

DIPLOMAMUNKA

Györke István Károly

Debrecen
2010

**Debreceni Egyetem
Informatikai Kar**

BEFUTÓSORI SZELEPEK HOZAM SZABÁLYOZÁSA

Témavezető:
Dr Aszalós László
egyetemi adjunktus

Készítette:
Györke István Károly
PTi-MSc

Debrecen
2010

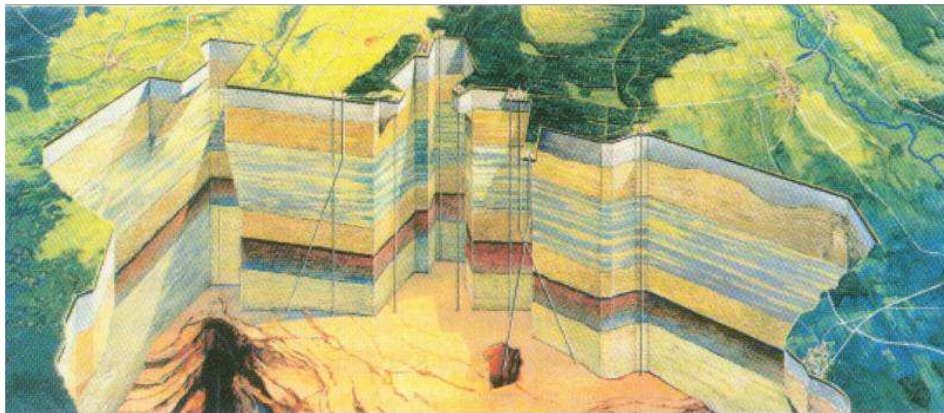
Tartalomjegyzék

1	A földalatti gáztárolás.....	3
1.1	Földalatti gáztárolás Magyarországon.....	4
2	Szabályozási feladat a befutósoron.....	8
2.1	A jelenlegi algoritmus	9
2.2	Probléma a jelenlegi algoritmussal.....	11
2.3	A dolgozat célja	11
2.4	A tervezett programszerkezet	12
3	Az evolúciós stratégia	14
4	Az algoritmus megvalósítása	16
4.1	Egyedek	16
4.2	Fitneszfüggvény és kiértékelése	18
4.3	Szelekció.....	21
4.4	Mutáció	22
4.5	Visszahelyezés.....	24
5	Implementáció	25
5.1	A megjelenítési és kezelés felület.....	25
5.2	Módszerek	28
5.3	Kezelői Utasítás.....	39
6	Eredmények	41
7	Összefoglalás	43
	Alkalmazott rövidítések jegyzéke	44
	Irodalomjegyzék	45
	Ábrajegyzék.....	46
	Köszönetnyilvánítás	46

1 A földalatti gáztárolás

A földgázkészletek csökkenésével egy időben növekvő energiaszükséglet miatt a környező országok és így Magyarország megnövekedett földgázszükségletét a téli időszakban a földalatti gáztárolókból visszanyert földgázzal lehet kielégíteni [1].

A földgáztárolás a nyári hónapokban távvezetéken érkező vásárolt gáznak földalatti gáztárolókba történő besajtolását jelenti. A földalatti gáztárolás kanadai találmány. 1915-ben Ontarióban létesítették a világ első földalatti gáztárolóját egy kiürült gázmezőben. Ennek a technológiának a jelentősége egyre inkább növekszik. „A földgázforrások ugyanis egyre távolabb kerülnek a felhasználás helyétől, ezért a földgáztárolók fontossága is jelentősen megnövekedik: az igények ingadozásának kiegyenlítésén túl az ellátásbiztonság garantálásban is meghatározó szerepet fognak játszani. A földgáztárolók ugyanis fontos kiegyenlítő funkciót töltenek be a gázipar infrastruktúrájában: a kitermelő országokból a földgáz folyamatosan érkezik a célországokba, a fogyasztás ugyanakkor szezonális és egyéb okok miatt erősen ingadozik. A hazai gázpiaci nyitás következménye, hogy több kereskedő nagyobb rugalmasságot keres a kereskedelmi szerződéseinek optimalizálására. Itt jut kiemelt szerep a tárolóknak.” [2]



1.1 ábra Egy földalatti gáztároló sematikus rajza

Alapvetően két főbb geológiai képződmény szolgál a földgáz tárolására: a kaverna (vagy más néven: barlangtároló) és a pórusos tároló. A kaverna tárolókat alkotó - többnyire henger alakú - sóbarlang-üreg kioldódás útján keletkezik a felszín alatti sótartalékokból, amelyekbe aztán belepumpálják a földgázt. A sórétegek által képzett burkon nem szivárog el a gáz. A pórusos tárolók ezzel szemben porózus vagy repedéses kőzetekből állnak, amelyek szivacsként szívják magukba a földgázt. Ezt felülről egy gázt át nem eresztő kőzetréteg zárja le, lefelé pedig vízvezető talaj biztosítja a szigetelést. A korábbi olaj- vagy gázmezők is a pórusos tárolók kategóriájába tartoznak. A kétfajta tároló tulajdonságai jelentősen különböznek egymás-

tól. A kavernatárolók kedvező bányamechanikai adottságaik miatt különösen magas tárolói nyomás kialakítására is alkalmasak, ennek következtében a bennük tárolt gáz sokkal gyorsabban kinyerhető. A pórusos tárolók ezzel szemben lassabban reagálnak a tárolói műveletekre. Egy földalatti gáztároló sematikus rajza látható az 1.1 ábrán.

1.1 Földalatti gáztárolás Magyarországon

Ma Magyarországon a földalatti gáztárolást az EON Földgáz Storage Zrt (EFS) végzi. Jelenleg öt földalatti gáztárolóval rendelkezik, melyek tárolási kapacitás szerinti sorrendben a következők:

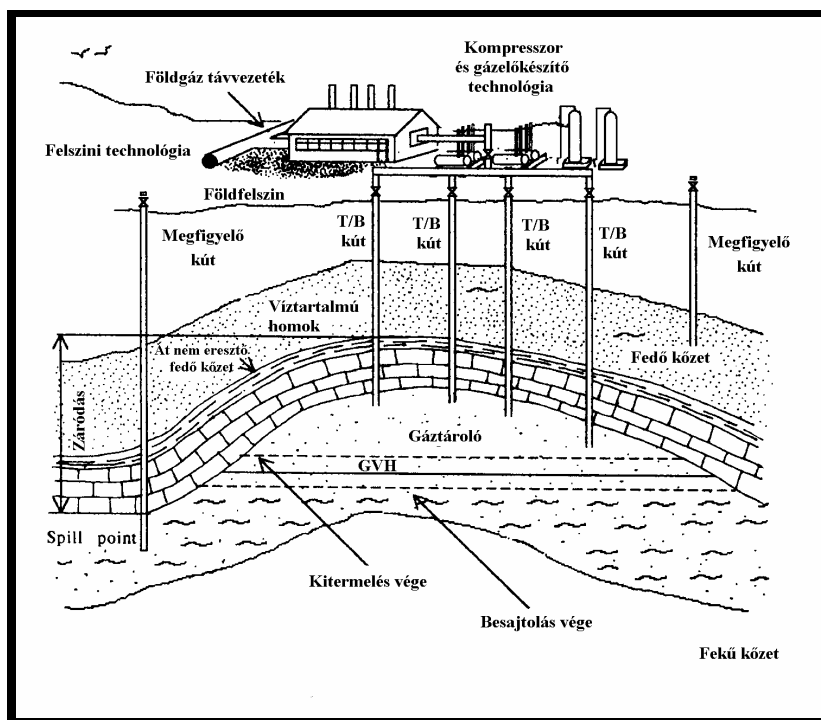
- Zsana (2,1 milliárd m³),
- Hajdúszoboszló (1,6 milliárd m³),
- Pusztaderics (340 millió m³),
- Kardoskút (260 millió m³),
- Maros-1 (125 millió m³).

Ide tartozik még a MOL Rt. tulajdonát képező – 2009-ben Szőregen átadott – úgynevezett Stratégiai Tároló, melyben 1,2 milliárd m³ földgázt stratégiai célra, míg 700 millió m³-t kereskedelmi célra tárolnak.

Az EFS földgáztárolói korábbi földgázmezőkön épültek, így a pórusos tárolók kategóriájába tartoznak. A Pusztadericsi, a Hajdúszoboszlói, a Kardoskúti és a Maros-1 tároló homokkőre épül, míg a Zsanai tároló mészhomokkő alapú. A homokkő tárolók különösen „érzékenyek”, nem lehet az előírtnál nagyobb ütemben gázt besajtolni vagy kitermelni belőlük. A Zsanai tároló réteg anyaga ennél jóval rugalmasabb, így az igények gyorsabb változására is biztonságosan tud reagálni. A homokkő tárolók felterhelése – azaz a beindítástól a kitermelési csúcs eléréséig tartó idő – általában legalább öt napig tart. A Zsanai tároló viszont mészkőrétegeinek kedvezőbb tulajdonságai miatt ennél sokkal gyorsabban tud reagálni az indításra.

