

A Formula-1-es versenynaptár optimalizációja genetikus algoritmus segítségével

Optimization of the Formula-1 race calendar using genetic algorithm

Polyák, G., Póth, M.*

Faculty of Technical Sciences, University of Novi Sad, Serbia

* Subotica Tech –Collage of Applied Sciences

gabriellapolyak8@gmail.com

pmiki@vts.su.ac.rs

Összefoglalás — Genetikus algoritmusok alatt olyan keresési technikák egy csoportját értjük, melyekkel optimumot vagy egy adott tulajdonságú elemet lehet keresni. A genetikus algoritmusok speciális evolúciós algoritmusok, technikáikat az evolúcióbiológiából kölcsönözték. Számos alkalmazási területe van, azaz számos problémátípust meg lehet oldani a segítségével. A munka célja egy optimalizálási probléma megoldása genetikus módszerrel. Tekinthejtük ezt egy Az utazó ügynök problémához hasonlóknak, ugyanis útvonal optimalizációról van szó. Alapul a Formula-1-es versenynaptár, illetve a nagydíjhelyszínek földrajzi elhelyezkedésének koordinátái szolgálnak. Az optimalizálás két módon történt, a Microsoft Excel egyik bővítményével, az Excel Solverrel, illetve Matlabban íródott programban. Az Excel Solver működési elve, hogy a megszorítások figyelembevételével megvizsgálja a lehetséges megoldásokat, majd azok közül kiválasztja a számunkra legkedvezőbbet. A Matlabban írt program a genetikus algoritmus során használt módszereket alkalmazza, mint amilyen a szelekció, mutáció, keresztezés és a visszahelyezés. A munkában az említett két programban kapott eredmények kerülnek összehasonlításra és összegzésre.

Kulcsszavak: útvonal-optimalizáció, genetikus algoritmus, Formula-1, problémamegoldás

Abstract — Genetic algorithms are a group of search techniques, which can be used to search for an optimum or an element with a given property. Genetic algorithms are special evolutionary algorithms, borrowing their techniques from evolutionary biology. They have many fields of application, i.e. many types of problems can be solved with their help, such as e.g. subset selection, structure development, technical design, etc. The aim of this work is to solve an optimization problem using the genetic method. This could be considered similar to the Traveling salesman problem (TSP), as the topic is route optimization. The study is based on the Formula 1 race calendar and the geographical location of the grand prize venues. The optimization was done in two ways, with an add-on of Microsoft Excel, Excel Solver, and a program written in MATLAB. Excel Solver is based on: examining the possible solutions taking into account the constraints, then selecting the one that fits the most. The program written in MATLAB uses methods used in the genetic algorithm, such as selection, mutation, crossing, and

insertion. In this work, the results obtained in the two programs are compared and summarized.

Keywords: route optimization, genetic algorithm, Formula-1, problem solving

1 BEVEZETÉS

Rohamosan gyorsuló és fejlődő világunkban egyre összetettebb problémákba ütközünk. Váratlan fordulatok akármikor történhetnek, a megoldások pedig nem várathatnak magukra.

A munka célja egy optimalizálási probléma megoldása evolúciós módszerrel. Tekinthejtük ezt egy Az utazó ügynök problémához (Traveling Salesman Problem, TSP) hasonlóknak, ugyanis útvonal optimalizációról van szó. Alapul a Formula-1-es versenynaptár, illetve a nagydíjhelyszínek földrajzi elhelyezkedésének koordinátái szolgálnak.

A 2021-es évi végleges versenynaptár került optimalizálásra, először a Microsoft Excel beépített alkalmazásával, a Solverrel, majd pedig Matlabban írt programmal, ami az evolúciós algoritmus módszereit használja, mint amilyen a szelekció, mutáció, keresztezés és a visszahelyezés. Mivel a Solver egy beépített funkció így nem lehet tudni, hogy valójában mi történik miközben a program fut, csak a bemeneti adatokat lehet ismerni, illetve miután lefutott a program, a kimeneti adatokat, azaz az eredményeket. Ez volt az oka annak, hogy egy program lett megírva Matlabban, hogy tudjuk pontosan hogyan is történik az optimalizáció. Miután mindkét módszerrel megtörtént az optimalizáció, az eredmények kielemezésére került sor, majd pedig a kapott adatok kerültek összehasonlításra.

Ugyanazzal a módszerrel a Formula – 1 1950 óta íródó történelmében, azaz az összespálya közötti optimális útvonal is kiszámításra került.

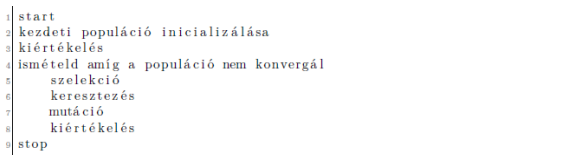
Mivel a 2020-as szezon elég kaotikusra sikerült, a szervezők igyekeztek enyhíteni ezt a 2021-es szezonra, azonban ismét elég sok problémába ütköztek. A 2020-as szezonhoz hasonlóan ismét több futam megrendezése is elhalasztásra került, illetve le lett mondva. Végül a 2021-es szezon 21 nagydíjhelyszínből és 22 megrendezett futamból

állt össze. Az ausztriai Red Bull Ringen 2 nagydíj is lett szervezve Osztrák, illetve Stájer Nagydíj néven.

