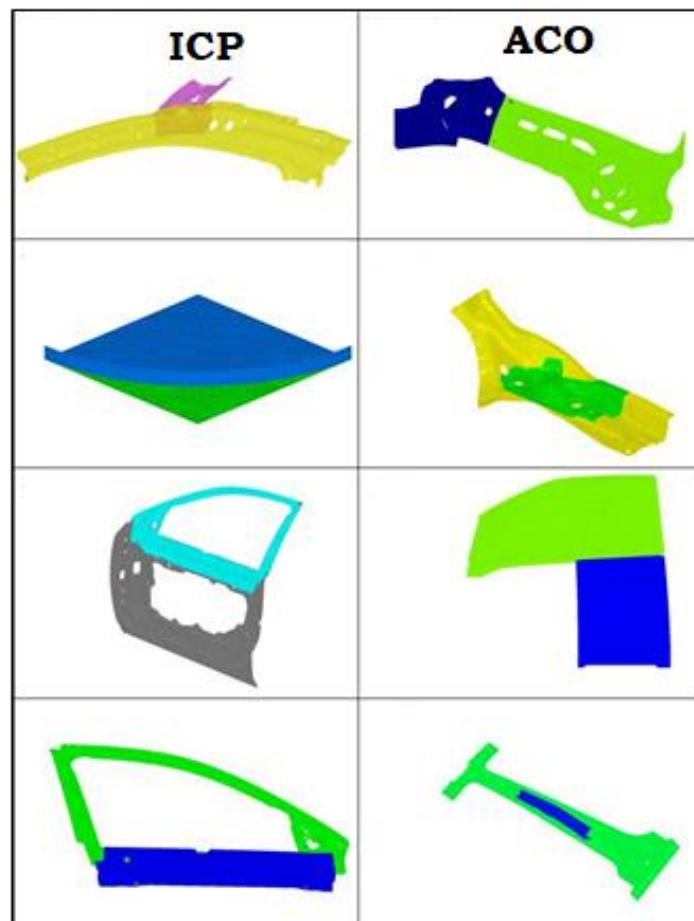
2. ábra: Legrövidebb pont-pont közötti távolságszámítás

A hangyakolónia algoritmust segítségével nem csak a mérnöki és a robot koordináta rendszeren lévő koordináták pontjait tudjuk meghatározni könnyen, hanem segítségre lehet a tűrési érték kiszámításánál is, ugyanis számos hegesztési pontnál tapasztalhatunk sok olyan pontot, ami a tűrészatháron kívülre esik. Így amikor megkapjuk a mérnöki pont koordinátáit és megvizsgáljuk az ACO algoritmussal, hogy mely

mérnöki pontokhoz mely robotpontok találhatóak közel akkor megkaphatjuk azon tûrési értékeket, amik TL(X-tengely), BL(Y-tengely), WL(Z-tengely) – pontoknál keletkezik.

#### 4. Összefoglalás

Hegesztési pontsorrend optimalizálás keretein belül szükséges a mérnöki és a robot koordináta rendszer pontjainak összekapcsolásához egy olyan keresési algoritmus, amely segít megtalálni a ponthegesztéseket a két koordináta rendszer között. Ez a folyamat ugyanis elengedhetetlen, mivel a mérnöki koordináta rendszerben a hegesztési pontkoordináták teljesen más helyen szerepelnek egy adott alkatrészben vagy modellen., mint a robot koordináta rendszerben, így nagyon könnyen találkozhatunk ponthegesztési hibákkal. Ahhoz, hogy mind a ponthegesztés és mind az alkatrészek / modellek összekapcsolása pontos legyen a legkisebb távolsági különbséget is meg kell találnunk a két koordináta rendszer között. Ennek segítségére az ICP mátrixot vagy pedig az ACO algoritmust tudjuk alkalmazni. Mindkét algoritmust fel tudjuk használni, mivel a kifejlesztett optimalizációs rendszerünkhöz mindkettő megfelelően tud alkalmazkodni. A 3. ábrán néhány helyes összekapcsolás látható a modellek között, amelyek összeillesztési pontjait az ICP és az ACO algoritmus segítségével határoztunk meg.



3. ábra: Modellek összeillesztése

## 5. Köszönetnyilvánítás

A cikkben ismertetett kutatómunka a Bay Zoltán Alkalmazott Kutatási Nonprofit Kft. által valósulhatott meg.

## Irodalom

- [1] Ouisse, M., Cogan, S., Huang, S.-J., Guillaume, B. *An efficient procedure for spot welds optimization in automotive structures*. 2006, IMAC-XXIV: Conference & exposition on structural dynamics, St Louis, 30 Jan-2 Feb 2006
- [2] Raut, M., Achwal, V. *Optimization of Spot Welding Process Parameters for Maximum Tensile Strength*, International Journal of Mechanical Engineering and Robot 2014 3(49):506-514.
- [3] Hamed, M., Shariatpanahi, M., Mansourzadeh, A. *Optimizing spot welding parameters in a sheet metal assembly by neural networks and genetic algorithm*, Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture 2007, 221(7):1175-1184. <https://doi.org/10.1243/09544054JEM476>
- [4] Islam, M., Buijk, A., Rais-Rohani, M., Motoyama, K. *Simulation-based numerical optimization of arc welding process, for reduced distortion in welded structures* Center for Advanced Vehicular System, Mississippi State University, USA, 2014. <https://doi.org/10.1016/j.finel.2014.02.003>
- [5] Bader Al-Saleh, A., Bachir Menai, M. *Ant Colony System for Multi-Document Summarization*, Proceedings of the 27th International Conference on Computational Linguistics 2018, pp.734-744.
- [6] Dorigo, K., Gambardella, L.M. *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, Université Libre de Bruxelles, Belgium, 1996.
- [7] Katona, G. *Útvonaltervező algoritmusok*, Közlekedéstudományi szemle 2016, 66(1):35-45.
- [8] Dorigo, M. *Optimization, learning and natural algorithms (in Italian)*, Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1992.
- [9] Bella, J.E., McMullen, P.R. *Ant colony optimization techniques for the vehicle routing problem*, Advanced Engineering Informatics 2004, 18(1):41-48. <https://doi.org/10.1016/j.aei.2004.07.001>
- [10] Bonabeau, E., Dorigo, M., Theraulaz, G. *Swarm intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999. <https://doi.org/10.1093/oso/9780195131581.001.0001>
- [11] Dorigo, M., Gambardella, L. M. *Ant colony system: A cooperative learning approach to the travelling salesman problem*, IEEE Transactions on Evolutionary Computation 1997, 1:53-66. <https://doi.org/10.1109/4235.585892>
- [12] Dorigo, M., Stützle, T.: *Ant Colony Optimization*, Ant Colony Optimization, MIT Press, 2004. <https://doi.org/10.7551/mitpress/1290.001.0001>