Egy gázszárító kapacitását nem a benne tárolt gáz mennyisége, hanem inkább az jellemzi, hogy mennyi gázt és mennyi időn keresztül képes kiadni. Az EFS jelenlegi gáztárolói képesek biztosítani az országos gázigény mintegy 70%-át két hónapon keresztül. A tárolók túlnyomása feltöltött állapotban 100 barg körüli. A gázipari gyakorlatban megkülönböztetik az abszolút nyomást (bara) valamint a légköri nyomással csökkentett túlnyomást (barg). Az egyes tárolók túlnyomása – helyi geológiai adottságoknak megfelelően – eltérő.

Kitermeléskor a kitermelt gázt szárítani kell a távvezetéki szállítás előtt, hiszen a távvezetékre csak bizonyos harmatpont alatt adható ki gáz. Kitermelési üzemmódban fokozottan kell gondoskodni arról, hogy a gáztárolók magas nyomása ne tudjon kikerülni az országos gázhálózat 64 barg-os rendszerére.



1.2 ábra A földalatti gáztárolás sémája

A földalatti gáztárolók 1-2.000 m mélységben lévő, letermelt gáztelepek, melyek zárt geológiai szerkezetük révén alkalmasak a téli időszakra, illetve a nagyobb fogyasztási csúcsok eliminálására előre megvásárolt gáz tárolására.



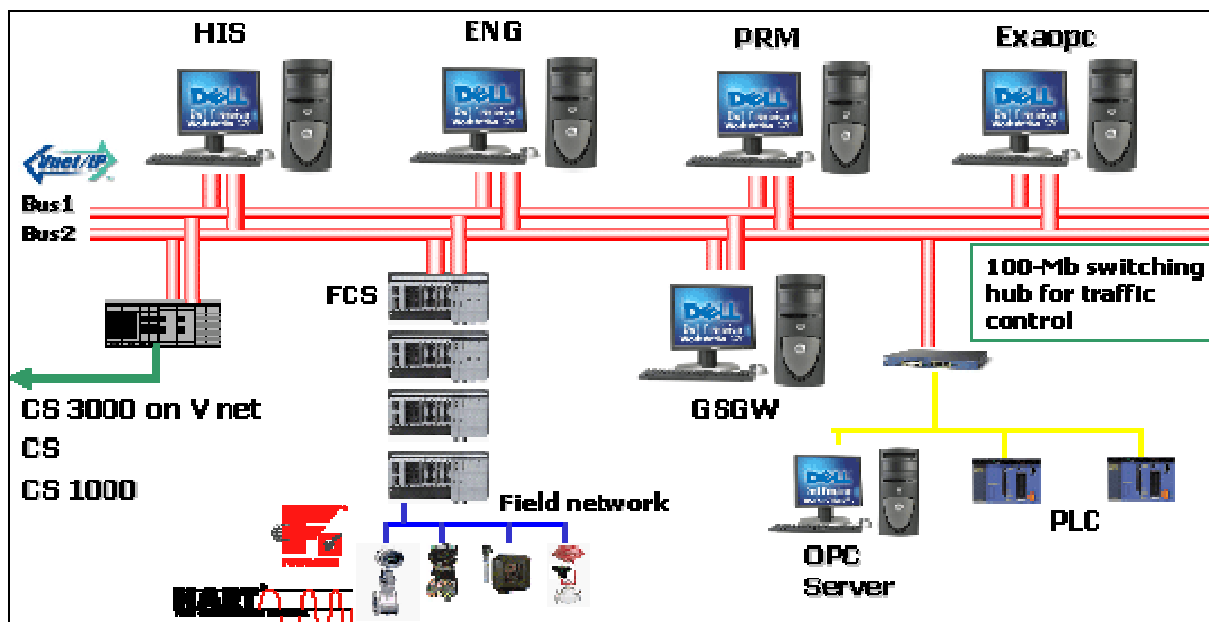
1.3 ábra Egy gázkút Hajdúszoboszló körzetében

A földalatti gáztárolók viszonylag egyszerű technológiai folyamattal rendelkeznek. Egy tároló sematikus rajza látható az 1.2 ábrán.

A nyári hónapokban a távvezetéken érkező földgázt – általában – dugattyús és villany - vagy gázmotorral hajtott kompresszorokkal sajtolják a tároló rétegbe. Télen a rétegből a gáz abszorpciós gázszáritókon keresztül áramoltatva szabadul meg a felvett víztől, majd utána kerül az országos távvezetéki rendszerbe.

A tároló réteggel – mely Hajdúszoboszlón 800-1.200 m mélyen helyezkedik el – a kapcsolatot a gázkutak tartják. Hajdúszoboszló körzetében mintegy 109 db kút van a rétegbe fúrva. Ezeket a kutakat csővezetékek kötik össze földgáztárolónak nevezett üzemmel, ahol a kompresszorok, gázszáritók és egyéb felszíni szerelvények találhatók. Egy gázkút fényképe látható az 1.3 ábrán.

Az egyszerű technológia ellenére, a - fokozott tűz és robbanásveszély miatt – ezek az üzemek fejlett irányítástechnikai rendszerrel rendelkeznek. A folyamatirányítást általában elosztott irányító rendszerek (DCS) végzik. Ezek robusztus felépítésű, magas hibatűrésű, nagy megbízhatósággal rendelkező ipari számítógépek összekapcsolásával nyert hálózatok. A DCS gyártók ma már az úgy nevezett „hét-kilences” azaz 99,99999%-os megbízhatóságot garantálják. Ez a szám az MTBF (Meghibásodások között átlagosan eltelt idő). Ez alatt azt kell érteni, hogy minden tízmilliomodik órában történhet egy meghibásodás. Vagyis 1.000 telepített rendszer közül egy 10.000 óránként romolhat el úgy, hogy üzemzavart okoz. Ezt a biztonságot azzal érik el, hogy hardveres és szoftveres redundanciákat tartalmaznak.



1.4 ábra DCS struktúra redundáns busszal

Hajdúszoboszlóra a YOKOGAWA cég CENTUM CS3000 típusú rendszere lett telepítve 1999/2000-ben. Az időközbeni fejlesztések eredményeként ma már három generációja van jelen a rendszerben. Egy redundáns DCS hálózatot mutat be az 1.4 ábra.

A Hajdúszoboszlói Földalatti Gáztároló két üzemrésszel rendelkezik. Az úgynevezett „Déli-telep” Nagyhegyes és Hajdúszoboszló között félúton, míg az „Északi-telep” tőle hat km-re Nagyhegyes területén található. Mivel Control Unit-ok mindkét telephelyen vannak, a DCS hálózat belső buszát (Vnet) redundáns és determinisztikus üvegkábel köti össze.

A rendszer jelenleg mintegy 17 db terepi vezérlőt (FCS), 3 db biztonsági állomásvezérlőt (SCS) és 8 db kezelői állomást (HIS) tartalmaz. Az 1.5 ábra a DCS fejlesztő rendszerének, a SystemView-nek a részletét mutatja. Az FCS0105 szimbóluma ki van nyitva, alkönyvtárként látszanak az összetevői, valamint a jobb oldali ablakban, az ebben az FCS-ben található funkció blokkok szimbólumai.