2 GENETIKUS ALGORITMUSOK

A biológiai evolúcióban az élőlények, mint komplex rendszerek fejlődését figyelhetjük meg. Az élőlények evolúciója egy olyan folyamat, melynek során az élőlények megtanulnak változó környezetükhöz alkalmazkodni. Ez az alkalmazkodás nem egyformán sikeres minden élőlény számára, így egyre változatosabb formák mellett szelekciónak is szemtanúi vagyunk. Darwin (1859) „A fajok eredete” című műve óta ez az evolúciós folyamat egyre inkább a kutatás tárgyát képezi.

A genetikus algoritmusok heurisztikus keresési megközelítések, amelyek az optimalizálási problémák széles skálájára alkalmazhatók. Ez a rugalmasság a gyakorlatban kedvezővé teszi őket számos optimalizálási probléma szempontjából [6]. A genetikus algoritmus modellje természetesen távol áll a tényleges biológiai evolúciós folyamatától, csak néhány fogalom, technika állítható párhuzamba, mint amilyen a populáció, egyed, szelekció, mutáció, szaporodás, és a tényleges élőlények generációi helyébe a folyamatosan változó populációk sora lép, az 1. ábrán a genetikus algoritmus általános felépítése látható.

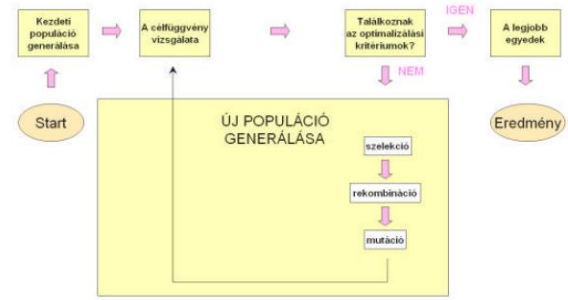


1. ábra - A genetikus algoritmusok általános felépítése

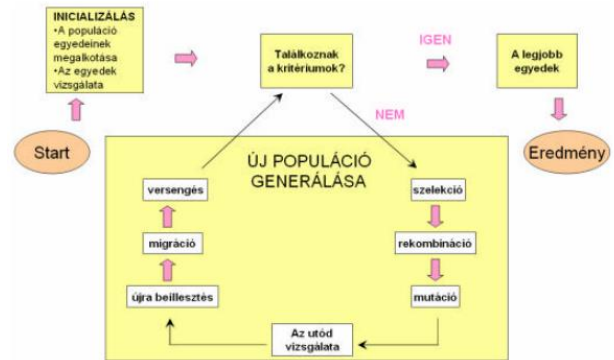
Bár a genetikus algoritmus születését egyértelműen a biológiai evolúció motiválta, a kialakított modell más jellemzőit is kiemelhetjük. Tekinthejtük kereső eljárásnak, tanuló algoritmusnak, vagy populációs alapú algoritmusnak, amelyben biológiai analógiák, vagy információcsere jellemzők alapján képezzük egyes műveleteket. E megközelítés alapján az evolúciós algoritmust a mesterséges intelligencia probléma független kereső eljárásai, vagy a tanuló algoritmusai közé sorolhatjuk, vagy akár az operációkutatás populáció alapú algoritmusának is tekinthetjük [1].

2.1 A genetikus algoritmusok működési elve és módszerei

Az evolúciós algoritmus a probléma megoldását egy evolúciós eljárással keresi. Az eljárás ciklikusan működik, minden ciklusban egyidejűleg több egyeddel (megoldással) dolgozva. Egy populációból válogatással, majd különböző kereső műveletekkel állít elő újabb utódokat, a 2. ábrán egy egyszerű genetikus algoritmus szerkezeti felépítése látható, míg a 3. ábrán egy összetettebbé [5]. Az egyes egyedek rátermettségét a fitnessfüggvénnyel méri, amely a rátermettebb egyedekhez nagyobb (kisebb) számot rendel, mint a kevésbé rátermett egyedekhez. Minden iteráció végén értékeli az egyedeket és utódokat, vagy csak az utódokat, és újabb populációt alakít ki belőlük. A ciklusok, azaz generációk mindaddig ismétlődnek, amíg egy kívánt feltétel nem teljesül (pl. a generációk száma elér egy kívánt értéket).



2. ábra - Egy egyszerű genetikus algoritmus szerkezeti felépítése



3. ábra - Egy összetettebb genetikus algoritmus szerkezeti felépítése

2.1.1 Egyed és populáció

Az algoritmus egy kezdeti egyedhalmazzal indul, amelyet populációnak nevezünk. Mindegyik egyed potenciális megoldás a problémára. Az egyedek pedig génekből állnak, általában bináris reprezentációval. Mivel a módszer egyszerre több egyeddel „kezel”, így a globális optimum megtalálásának esélye megnő. A populáció az egyedek fejlődésén keresztül fejlődik. Tehát minél fittebbek az egyedek, annál közelebb kerül a populáció is a feladat (egy) megoldásához.

2.1.2 Fitness függvény

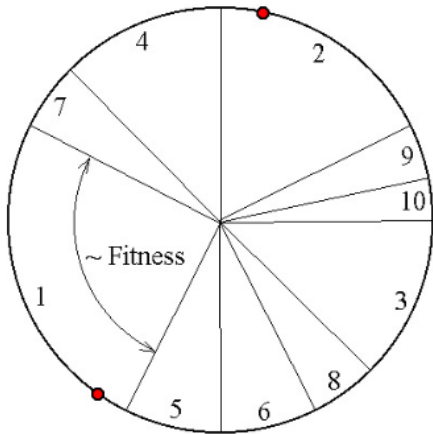
Az egyedek rátermettsége a fitness függvény segítségével értékelődik ki, vagyis számszerűsíti azt, hogy mennyire fitt (mennyire versenyképes). A függvény mindig problémafüggő.

2.1.3 Szelekció

A szelekció, vagy kiválasztás folyamata alatt azt a folyamatot értjük, amikor is egyedeket választunk ki a populációból, hogy azok tulajdonságait tovább örökössük a következő generációba [8]. Két, a fitness értékeik alapján kiválasztott egyed alkot egy szülőpárt. Minél nagyobb a fitness értéke egy egyednek, annál nagyobb eséllyel kerül kiválasztásra, de kevés esélyt gyengébben teljesítő egyedek is kapnak [7].

Többfajta szelekciós technikáról beszélhetünk:

- Rátermettség arányos szelekció (roulette wheel selection vagy fitness proportional selection, 4. ábra): Amikor is a nagyobb fitness értékű egyedek nagyobb eséllyel kerülhetnek kereszteződésre, de esélyt kaphatnak a gyengébben teljesítő egyedek is, bár jóval kisebb valószínűséggel (eséllyel). Ez esetben a teljes populációból választunk véletlenszerűen az arányokat figyelembe véve. Illetve fontos megjegyezni azt is, hogy egy egyed többször is kiválasztásra kerülhet.



4. ábra - Rulettkerék szelekció

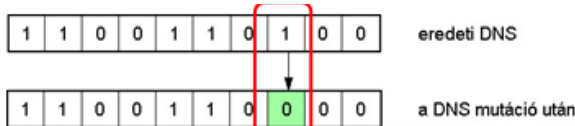
- Párok versenyztetése (tournament selection): Amikor véletlenszerűen kiválasztott egyedek közül az kerül keresztezésre, amelyiknek nagyobb a fitness értéke.
- Rangsorolás (rank selection): Abban az esetben használják, amikor az algoritmus az optimum köré gyűlik, így minden egyes egyednek relatíve egyforma szelet jut a rulet keréken. Így a rátermettség helyett a rátermettségi rangsor szerint osztják el arányosan a lehetőséget a kiválasztódásra.
- Elitista szelekció (elitism selection): Elitista szelekciónról beszélünk, ha a legrátermettebb egyed(ek) mindig átkerül(nek) változatlanul az új populációba. Garantálva ezzel, hogy az eddig megtalált legjobb megoldásnál rosszabbul nem tud teljesíteni a teljes populáció.

2.1.4 Mutáció

Mutáció esetén véletlenszerűen megváltoztatunk egy egyed egy-egy génjét, egy példa az 5. ábrán látható. Lehet ez:

- Bitek 0-ról 1-re való cseréje és fordítva
- Gén cseréje a felső és alsó határ között egyenletes eloszlással generált véletlen számmal
- A kiválasztott génhez hozzáad egy Gauss eloszlású véletlenszerű értéket; ha ez kívül esik az adott gén megadott alsó vagy felső határán, akkor az új génértéket levágják a határnál

Kezdetben a mutációk az állapottér hatékony feltárását eredményezik, a későbbiekben a lokális minimumokból való kiküszöbölés a szerepük. A globális optimum közelében tartózkodó populáció esetében azonban kérészetűek, hiszen kevésbé lesznek fittek társaiknál.



5. ábra – Mutáció

2.1.5 Keresztezés

Keresztezés vagy rekombináció alatt értjük azt a folyamatot, amikor két kiválasztott egyed valamilyen „módszer” alapján egyesítjük. Minden egyes szülőpár esetén két utódot hozunk létre. A genetikus algoritmus előnye főleg a keresztezésből származik, így ez a rész van kerül a legrészletesebb ismertetésre.

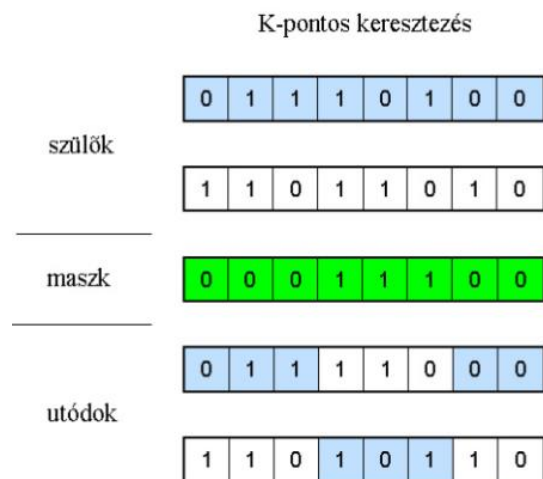
A kereszteződés történhet többféleképpen, mindig a problémához kell igazítani a módszert:

- Egyponos keresztezés – A szülőket két részre osztja, és ezeket a részeket cseréli fel egymással, példa a 6. ábrán látható.