All Folders		Opened Folder: FUNCTION_BLOCK			
		Name	Type	Modified	Comment
SYSTEM VIEW		DR0001	Control Drawing	2004/10/10 15:57	A 169-es kút szab. köre
FGT		DR0002	Control Drawing	2004/10/10 15:57	A 172-es kút szab. köre
COMMON		DR0003	Control Drawing	2004/10/10 15:57	A 164-es kút szab. köre
BATCH		DR0004	Control Drawing	2004/10/10 15:57	A 171-es kút szab. köre
FCS0101		DR0005	Control Drawing	2004/10/10 15:57	A 166-os kút szab. köre
FCS0102		DR0006	Control Drawing	2004/10/10 15:57	A 145-ös kút szab. köre
FCS0103		DR0007	Control Drawing	2004/10/10 15:57	A 165-es kút szab. köre
FCS0104		DR0008	Control Drawing	2007/11/22 11:38	A 159-es kút szab. köre
FCS0105		DR0009	Control Drawing	2004/10/10 15:57	A 173-as kút szab. köre
CONFIGURATION		DR0010	Control Drawing	2004/10/10 15:57	A 157-es kút szab. köre
SEQ_LIBRARY		DR0011	Control Drawing	2004/10/10 15:57	A 167-es kút szab. köre
IOIM		DR0012	Control Drawing	2006/12/13 14:03	A 200-as kút (Régi168)
SWITCH		DR0013	Control Drawing	2004/10/10 15:57	A 158-as kút szab. köre
MESSAGE		DR0014	Control Drawing	2004/10/10 15:57	A 154-es kút szab. köre
FUNCTION_BLOCK		DR0015	Control Drawing	2004/10/10 15:57	A 153-as kút szab. köre
DISPLAY		DR0016	Control Drawing	2004/10/10 15:57	A 170-es kút szab. köre
FCS0106		DR0017	Control Drawing	2004/10/10 15:57	A 140-es kút szab. köre
FCS0107		DR0018	Control Drawing	2004/10/10 15:57	A 139-es kút szab. köre
FCS0108		DR0019	Control Drawing	2004/10/10 15:57	A 143-as kút szab. köre
FCS0109		DR0020	Control Drawing	2004/10/10 15:57	A 135-ös kút szab. köre
FCS0110		DR0021	Control Drawing	2004/10/10 15:57	A 52-es kút szab. köre
FCS0111		DR0022	Control Drawing	2007/02/08 12:13	A 137-es kút szab. köre
FCS0112		DR0023	Control Drawing	2004/10/10 15:57	A 144-as kút szab. köre
FCS0113		DR0024	Control Drawing	2005/11/05 15:12	A 194 kút
FCS0114		DR0025	Control Drawing	2007/11/20 15:48	kutak adatai
BCVV0132		DR0026	Control Drawing	2007/11/20 15:26	Kút üzemáll. számítási konstansok
HIS0160		DR0027	Control Drawing	2004/10/10 15:57	
HIS0161		DR0028	Control Drawing	2004/10/10 15:57	
HIS0162		DR0029	Control Drawing	2004/10/10 15:57	
HIS0163		DR0030	Control Drawing	2004/10/10 15:57	
HIS0164		DR0031	Control Drawing	2004/10/10 15:57	
SCS0301		DR0032	Control Drawing	2004/10/10 15:57	
FCS0302		DR0033	Control Drawing	2004/10/10 15:57	
SCS0303		DR0034	Control Drawing	2004/10/10 15:57	
FCS0304		DR0035	Control Drawing	2004/10/10 15:57	
SCS0305		DR0036	Control Drawing	2004/10/10 15:57	
FCS0306		DR0037	Control Drawing	2004/10/10 15:57	
HIS0361					
HIS0362					
HIS0364					

1.5 ábra A SystemView részlete

A kezelő személyzet a kapcsolatot a rendszerrel a HIS állomásokon keresztül tartja. Ezek PC-k, melyeken Windows alapú szoftver fut. Rá vannak kapcsolva a rendszer buszára, a Vnet-re. Kitüntetett közöttük a mérnöki állomás. Ezek az állomások a megjelenítést és adatbevitel szolgáltatják, de maga a DCS rendszer ezek kiesése esetén is működőképes maradna.

2 Szabályozási feladat a befutósoron

A gáztároló kutak nagy száma és a változó terhelés melletti irányíthatóság, valamint a kutak védelme megköveteli, hogy a kutak egyedi hozamszabályozási rendszerrel működjenek, melyet a folyamatirányító rendszer irányít és felügyel.

Dolgozatomban, a hajdúszoboszlói tárolóban megvalósított algoritmust mutatom be – a dolgozat terjedelme által megengedett kereteken belül.

A kutak nagy számát ezen a tárolón a 109 db kút jelenti. Az országos gázellátás számára legyen az kitermelés, vagy besajtolás, egy tároló csak globálisan, egy mennyiségi adatként jelentkezik. Vagyis hogy mennyi földgázt kell az adott tárolónak óránként betárolni, vagy kitermelni. Ez Hajdúszoboszló esetében besajtoláskor maximum 600 ezer Nm³, míg kitermeléskor 100 és 830 ezer Nm³ közötti földgázt jelent óránként.

Ezt a mennyiséget kell szétosztani a 109 db kút között. De a gázkutaknak is meg van a maguk kezdeti, illetve maximális kapacitásuk, amit az úgynevezett REZSIM utasítás szabályoz le. Erre a tároló réteg porózus szerkezete, illetve a kutak védelme miatt van szükség.

A változási sebesség sem választható meg szabadon, mert a kutaknál fokozott a homokosodás és a vízbetörés veszélye.

A kutak nem mindegyike alkalmas mindkét üzemmódra. Vannak KÉTIRÁNYÚ, illetve csak BESAJTOLÓ, illetve csak KITÁROLÓ kutak.

A mennyiségsszabályozásban nem csak a kutak vesznek részt, hanem a kompresszorok és a gázszárítók is. Ezen kívül külön követelményeket támaszt az algoritmussal szemben, hogy a tároló – a mennyiség tekintetében – stabilizálódott, vagy tranziens illetve kitermelési vagy besajtolási üzemmódban van-e.

A besajtolási üzemállapot az egyszerűbb, ilyenkor:

1. Tranziens üzemállapotban: a kutaknál a rezsimumra történik szabályozás. A mennyiséget ilyenkor a kompresszorokkal állítják be.
2. Állandósult állapotban: a mennyiséget a kutak oldalán kell biztosítani, a kompresszorokat pedig a mennyiségnek megfelelően terhelni.

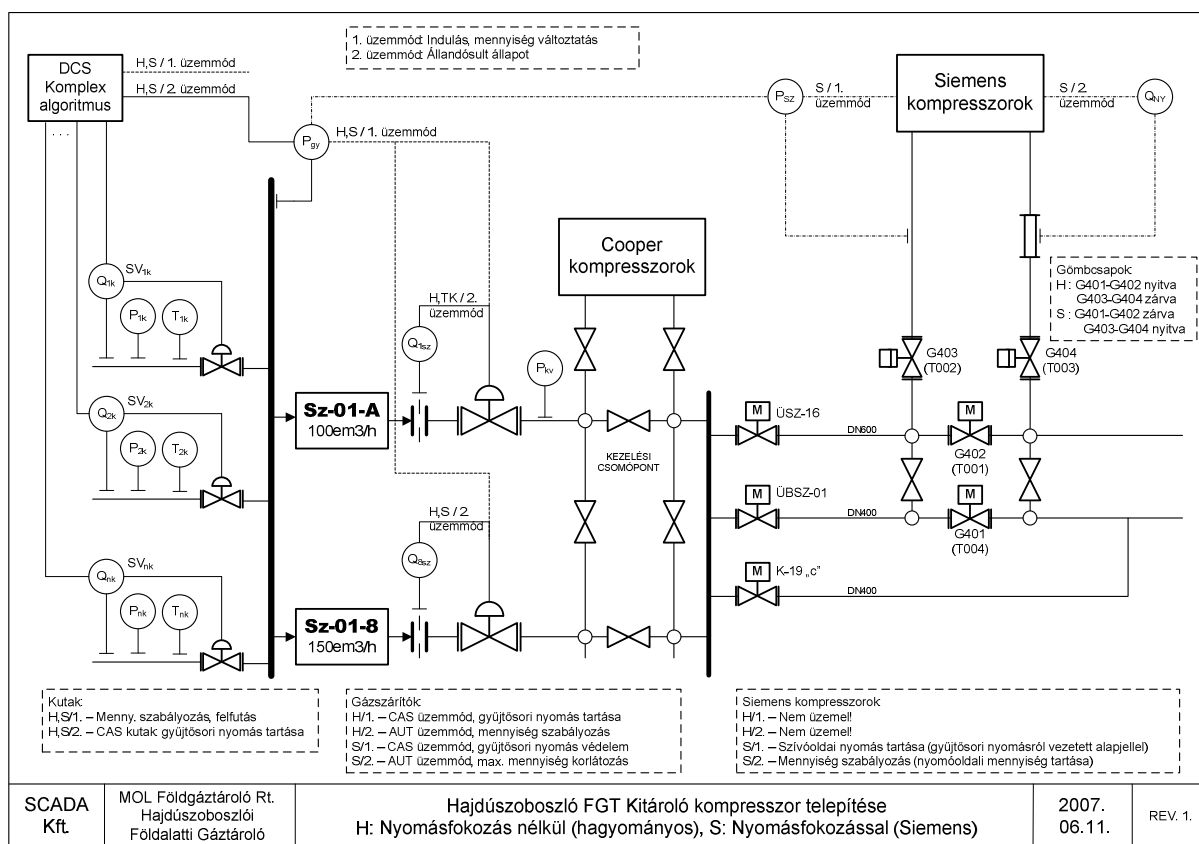
A kitermelési üzemállapot ettől komplikáltabb:

1. Tranziens állapotban: a kutaknál egyedi mennyiségsszabályozás történik, míg a gázszárítóknál a nagy szárítók túlnyomás szabályozást, a kisebb szárítók pedig mennyiség szabályozást végeznek.

2. Állandósult állapotban: a kutak oldalán a gyűjtőori nyomás beállítása az algoritmus feladata. A gázzárítók mindegyike mennyiséget szabályoz.