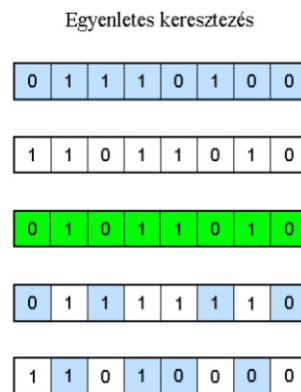
6. ábra - Egyponos keresztezés

- K-pontos keresztezés – A szülőket k+1 részre osztja, majd ezekből hozza létre az utódokat, példa a 7. ábrán látható.



7. ábra - K-pontos keresztezés

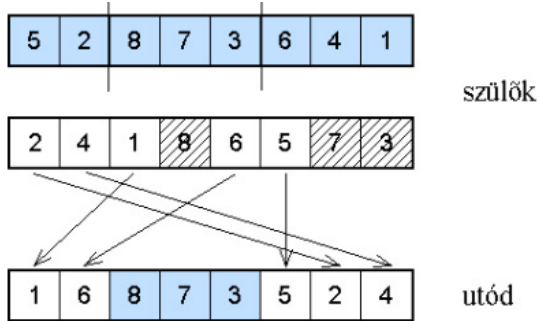
- Egyenletes keresztezés (uniform crossover) – Minden pont meghatározott valószínűséggel származhat, egyik vagy másik szülőtől, példa a 8. ábrán.



8. ábra - Egyenletes keresztezés

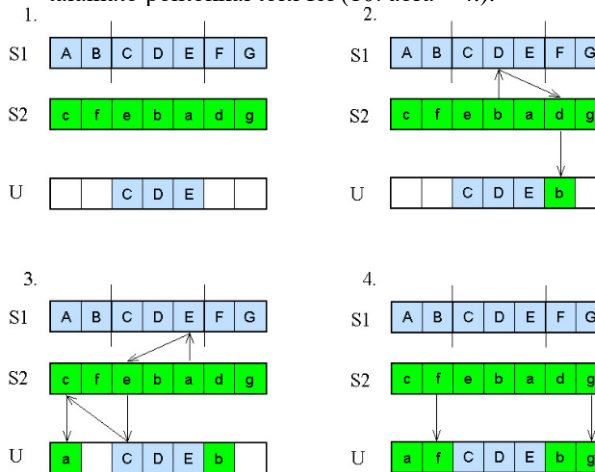
- DAVIS sorrendi keresztezés (Order crossover, OX) – Az első permutáció véletlenszerűen kiválasztott szakasz (8,7,3) átmásolásra kerül az utód ugyanazon

részére. A maradék helyek a második permutációból (szülőből) töltődnek fel, a sorrendet betartva, de a már felhasznált pontokat kihagyva, példa a 9. ábrán található. Az utód az első szülőtől relatív sorrendi, abszolút helyzeti és szomszédsági, míg a másodiktól csak relatív sorrendi információkat örököl.



Ábra 9. ábra – Order crossover (OX)

- Részletes megfeleltetésű keresztezés (Partially mapped crossover, PMX) – Ezt az operátort Goldberg és Lingle alkották meg 1985-ben. Az eljárás a sorrendi keresztezésnél megismert módszertől abban különbözik, hogy a második szülőtől származó pontokat másképp helyezi el (10. ábra). Az S1 szülőtől átmásolt CDE részlet után (10. ábra – 1.), először azokat a pontokat helyezi el az U utódban, melyek még nem szerepelnek benne és az S2 szülőnél azokon a helyeken vannak, melyeket az utódnál már betöltöttünk, ezek a b és az a pontok. A b pont elhelyezése úgy történik, hogy megnézi melyik pont van az S1-ben a helyén, ez a D pont, ezután arra a helyre teszi a b pontot, amit a d foglal az S2-ben (10. ábra – 2.). Az a pont elhelyezése: a helyén S1-ben E van, de e helyén U-ban már C, ezért tovább kell folytatni a módszert: c helye U-ban még üres, tehát ide kerül a (10. ábra – 3.). A maradék helyeket az S2-ből ugyanazon a helyen található pontokkal tölti fel (10. ábra – 4.).

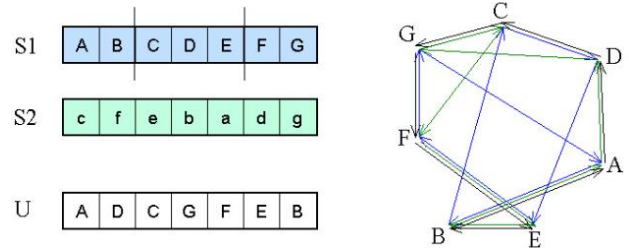


10. ábra - Partially mapped crossover (PMX)

- Order crossover 2 - Ez a módszer véletlenszerűen kiválaszt K pontot az első szülőből, megkeresi ezek helyét

a másodikban, és a K pontot a második szülőben elfoglalt helyekre, de az első szülőben meghatározott sorrendben, helyezi el az utódban. A maradék pontokat a második szülőből közvetlenül másolja át az utódba.

- Élkeresztesítés - Az élkeresztesítő eljárás az utazó ügynök problémához készített speciális operátor. Az algoritmus a permutációk szomszédsági tulajdonságait veszi figyelembe, mert ez határozza meg egy útvonal hosszát. Az eljárás először kifejezi a szülőkből a szomszédsági információkat, és tárolja őket egy táblázatban, tulajdonképpen egy mindkét szülő éleit tartalmazó gráfot rajzol fel. Ezután a táblázat (gráf) alapján egy új egyed (útvonal) hoz létre. Az új egyed létrehozó eljárás többször is zsákutcába juthat az útvonal keresése közben – a 11. ábrán egy lehetséges utódot mutat [4].



11. ábra – Élkeresztesítés

2.1.6 Visszahelyezés

Az utódok létrehozása után ki kell alakítani az új populációt, amelyben már az utódok is helyet kapnak. Az új populáció többféleképpen állítható elő. Az utódok lecserélhetik az előző populációt, valamilyen arányban az előző populáció egyedeiből és az utódokból fog állni, vagy pl. a fitnessfüggvény alapján dönti el az algoritmus, melyek lesznek az új populáció egyedei [1].

2.2 Alkalmazási területek

Hogyha megértjük az adaptív rendszerek működését, ez alapján mesterséges adaptív rendszerek fejleszthetők [2].

Adaptáció olyan folyamatot értett, amely valaminek a struktúráját állandóan módosítja, ezzel környezetében egyre jobb hatásfokot ér el. Adaptív rendszerként a biológiai evolúciót vizsgálta, amely az alkalmazkodást azzal teszi lehetővé a változó környezeti feltételekhez, hogy állandóan változó genetikai struktúrákat generál, és örökléssel terjeszti őket.

Valamilyen szempontból egyre jobb hatásfokú, vagy egyre jobban alkalmazkodó struktúrák előállítása tulajdonképpen egy optimalizációs folyamat. Természetesen az optimum elérését összességében nem biztosítja az adaptív rendszer, de optimalizációs problémák megoldásához könnyen továbbfejleszhető. Így az evolúciós algoritmusra, mint optimalizációs módszerre szokás tekinteni, amely a legösszetettebb problémák esetén alkalmazható. Számos feladtnál, amelynek megoldása optimalizálásra visszavezethető, sikerrel alkalmazható az evolúciós algoritmus.

Az optimalizálás mellett a legkülönbözőbb területeken találkozunk az evolúciós algoritmus alkalmazásával. A mérnöki munka tervezési folyamatában legyen az akár áramkör, repülőgép tervezés, gazdasági folyamatok szimulációja, munkautemezés, piaci árak meghatározása, szabályzók

tervezése, információ visszakeresése stb. egyaránt találunk sikeres alkalmazásokat.

A problémátípusok, amelybe besorolható az alkalmazási területek többsége:

1. Részhalmaz kiválasztás: az objektumok egy halmazából ki kell választani egy részhalmazt (pl. hiba diagnosztika)
2. Szenárió generálás: a rendszer szimulálása, hogy különböző feltételek mellett vizsgálhassuk a rendszer dinamikáját, fejlődési tendenciáját (pl. gazdasági folyamat modellezés)
3. Paraméter beállítás: adott struktúrájú rendszer paramétereinek becslése (pl. paraméter becslés matematikai képletekhez, neurális hálókhoz).
4. Struktúra fejlesztés: optimális, megfelelő minőségű rendszerstruktúra kialakítása (pl. neurális háló struktúra, klaszterek kialakítása).
5. Hozzárendelési problémák: objektumok egy halmazát egy másik objektum halmazra képezzük le, figyelembe véve a megadott feltételeket
6. Műszaki tervezés: műveletek olyan sorrendjének a meghatározása, amelynek egy kiindulási állapotot egy kívánt állapotba transzformálnak [1]

3 ÚTVONAL-OPTIMALIZÁLÁS

3.1 Excel Solver

A Solver típusú programok olyan matematikai optimalizálási alkalmazások, amellyel matematikai feladatot lehet megoldani. Cél olyan általános modell megalkotása, melynek segítségével számos hasonló probléma megoldható. A Microsoft Excel bővítménye az Excel Solver működési elve, hogy a megszorítások figyelembevételével megvizsgálja a lehetséges megoldásokat, majd azok közül kiválasztja a számunkra legkedvezőbbet.

Excel Solver segítségével az ún. célértékcéllában található képlet optimális (minimális, maximális vagy adott célértéket felvevő) értékét lehet megkeresni a megkötések vagy korlátozások felállításával a munkalapon szereplő többi képletcella értékeiben. Ehhez a Solver a cellák olyan, döntési változóknak vagy egyszerű változócelláknak nevezett csoportját használja fel, amelyek a képletek kiszámításához használhatók a célérték-vagy a korlátozóscellában. A Solver úgy módosítja a döntési változócellák értékeit, hogy megfeleljenek a korlátozóscella megkötéseinek és a célértékcéllához kívánt eredményt hozzák létre [3].

3.1.1 A 2021-es versenynaptár

A 2021-es versenynaptár végül 21 nagydíjhelyszínből és 22 megrendezett futamból állt. Egy alkalommal volt dupla hétvége, ami azt jelenti, hogy egymást követő két hétvégén ugyanazon a nagydíjhelyszínen két különböző néven futó nagydíjat rendeztek.

Az 1. táblázatban lehet látni a pályák nevét és a földrajzi koordinátáikat.