Amennyiben kompresszorozásra is szükség van a kiadandó mennyiség előállításához, akkor tranzien্স üzemállapotban a kompresszorok a gyűjtőori nyomást igyekeznek tartani, állandósult állapotban pedig mennyiséget szabályoznak.

A szabályozási algoritmus feladatait részletesen nem ismertetem, mert a részletek meghaladják a dolgozat terjedelmét. A több részletben készült, részletes leírásokat a DCS rendszerterv tartalmazza. A 2.1 ábra a komplex irányítástechnikai algoritmust szemlélteti.

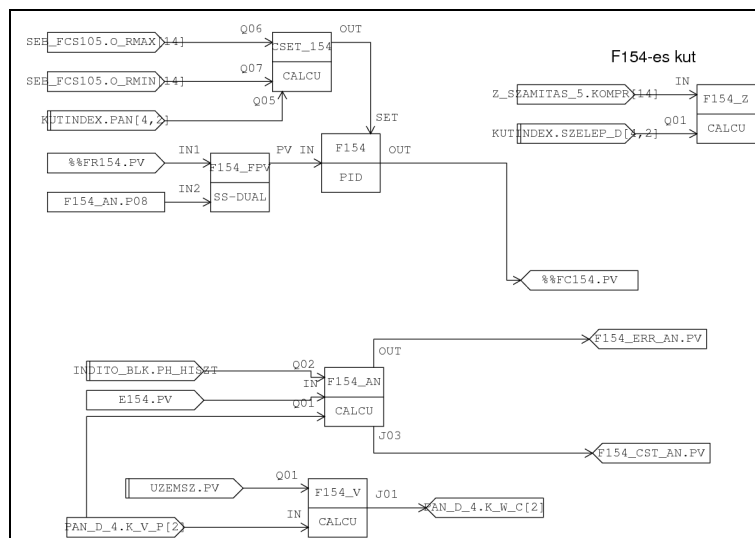


2.1 ábra A jelenlegi komplex algoritmus elvi ábrája

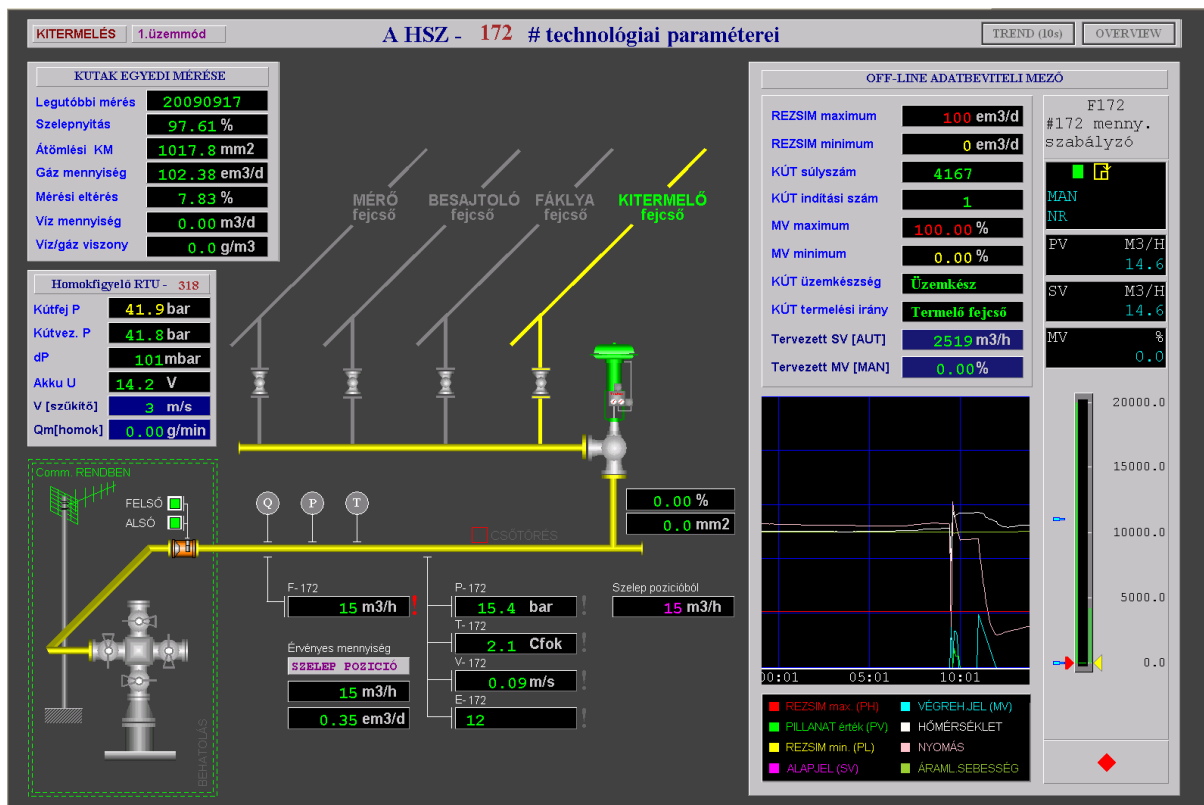
2.1 A jelenlegi algoritmus

A jelenleg futó algoritmus mintegy 10 éves fejlődés eredményeként jött létre. Ez először csak a befutósorra vonatkozott. Később az irányítástechnikai korszerűsítésen átesett gázmotoros kompresszorok, majd a direkt kitermelési céllal készült turbinás kompresszorok módosították lényegesen. De módosult, amikor megvalósult a homokfigyelés, vagy akkor is, amikor a gázzárítók átalakítása történt.

A módosításokat nem mindig ugyan az a személy végezte, nem mindig ugyanazon a programozási nyelven. A DCS rendszer belső buszát nagyon leterhelné egy állandóan futó algoritmus, ha egyetlen terepi vezérlőben futna. Ezért az algoritmus több terepi vezérlő között oszlik meg, több Task-on belül fut. Globálisan nem, csak részleteiben tekinthető át. Egy szabályozó körnek a SystemView-ben történő funkcióblokkos megvalósítását mutatja a 2.2 ábra, valamint egy kút megjelenítését – a kezelő számára – a 2.3 ábra.



2.2 ábra A Hsz-154-es kút vezérlésének funkcióblokkos sémája



2.3 ábra A 172-es kút megjelenítése a HIS állomáson

2.2 *Probléma a jelenlegi algoritmussal*

Jelenleg a termelő kutak között meg vannak különböztetve a nagyobb hozamú és gyengébb hozamú kutak. A mennyiség szétterhelése köztük úgy történik, hogy a kisebb hozamú kutak gyakorlatilag egy – az algoritmus által rájuk kiosztott – adott mennyiséget próbálnak tartani, a „jobb” kutak pedig a korrekciókat végzik. Ezek a „jobb” kutak a rájuk osztott mennyiségtől $\pm 20\%$ -al térhetnek el. Amennyiben a mennyiséget már nem tudják tartani, akkor a kezelő kezdeményezésére új szétosztást kell végeztetni a folyamatirányító géppel.

Ebből, valamint a PID szabályozók tehetetlenségéből adódóan a tényleges mennyiség mindig eltér az előírttól.

2.3 *A dolgozat célja*

Célom egy olyan algoritmus elkészítése volt, amely a mennyiségnek a kutakra szétosztását automatikusan, a változások követését valós időben végzi.

Tekintettel a dolgozatom kereteire, valamint arra hogy a meglévő algoritmus hosszas csapatmunka fejlesztésének eredménye, a jelenlegi algoritmusnak csak a kutak szabályozására vonatkozó részét modelleztem.

A munkám nem terjed ki:

- az I. és II. üzemmód megkülönböztetésére,
- a befutósori nyomás figyelésére,
- a gázszáritók, illetve kompresszorok hozamszabályozási szerepére.

A megvalósított feladat alapja a 2009.10.14.-én kiadott (kitermelési üzemmódra vonatkozó) „Egyedi rezsimutasítás a hajdúszoboszlói földalatti gáztárló kútjaira” belső utasítás.

Az Egyedi rezsimutasítás szabályozza:

- az adott - kitermelési – ciklusban résztvevő kutak körét, a három csoport valamelyikébe tartozását,
- minimális kezdő és maximális hozamát,
- a tárolóra vonatkozó minimális és maximális hozamot,
- a hozamváltoztatás megengedett sebességét (különböző hozamtartományokban),
- kétféle indulási módozatot, és az azokhoz tartozó:
 - minimális hozamokat,
 - felterhelési sebességeket,
 - „pihentetési” időket
- meghatározza melyik kútcsoporthoz milyen mennyiségnél, kell beindítani.

Az algoritmus megvalósítását a MS Excel és VBA felületén végeztem. Az Excel megfelelő az adatok felvitelére, majd az eredmény megjelenítésére és résszámitások elvégzésre, míg a VBA a program megírására és futtatására.