1. táblázat - A pályák elnevezései és földrajzi koordinátái

	Circuit	Latitude	Longitude
1	Bahrain International Circuit	26.0325	50.510556
2	Imola Circuit	44.34111	11.713333
3	Algarve International Circuit	37.232	-8.632
4	Circuit de Barcelona-Catalunya	41.57	2.261111
5	Circuit de Monaco	43.734722	7.420556
6	Baku City Circuit	40.3725	49.853333
7	Circuit Paul Ricard	43.250556	5.791667
8	Red Bull Ring	47.219722	14.764722
9	Silverstone Circuit	52.078611	-1.016944
10	Hungaroring	47.582222	19.251111
11	Circuit de Spa-Francorchamps	50.437222	5.971389
12	Circuit Zandvoort	52.388819	4.540922
13	Monza Circuit	45.620556	9.289444
14	Sochi Autodrom	43.410278	39.968271
15	Istanbul Park	40.951667	29.405
16	Circuit de Americas	30.132778	-97.64111
17	Autódromo Hermanos Rodriguez	19.406111	-99.0925
18	Interlagos Circuit	-23.70111	-46.69722
19	Losail International Circuit	25.49	51.454167
20	Jeddah Corniche Circuit	21.631944	39.104444
21	Yas Marina Circuit	24.467222	54.603056

A következő lépés a pályák földrajzi koordinátái között kiszámítani a távolságot euklideszi módszerrel.

A 2. táblázatban balról láthatjuk az optimalizálás előtti távolságokat, sorrendet és az összegüket, ez a sorrend az, amit a valóságban meg is valósítottak, tehát a versenynaptár sorrendje. Jobbról láthatjuk az optimalizálás utáni eredményeket, láthatjuk a megváltozott sorrendet.

Az eredeti összeg 630,497(66 288,536 km) volt, míg optimalizáció után az eredmény 396,3041(41 957,074 km) lett. Ez jelentős javulást jelent, ki tudjuk jelenteni, hogy az Excel Solver kitűnően végezte munkáját.

2. táblázat - Az útvonal hossza optimalizálás előtt és után

1	42.9002		7	3.9101
2	21.5516		4	11.7251
3	11.7251		3	71.8457
4	5.5952		18	67.8491
5	42.5658		17	10.8244
6	44.1556		16	99.0851
7	9.8117		9	5.5665
8	16.5127		12	2.4197
9	20.7608		11	9.3635
10	13.5832		8	4.5010
11	2.4197		10	12.1271
12	8.2679		15	10.8456
13	30.7583		14	10.3413
14	10.8456		6	16.5993
15	127.5059		21	3.3108
16	10.8244		19	1.0884
17	67.8491		1	12.2256
18	109.7883		20	35.5806
19	12.9383		2	2.7408
20	15.7558		13	2.6550
21	4.3816		5	1.6993
1			7	
Total	630.4969		Total	396.3041

3.1.2 Az összes eddigi, azaz 73 pálya

A Formula – 1 1950 óta íródo történelmében a 2020-as évig összesen 73 pálya szerepelt. Érdekes optimalizációs feladat megtalálni azt a legrövidebb utat, amivel az összes pályát be lehet járni. Ennél a feladatnál jóval nagyobb táblázatokról van szó, $73 \times 73 = 5329$ cellából álló. Szintén a pályák földrajzi koordinátái és a közöttük lévő euklideszi módszerrel kiszámított távolság alapján történt az optimalizáció.

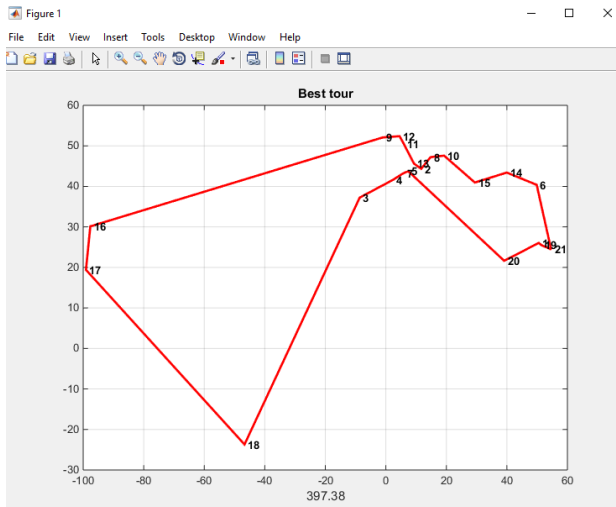
Mikor ABC sorrendbe lettek állítva a pályák, a közöttük lévő távolság összege 5232,484 (553 973,546 km) volt, ezután használatra került az Excel Solver és optimalizáció után ez a távolság 953,542 (100 953,399 km) lett. Solver ismét remekül teljesített, sokkal rövidebb optimális útvonalat talált, mint amilyen az eredeti volt.

3.2 Matlab

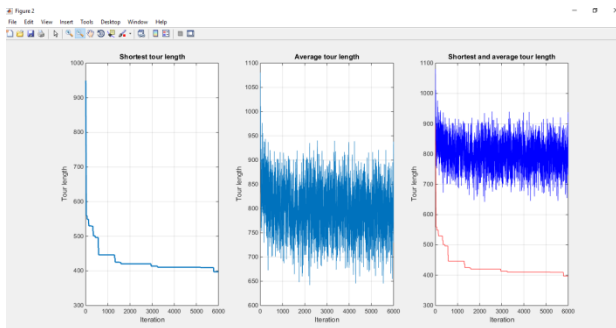
A Matlabban írt program az evolúciós algoritmus során használt módszereket alkalmazza, mint amilyen a szelekció, mutáció, keresztezés és a visszahelyezés. A program egyszerre 8 egyednél keresi az optimális útvonalat, tetszőleges számú generációra tudjuk lefuttatni a programot. Ahhoz, hogy minél jobb, azaz optimális, eredményt kapjunk, célszerűbb minél több generációra futtatni a programot, egy bizonyos pontig, amíg látjuk az eredmény javulását.