A sikeres szimulációk után az algoritmus átírható a YOKOGAWA CENTUM CS3000 DCS SEBOL programozási nyelvére.

A megvalósítandó feladatot a 2.4 ábrán vázoltam.

2.4 A tervezett programszerkezet

Tekintettel arra, hogy nagyszámú kimenetet és bemenetet kell a megvalósított algoritmusnak kezelni, valamint sokféle korlátozó feltételt figyelembe venni, a feladathoz megfelelően illeszkedő algoritmust kellett választanom.

Az ilyen típusú feladatok megoldására az evolúciós algoritmusok a legmegfelelőbbek. Azonban egy ilyen veszélyes üzem irányítástechnikai elvei megkövetelik az irányítástechnikai folyamatok teljes determinisztikusságát. Ebből kifolyólag nem engedhető meg az evolúciós folyamatok előre 100%-osan nem garantálható sikeressége, illetve a feladat időben nem előre látható sikeres végrehajtása.

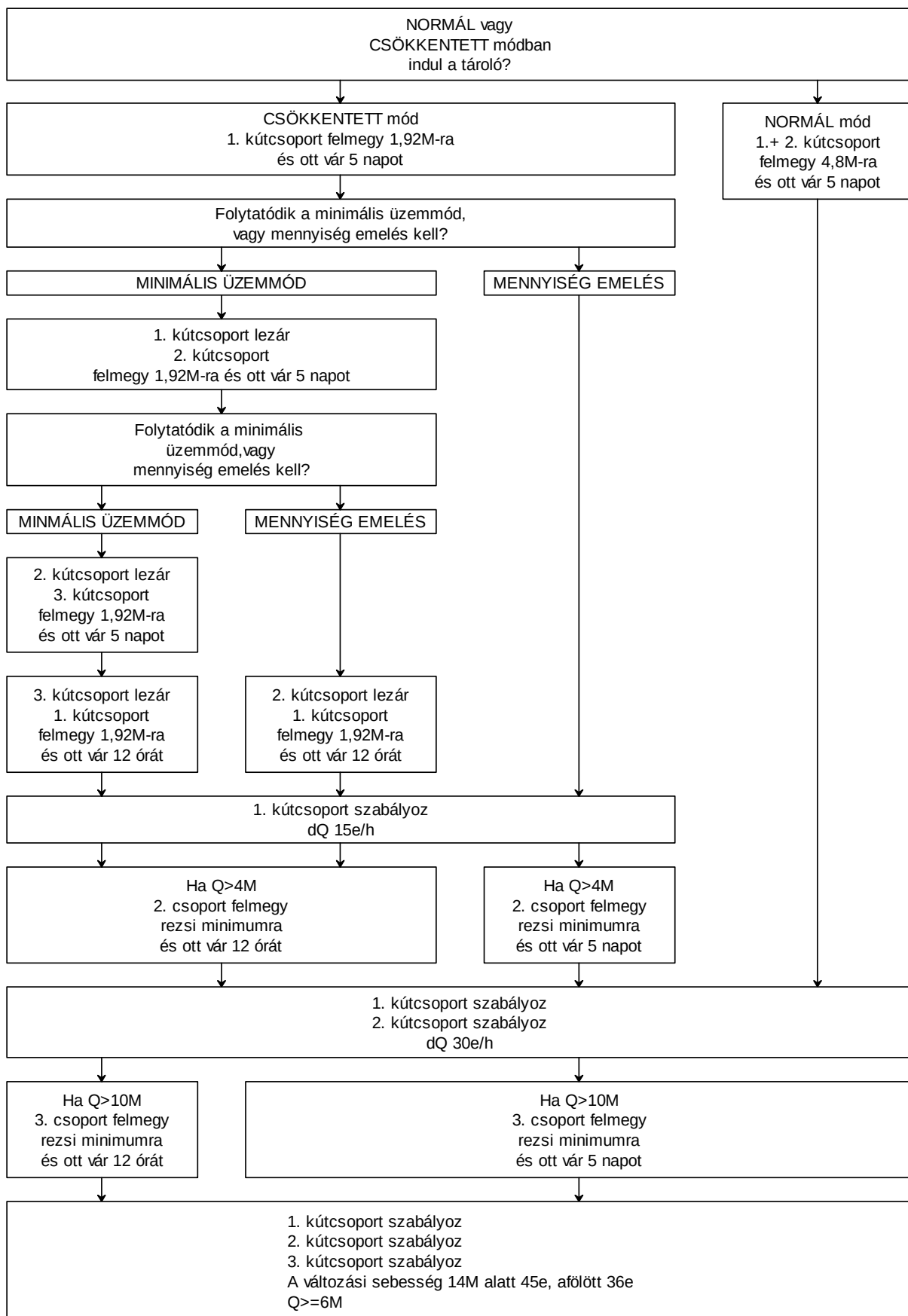
Ezért az evolúciós algoritmusok eredményeiből csak annyit használtam fel, amennyi megkönnyíti a nagyszámú elem kezelését, de a véletlen eseményeket ki kellett hagynom belőle. Mivel a kutakat nem lehet hirtelen, illetve egyszerre nagyobb mértékben mozgatni, elvettem a keresztezést is, mint evolúciós elemet.

Célomnak leginkább az *Evolúciós stratégia* felelt meg, mert nem használ keresztezést, hanem mutációval közelíti a célfüggvényt.

„Az általános ES hasonlóan a GA-hoz egy μ ($\geq$) számú egyedből álló populációt alkalmaz. Fő művelete a mutáció, de lényeges a rekombináció művelete is, amely most nem genotípus szinten kerül alkalmazásra. ...

Az ES ciklus megfogalmazásánál először a kezdő populációt kell kialakítani. Ha nincsenek előre adott megoldások, amiből kiinduljunk, akkor az értelmezési tartomány tetszőleges pontjai lehetnek az egyedek. ...

Az ES alkalmazását számos változat könnyíti meg. Különböző rekombináció és mutáció műveleteket, vagy szelekciós módot választhatunk, amelyek speciális változatként értelmezhetők.” [7]



2.4 ábra A megvalósítandó folyamat

3 Az evolúciós stratégia

„A mérnöki gyakorlatban, a gazdasági életben, kutatások során számos olyan problémával találkozunk, melyek megoldására nincs egzakt módszer. Vagy azért mert a probléma nagyon összetett, vagy azért mert nagyon számítás igényes és nincs gazdaságos módszer a megoldásra. Pl. magasabb dimenziós függvényeket kell optimalizálni, adott számú gépen ütemezni kell különböző munkák végrehajtását, vagy áramkörök, VLSI lapok, helységek gazdaságos elhelyezését kell tervezni. Az ilyen feladatok megoldásához rendszerint heurisztikus módszereket kell alkalmazni. A leggyakoribb heurisztikák a különböző kereső algoritmusok, és a mesterséges intelligencia algoritmusai, mint pl. a szimulált hűtés, az evolúciós algoritmusok. ...

Az evolúciós algoritmus (vagy az amerikai szóhasználatban genetikus algoritmus) a mesterséges intelligencia egyik metaheurisztikája. Egy általános problémamegoldó séma, melynek kialakítását a biológiai evolúció motiválta. ...

Genotípus - fenotípus¹ alapján három standard EA változat létezik: a genetikus algoritmus (GA), az evolúciós stratégia (ES) és az evolúciós programozás (EP). A GA a genotípus szintű genetikus mechanizmusokat hangsúlyozza, azaz elsősorban a génstruktúra változtatásával keres jobb megoldásokat. Az ES a fenotípus szintű evolúciót modellezi. Az EP szintén fenotípus alapján modellezi a mechanizmusokat, de az evolúciót nem egyedek, hanem a teljes populáció, azaz a „fajok” szintjén értelmezi.” [7]

A fentiek alapján a feladatot az evolúciós stratégia (ES) módszerével valósítom meg, mely algoritmus ez evolúciós eljárással keresi a probléma megoldását. Egy általános evolúciós algoritmus felépítése a következő:

- *Egyedek*: az egyed, vagy individuum mindig egy megoldást jelöl. Formai szempontból igen változatos: sztring, valós vektor, de akár összetett struktúra (program, áramköri rajz, zenem, stb.) is lehet. Összetett EA-knál az egyedeket csoportosíthatjuk valamilyen szempont szerint, és e csoporttal, mint egyeddel dolgozhat egy másik EA.
- *Fitneszfüggvény (fitness function)*: az egyed jóságát kifejező valós függvény (nevezhetjük jósági függvénynek, alkalmassági függvénynek is). Egyszerűbb problémáknál lehet a célfüggvény is a fitneszfüggvény.

¹ A *genotípus* a génekben tárolt genetikai információk összessége, amely a környezeti viszonyokkal kölcsönhatásban meghatározza a szervezet külső megjelenését, a *fenotípusát*. Ugyanaz a genotípus különböző környezetben különböző fenotípusok formájában realizálódhat (Straub, 1987.)

- *Fitneszfüggvény kiértékelés*: ez az eljárás a populáció minden egyedénél, vagy az utódoknál kiszámítja a fitneszfüggvény értékét.
- *Szelekció (selection)*: a szelekció művelet szülőket válogat a populáció egyedei közül. A válogatás általában olyan eljárásokkal történik, amelyek figyelembe veszik az egyedekhez tartozó fitneszfüggvény értéket, és a rátermettebb egyedek nagyobb súllyal lehetnek szülők.
- *Rekombináció (recombination)*: egy kereső művelet, amellyel utódokat állít elő az EA. Az utódokat egy-több szülő felhasználásával, információs anyagaik kombinálásával állítja össze.
- *Mutáció (mutation)*: egy kereső művelet, amely megváltoztatja az utódokat. Az utódot jellemző változókat kis értékkel (zajokkal) módosítja.
- *Visszahelyezés (reinsertion)*: az utódok létrehozása után ki kell alakítani az új populációt, amelyben már az utódok is helyet kapnak. Az utódok lecserélhetik az előző populációt valamilyen arányban.

„Az ES, amely fenotípus szinten modellezi az evolúciót, tulajdonságokkal írja le az egyedeket (valós vektorként ábrázolva az egyedet). A rekombináció művelete kisebb súlyú, mint a GA -nál, alkalmazásával főleg a szülők egyes „domináns” tulajdonságait lehet az utódoknak átadni. A fő művelet a mutáció, amely normál eloszlású véletlen számokkal módosít minden tulajdonságot. Az új populáció kialakítása determinisztikusan történik: az utódok és a szülők közül a legrátermettebbek fogják az új populációt képezni.” [7]

A kutak pillanatnyi hozamának ábrázolására legmegfelelőbb a valós vektor, a szabályozástechnika pedig megköveteli a szabályozási folyamat determinisztikusságát.

Az általános evolúciós stratégia lépései:

- Reprezentáció, fitnesz kiértékelés.
- Szelekció.
- Rekombináció.
- Mutáció.
- Visszahelyezés

Munkám során ezt az elvet követtem, azonban a klasszikus belső tartalomtól a determinisztikusság érdekében eltértem. Így nem egy evolúciós stratégiával készült algoritmust, csupán egy evolúciós elemeket tartalmazó algoritmust hoztam létre.

4 Az algoritmus megvalósítása

A befutósori szelepek problémáját megoldó algoritmusnak szüksége van bemeneti adatokra, valamint kimeneti felületre, ahová a legenerált egyedeket küldi.

A bemeneti felület egy Excel számolótábla, a kimenet pedig – ugyanazon a felületen – egy Excel diagram. A kimeneti felület felel meg a DCS rendszerben a PID szabályozók bemeneti pontjainak. Szintén az Excel táblán kaptak helyet a különböző futási paraméterek kijelzői, illetve a paraméterezés kezelőszervei.

4.1 Egyedek

A kezdeti és vég populációt alkotó egyedeket a 109 db gázkút jelenti. Minden egyes kúthoz tárolni kell a hozzá tartozó:

- Azonosítót **kútszám,**
- rezsím minimumot **minimum,**
- pillanatnyi beállítási értéket **share (SV),**
- rezsím maximumot **maximum,**
- engedélyezési állapotot **tiltás,**
- csoportba tartozást **group,**
- pillanatnyi mennyiséget **process (PV),**
- nyitási helyzet értékeket **open (%),**
- pillanatnyi jósági értéket **fitness,**
- a csoportnak a rendelkezésre állási állapotát **ready.**

26	kútszám	minimum	share (SV)	maximum	tiltás	group	process (PV)	open (%)	fitness	ready
27	Hsz-100	100 000		400 000	1	1	0	0,00	1	
28	Hsz-101				0		0	0,00	1	
29	Hsz-102	90 000		180 000	1	3	0	0,00	1	
30	Hsz-103				0		0	0,00	1	
31	Hsz-104				0		0	0,00	1	
32	Hsz-105	80 000		280 000	1	3	0	0,00	1	
33	Hsz-106				0		0	0,00	1	
34	Hsz-107				0		0	0,00	1	
35	Hsz-108				0		0	0,00	1	
36	Hsz-109				0		0	0,00	1	
37	Hsz-110				0		0	0,00	1	
38	Hsz-111				0		0	0,00	1	
39	Hsz-112				0		0	0,00	1	
40	Hsz-113	100 000		400 000	1	1	0	0,00	1	
41	Hsz-114	100 000		300 000	1	2	0	0,00	1	
42	Hsz-115	70 000		200 000	1	1	0	0,00	1	
43	Hsz-116	80 000		130 000	1	3	0	0,00	1	
44	Hsz-117	100 000		400 000	1	1	0	0,00	1	
45	Hsz-118	100 000		360 000	1	2	0	0,00	1	
46	Hsz-119				0		0	0,00	1	
47	Hsz-120				0		0	0,00	1	
48	Hsz-121				0		0	0,00	1	

4.1 ábra A kezdeti populáció részlete

A beállítási (SV) érték az algoritmus kimenete, egyben a diagram forrásadata. A 109 db gáz-kút nem mindegyike termel egy adott – besajtolási vagy kitermelési – ciklusban. Üzem közben is szükség lehet egy kút letiltására (például homokkihordás esetén), illetve üzembe helyezésére. A pillanatnyi (PV) értéket és a szelepek nyitási (%) értékét – természetesen – csak kézzel lehet modellezni ebben a kísérleti környezetben. Ezeket az adatokat célszerűen a „Befutósor.xls” *GasWells* munkalapján tárolom. Ennek részlete látható a 4.1 ábrán.

Az algoritmus számára – a kezdeti populációból –csak azok az egyedek érdekesek, amelyek engedélyezett állapotban vannak. Az algoritmus a Population tömbbe gyűjti ezek:

- megnevezését strWellName,
- rezsim minimum értékét lngRezsiMin,
- pillanatnyi beállítási értékét lngShareValue,
- rezsim maximum értékét lngRezsiMax,
- csoportba tartozását bytGroup,
- pillanatnyi mennyiségét lngProcessValue,
- nyitási helyzet értékeit sngOpen,
- pillanatnyi fitness értékét sngFitness,
- a csoportjától függő rendelkezésre állási állapotát strReady.

A VBA-ban ezt a kigyűjtés a következő algoritmus végzi:

```

bytJ = 0
For bytI = 27 To 135
    If Cells(bytI, 5) = 1 Then
        bytJ = bytJ + 1
        ReDim Preserve Population(1 To bytJ)
        With Population(bytJ)
            .strWellName = Cells(bytI, 1)
            .lngRezsiMin = Cells(bytI, 2)
            .lngShareValue = Cells(bytI, 3)
            .lngRezsiMax = Cells(bytI, 4)
            .bytGroup = Cells(bytI, 6)
            .lngProcessValue = Cells(bytI, 7)
            .sngOpen = Cells(bytI, 8)
            .strReady = Cells(bytI, 10)
            .sngFitness = Cells(bytI, 9)
        End With
    End If
Next bytI

```

Itt nyílik lehetőség annak leellenőrzésére, hogy egy adott kútcsoport le van-e zárva teljesen.

```
        If .bytGroup = 1 And .IngProcessValue > 0 Then
            blnFirstZERO = False
        ElseIf .bytGroup = 2 And .IngProcessValue > 0 Then
            blnSecondZERO = False
        ElseIf .bytGroup = 3 And .IngProcessValue > 0 Then
            blnThirdZERO = False
        End If
    End With
```

Amelyik kút viszont le van tiltva, de mégis nyitva van, azt - gyorsan - le kell zárni.

```
        ElseIf Cells(bytI, 3) <> "" Then
            If Cells(bytI, 3) > 5000 Then
                Cells(bytI, 3) = Cells(bytI, 3) - 5000
            Else
                Cells(bytI, 3) = ""
            End If
        End If
    Next bytI
    bytPopulationPIECE = bytJ
```

A Population tömböt dinamikus tömbként definiáltam, mert így mindig csak akkora méretű, mint amennyi az engedélyezett kutak száma.

Az tömb méretét – a későbbi eljárások számára – a bytPopulationPIECE nevű, az egész projektben látható globális változóban eltároltam.

4.2 *Fitnessfüggvény és kiértékelése*

A jósági érték meghatározása az alapján történik, hogy az adott mennyiségváltoztatási irányba mekkora tartalékkal rendelkezik az adott szelep. A tartalékot a szelepnnyitás százalékos értékéből és a szállított mennyiségnek a rezsim értéken belül elfoglalt értékéből lehet megállapítani. Tehát ha a mennyiséget növelni kell, akkor az a szelep legalkalmasabb erre, amelyik

- nulla százalékos nyitásban van és
- a szállított mennyisége is nulla.