3.2.1 A 2021-es versenynaptára

A 12-es ábrán lehet látni a koordinátarendszerben megadott pályák koordinátáit és az optimalizáció után eredményként berajzolt optimális útvonalat. Az optimalizáció után a kapott eredmény, azaz az optimális útvonalhossz 397,38 (42 071,415 km).



12. ábra - Az optimalizáció utáni útvonal grafikonon ábrázolva



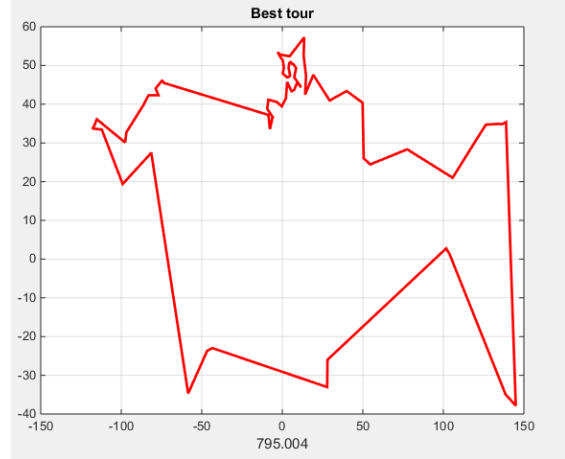
13. ábra - A legrövidebb, az átlagos útvonal és a kettő összevetve egy grafikonon

A program 6000 generációra lett lefuttatva (13. ábra), ha növeljük a generációs számot ettől nagyobbra, nincs jelentős javulás, nem produkált a program ettől rövidebb útvonalat. Az ábrán három grafikon látható, balról a legrövidebb útvonalak értéke van bejelölve, a középsőn az átlagútvonalak, jobbról pedig mindkettő összevetve.

Mindhárom grafikon a generációk futtatásának függvényében alakul.

3.2.2 Az összes eddigi, azaz 73 pálya

Egy kicsit egyszerűbb feladat után egy bonyolultabb lett feladva a Matlabban íródott programnak. Ugyanazokkal a koordinátákkal dolgozott ugyanolyan sorrendben, mint az Excel, azonban más eredményt produkált. Az optimális útvonal az alábbi ábrán látható. A program 50 egyedre és 10.000 generációra lett lefuttatva (14. ábra).

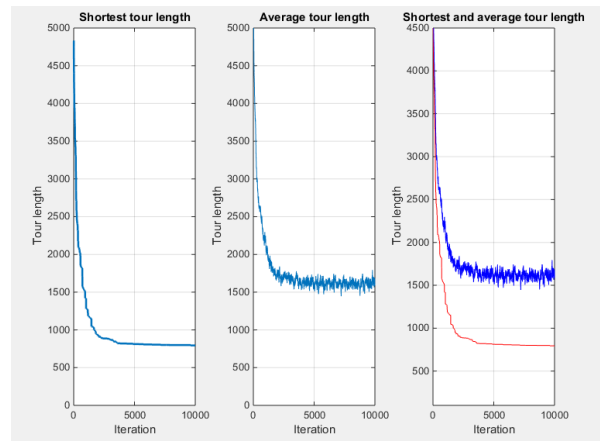


14. ábra - Az optimalizáció utáni útvonal és annak hossza

Láthatjuk, hogy a program lefutása után a kapott eredmény 795,004 (84 168,664 km) lett, ami jobb, mint az Excelben kapott. A 15. ábrán az optimális útvonal látható rávetítve a világtérképre.



15. ábra - Az optimalizált útvonal rávetítve a világtérképre



16. ábra - A legrövidebb, az átlagos útvonal és a kettő összevetve egy grafikonon

A 16. ábrán látszik, hogy az optimalizáció hasonlóan indult, mint Excelben, 5000 körüli értékétől, azonban az optimális útvonal hossza kevesebb lett.

A programot lehet futtatni Részletes megfeleltetésű keresztezéssel (PMX) és Order crossoverrel (OX) illetve keresztezés nélkül is. Hasonló eredmények születnek mindhárom alkalommal. Mikor keresztezés nélkül fut a program, lényegesen gyorsabban lefut, viszont korai konvergencia következik be, azaz gyorsan eléri az optimumot, utána pedig már nem javul tovább.

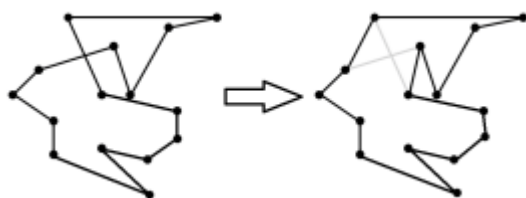
3.2.3 2-opt módszer

1958-ban Croes találta ki a módszert Az utazóügynök probléma megoldására. Ez egy javító algoritmus, tehát már egy meglévő utat fejleszt.