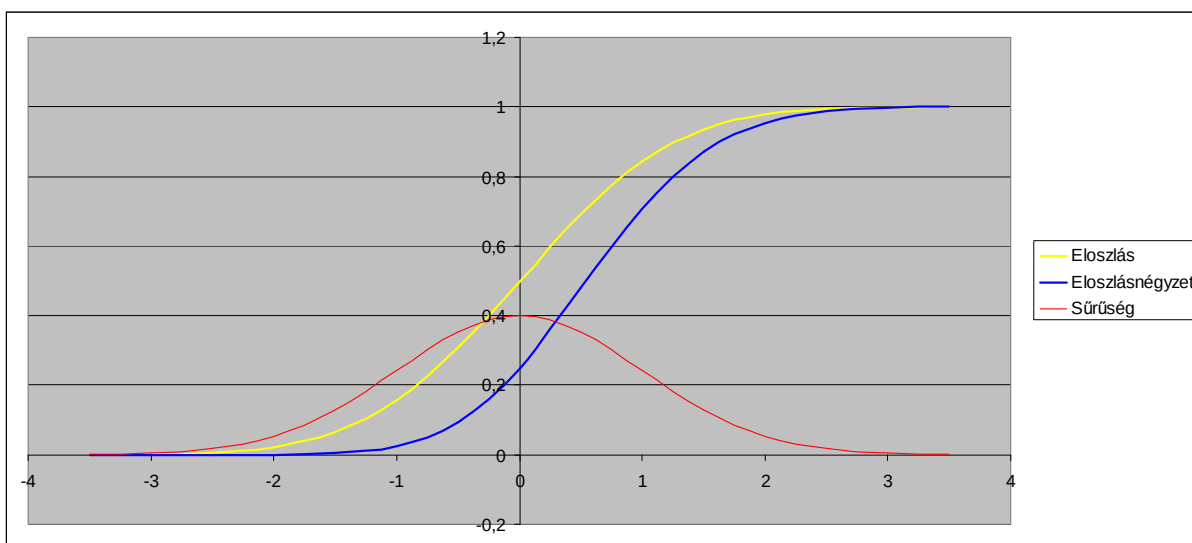
Természetesen, ha a két érték bármelyike maximális értékű, az adott szelep alkalmatlanná válik a mennyiség növelésére.

Ennek megfelelően, ha a mennyiséget csökkenteni kell, akkor az a legmegfelelőbb, amelyik

- száz százalékos nyitásban van és
- a szállított mennyisége a rezsimum (vagy a fölötti).

Ekkor is igaz, hogy amennyiben a két érték közül valamelyik nulla, akkor az adott szelep alkalmatlan a zárási irányba való mozgásra.

A két szélsőérték közötti átmenet biztosítására a normális eloszlás eloszlásfüggvényének görbét találtam megfelelőnek. Ezt az eloszlásfüggvény mutatja sárga színnel a 4.2 ábra „0” várható érték, „1” szórás érték és $-3,5 \leq x \leq 3,5$ tartomány esetére.



4.2 ábra A normál eloszlás függvényei

Az MS Excel a NORM.ELOSZL függvényt a következő számolási móddal kínálja:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{ választási lehetőséget adva arra, hogy a megadott paraméterek}$$

alapján a sűrűségfüggvényt, vagy az eloszlásfüggvényt adja vissza. A választás egy – a képletben nem szereplő – paraméter „IGEN” illetve „NEM” értékének megadásával lehetséges.

Maga az eloszlásfüggvény $\lim_{x \rightarrow -\infty} F(x) = 0$ és $\lim_{x \rightarrow +\infty} F(x) = 1$ közötti értéket vesz fel. Az adott esetben – mivel két feltétel határozza meg a valós jósági értéket – két eloszlásfüggvény szorzatát kell előállítani. Erre a fenti értékeket felvevő eloszlásfüggvény a legmegfelelőbb, mert az eredő érték továbbra is az eredeti határértékek között lesz. Az eloszlásfüggvény négyzetének görbéje a 4.2 ábrán kék színnel látható.

A szorzás miatt a görbe alakja az eredetihez képest torzul. Ezért az értelmezési tartományt némiképpen módosítani kell, hogy csak a görbe hasznos része legyen kihasználva.

A görbének az ábrázolt formájára akkor van szükség, amikor csökkenteni kell a mennyiséget, mivel ilyenkor nagy mennyiség esetén „1” közelében, kis mennyiség esetében „0” körül kell lennie a fitness értéknek.

Amikor a mennyiséget emelni kell, akkor a fitness értéket ennek a görbének a függőleges tengelyre tükrözött változatával kell meghatározni.

A generált fitness értékek kihasználhatóságának érdekében külön értékeket kell generálni a kút készenléti – READY – állapotára és az azt megelőző állapotra.

A táblázat huszonhetedik sorában a fitness értéket kiszámoló képlet:

```
=HA(VAGY($C27=""; $C27=0; $H27=""; $E27=0); 1; HA(J27="READY";  
HA($G$136>$M$8; NORM.ELOSZL(3-5*($D27-$C27)/$D27; 0; 1; IGAZ)*NORM.ELOSZL(($H27-  
40)/20; 0; 1; IGAZ); (1-NORM.ELOSZL(2-5*($D27-$C27)/$D27; 0; 1; IGAZ))*(1-NORM.ELOSZL(($H27-  
60)/20; 0; 1; IGAZ)));  
HA($G$136>$M$8; NORM.ELOSZL(3-5*($B27-$C27)/$B27; 0; 1; IGAZ)*NORM.ELOSZL(($H27-  
40)/20; 0; 1; IGAZ); (1-NORM.ELOSZL(2-5*($B27-$C27)/$B27; 0; 1; IGAZ))*(1-NORM.ELOSZL(($H27-  
60)/20; 0; 1; IGAZ))))
```

A fentiek figyelembevételével négyféle fitness érték kiszámítása történik az Excel beépített függvényeivel:

1. Mennyiség emelés a READY állapot előtt:
 - Értelmezési tartomány: $[0 \leq SV \leq \text{minimum}] \Rightarrow [-3 \leq X \leq 2]$
 - Képlet: $(1 - F(x)) \cdot (1 - F(x))$
2. Mennyiség csökkentés a READY állapot előtt:
 - Értelmezési tartomány: $[0 \leq SV \leq \text{minimum}] \Rightarrow [-2 \leq X \leq 3]$
 - Képlet: $F(x) \cdot F(x)$
3. Mennyiség emelés a READY állapot után:
 - Értelmezési tartomány: $[0 \leq SV \leq \text{maximum}] \Rightarrow [-3 \leq X \leq 2]$
 - Képlet: $(1 - F(x)) \cdot (1 - F(x))$
4. Mennyiség csökkentés a READY állapot után:
 - Értelmezési tartomány: $[0 \leq SV \leq \text{maximum}] \Rightarrow [-2 \leq X \leq 3]$
 - Képlet: $F(x) \cdot F(x)$

Mind a négy esetben a szorzatfüggvény egyik tagja a szelepnnyitás $0 \leq \text{open}(\%) \leq 100$ tartománya a megfelelő értékre átskálázva.

Az átskálázás módja mindkét esetben a következő:

$$\frac{x_{min} - x_{max}}{y_{min} - y_{max}} = \frac{X - x_{max}}{Y - y_{max}} \Rightarrow Y = y_{max} + \frac{(X - x_{max})(y_{min} - y_{max})}{x_{min} - x_{max}}$$

4.3 Szelekció

A szelekció a fitnessérték alapján történik a szabályzás alkalmával. Amikor egy kútcsoport felfuttatása a rezsim minimum értékre történik, nem pedig mennyiségre, akkor nem szabályozás, hanem csak vezérlés folyik.

A szelekciót több lépcsőben kell értelmezni.

- Csak a termelésre kijelölt kutak vesznek részt a termelésben. Eleve csak ezek kerülnek a kezdeti populációba. A kezdeti populáció felállításával kezdődik minden ciklus, így a menetközben letiltott kutak a következő ciklusban már nem vesznek részt.
- Az adott – tárolói – mennyiség meghatározza, hogy mely kútcsoportoknak kell részt venni a mennyiség szétosztásában, így csak azok kerülnek a szabályozandó kutak közé. A Choose eljárás az „X” paraméterben megadott (vagy annál kisebb csoportú) kutakat átmásolja a kezdeti populációból a Selection_POOL tömbbe.

```
Sub Choose(X As Byte, Y As Boolean, Z As Boolean)
    bytJ = 0
    For bytI = 1 To bytPopulationPIECE
        If Y = True And Z = False Then
            If Population(bytI).bytGroup = X Then
                bytJ = bytJ + 1
                ReDim Preserve Selection_POOL(1 To bytJ)
                Selection_POOL(bytJ) = Population(bytI)
            End If
        ElseIf Y = False Or Z = True Then
            If Population(bytI).bytGroup <= X Then
                bytJ = bytJ + 1
                ReDim Preserve Selection_POOL(1 To bytJ)
                Selection_POOL(bytJ) = Population(bytI)
            End If
        End If
    Next bytI
    bytSelectionPIECE = bytJ
End Sub
```

- A mindenkori legjobb 5 illetve 10 kút előírt mennyisége módosul minden ciklusban. Ezt biztosítja a Selection eljárás, amely minden – az „X” paraméterben megadott - rekordot átmásol a Selection_POOL-ból a Mating_POOL nevű tömbbe.

```

Sub Selection(X As Byte)
    For bytI = 1 To X
        ReDim Preserve Mating_POOL(1 To bytI)
        Mating_POOL(bytI) = Selection_POOL(bytI)
    Next bytI
    bytMatingPIECE = bytI - 1
    Cells(14, 13) = bytMatingPIECE
End Sub

```

- Mivel a legjobb fitnessértékű egyedeket kell kiválogatni, a másolás előtt nagyságrend szerint csökkenős sorrendbe kell rendezni a Selection_POOL tömböt. Ezt az $O(n \ln(n))$ bonyolultságú Kupacrendező algoritmussal végeztetem el [8].