Alapelve:

- Vesz két éleket a körútból. Legyen ezek súlya $s(a, b)$ és $s(c, d)$
- Összeköti a két élhez tartozó négy csúcsot (nem lehet a két élnek közös csúcsa) úgy, hogy új körutat kapjunk. Ezt csak egyféleképpen tehetjük meg úgy, hogy továbbra is körutat kapjunk.
- Az élek új súlyai $s(a, c)$ és $s(b, d)$. Ha ezek kisebbek, mint $s(a, b)$ és $s(c, d)$ akkor az eredeti éleket lecseréljük a kisebb összsúlyú élekre.

Az algoritmus akkor áll le, ha már nem tudunk cserét végrehajtani (17. ábra).



17. ábra - 2-opt. példa

A programba a 2-opt. módszer is beépítésre került, amelynek pozitívuma, hogy a program futása közben látható ahogy az élek cserélődnek és ahogy rövidül az útvonalhossz. Szintén hasonló eredményeket kaptunk, mint az előbb említett módszerekkel. A versenynaptár optimalizálásakor 40 848,594 km-t, míg az összes nagydíjhelyszín esetén 86 181,925 km-t.

4 ÖSSZEGRÉS

A munka az elméleti és gyakorlati részből áll. Az elméleti részben főleg az evolúciós algoritmusok fogalma, működési elve, alkalmazási területei kerülnek feldolgozásra. Milyen eljárásokat használ, mik a kulcsfogalmak. Számos alkalmazási területe van, azaz számos problémát lehet megoldani a segítségével, mint a pl. részhalmoz kiválasztás, struktúra fejlesztés, műszaki tervezés stb.

A gyakorlati részben útvonal-optimalizációra került sor a Microsoft Excel Solver nevű bővítményével, illetve Matlabbal. Mindkét módszer ugyanazokon az alapokon nyugszik, evolúciós algoritmust használ. Amíg a Solver egy beépített funkció, nem tudjuk valójában mi történik,

addig Matlabban, pontosan látjuk, milyen lépések mennek végbe, hogyan is történik valójában az optimalizáció. A 2021-es versenynaptár került optimalizálásra, ami 21 nagydíjhelyszínt tartalmaz, tehát 21 koordinátapár közötti utat kellett optimalizálni, tehát a legrövidebbet megtalálni. A koordináták között euklideszi módszerrel került kiszámításra a távolság, ezekkel a távolságokkal lett a későbbiekben dolgozva.

Solver csak egy bizonyos mennyiségű beviteli adattal tud jó eredményeket elérni, ugyanis mikor az eddigi összes nagydíjhelyszínt, azaz a szám szerint 73 koordinátát és a közöttük lévő távolság lett beviteli adatként megadva, közel sem tudott olyan mértékű optimalizálást végrehajtani, mint előtte kevesebb adattal.

Ami a versenynaptárt illeti Excelben kilométerben kifejezve az optimalizálás előtti útvonal hossza 66 288,536 km, míg az optimalizált 41 957,074 km, Matlabban (keresztezéssel és 2-opt-tal is) optimalizálás után 40 848,594 km-t kaptunk, közel 1000 kilométernyi a különbség a két módszer között.

Ami az összes eddigi, azaz 73 pályát érinti, az optimalizáció előtti útvonal hossz kilométerbe átszámítva 553 973,546 km (viszonyítási alapként ez több mint tizenháromszorosa az egyenlítő hosszának), Excelben optimalizálás után 139 465,186 km, míg Matlabban 86 131,107 km. Matlabban az optimális útvonal több mint 50000 kilométerrel rövidebb, mint Excelben. Mindkét módszerrel jóval rövidebb utat találtunk az eredetinél, azonban ebben a feladatban Matlabban írt program jobban teljesített az Excelnél. A 2-opt. módszerrel optimalizálás után 86 181,925 km-t kaptunk, tehát mindössze 50 kilométernyi különbség van ekkora távra a keresztezési módszerek és a 2-opt. módszer között, tehát mindkettő sokkal jobban teljesített, mint az Excel. A következő lépés a 3-opt módszer beépítése lenne a programba.

5 IRODALOMJEGYZÉK

- [1] Borgulya István: *Evolúciós algoritmusok*, Dialóg Campus Kiadó, 2004, ISBN 963 9542 41 5
- [2] John Henry Holland: *Adaptation in Natural and Artificial Systems*, 1975
- [3] Bence Keresztury: *Genetic Algorithms and the Traveling Salesman Problem*, 2017
- [4] Takács Á., dr. Kamondi L. - *A genetikus algoritmusok*
- [5] Kulcsár Z. – *Az utazóügynök probléma és alkalmazásai*, 2017
- [6] Kramer, O. (2017). *Genetic Algorithms*, *Studies in Computational Intelligence*, 11-19. Doi: 10.1007/978-3-319-52156-5_2
- [7] Shukla, A., Pandey, H. M., & Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). Doi: 10.1109/ablaze.2015.7154916
- [8] Sastry, K., Goldberg, D., & Kendall, G. (2005). *Genetic Algorithms. Search Methodologies*, 97-125. https://doi.org/10.1007/0-387-28356-0_4
- [9] *Algoritmi veštačke inteligencije*, edicija Grogking, Rishal Hurbans Kompiuter biblioteka – Beograd (2021) ISBN: 978-8673105611
- [10] Michael Negnevitsky: *Artificial Intelligence, A Guide to Intelligent Systems*. Pearson Education Canada; 3rd edition (2011) ISBN: 978-1408225745