4.4 Mutáció

A mutációt a – az eljárások között központinak mondható – Control nevű eljárás végzi. Az eljárás kezdetben meghívja – a már ismertetett, illetve említett – Choose, KupacSort és Selection eljárásokat. Valamint meghívja még a SetDirection eljárást, amely a mindenkori felfutási meredekség beállításához szükséges mutációs lépésközt állítja be.

```

Sub Control(X As Byte, Y As Byte, Z As Boolean, V As Boolean)
    Choose X, Z, V
    KupacSort Selection_POOL, bytSelectionPIECE
    Selection (Y)
    SetDirection

```

A szükséges lépésköz kiszámításához szükség van

- a pillanatnyi mennyiség,
- a szabályozásban éppen részt vevő kutak számának,
- a felfuttatásban résztvevő kutak számának,
- a ciklus lefutási idejének ismeretére.

A szabályozás szempontjából lényeges tisztában lenni azzal, hogy a felfuttatás egy olyan „erőszakos” folyamat, amely figyelmen kívül hagyja, hogy a célfüggvény éppen teljesült-e, csak a számára beállított cél elérésére törekszik.

Megjegyzés: A programban használt felfuttatási idők a szimulációhoz igazítottak. A valóságban a felfuttatási időket napokban kell mérni, tehát az időzítési értékeket át kell számolni egy esetleges valós alkalmazás esetén.

Az eljárás részlete:

```
If Cells(11, 13) >= 6000000 And blnThirdReady = True And blnThirdZERO = False Then
    intD = (180000 * sngT - bytGrownPIECE * 1000) / bytControlIN
End If

If Cells(11, 13) > 10000000 And blnThirdReady = True Then
    intD = (180000 * sngT - bytGrownPIECE * 1000) / bytControlIN
End If
```

A Control eljárás következő vizsgálata a célfüggvény teljesülésére vonatkozik. Erre azért van szükség, mert a mennyiséget változtatni csak akkor kell, ha a célfüggvény nem teljesült.

```
If blnQReady = False Then
    For bytI = 1 To bytMatingPIECE
        With Mating_POOL(bytI)
            If Cells(8, 13) > Cells(11, 13) Then
                If .IngProcessValue + intD <= .IngRezsiMax Then
                    .IngShareValue = .IngShareValue + intD
                    If Z = True Then
                        blnFlag = True
                    End If
                End If
            ElseIf Cells(8, 13) < Cells(11, 13) Then
                If .IngProcessValue - intD >= 0 Then
                    .IngShareValue = .IngShareValue - intD
                    If Z = True Then
                        blnFlag = True
                    End If
                Else
                    .IngShareValue = 0
                End If
            End If
        End With
    Next bytI
End If
```


A továbbiakban meg kell különböztetni azt az állapotot, amikor a pillanatnyi mennyiség kisebb, illetve azt, amikor nagyobb, mint az előírt mennyiség, hisz a két állapothoz eltérő előjelű változtatásra van szükség. Az If Then elágazás IGAZ ágán a mennyiség emelésére, az Else ágán pedig a csökkentésre vonatkozó algoritmus van. Mindkét állapot csak akkor végez változtatást, ha a változtatás mértékével nem történik határérték túllépés. A blnFlag az eljárásban segédváltozóként szerepel.

Az eljárás a szabályozási rész végén a visszahelyezést hívja meg.

Replace Population(), bytPopulationPIECE, Mating_POOL(), bytMatingPIECE

Végezetül itt kapott helyet az a korlát, amely – a rezsim utasításnak megfelelően - megakadályozza a 6M Nm³/nap mennyiségnél kisebb mennyiség beállítását, ha mind a három kútcsoport üzemel.

<pre> If blnTenMillion = True Then If Cells(8, 13) < 6000000 Then Cells(8, 13) = 6000000 End If End If End Sub </pre>
--

4.5 Visszahelyezés

A módosított adatokkal rendelkező egyedek visszahelyezését a kezdeti tömbbe a Replace eljárás végzi. A visszahelyezésnél a kútnév alapján azonosítja az egyedeket, és csak azokat a rekordokat írja felül, ahol a szétosztási érték változott.

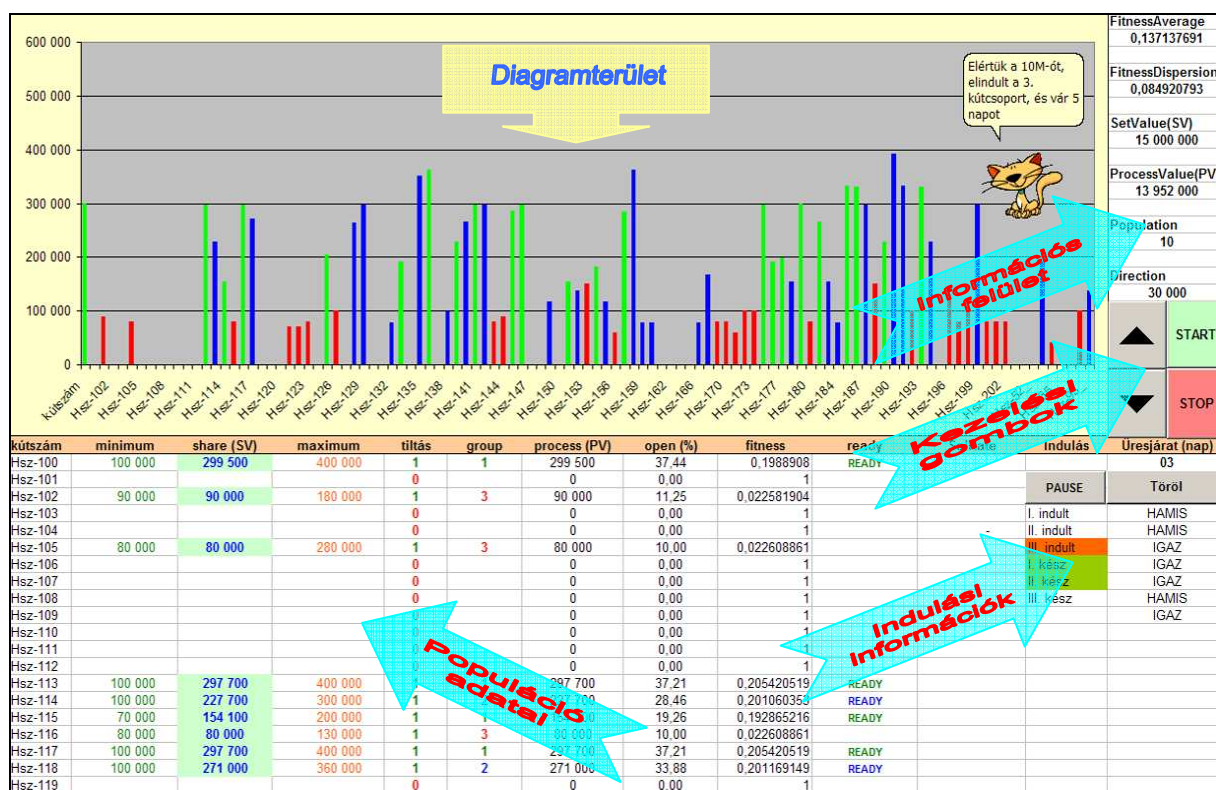
<pre> Sub Replace(X() As GasWell, Z As Byte, Y() As GasWell, V As Byte) For bytI = 1 To V bytJ = 1 Do If Y(bytI).strWellName = X(bytJ).strWellName_ And Y(bytI).lngShareValue <> X(bytJ).lngShareValue Then X(bytJ) = Y(bytI) End If bytJ = bytJ + 1 Loop While Y(bytI).strWellName <> X(bytJ - 1).strWellName Next bytI End Sub </pre>

5 Implementáció

A megvalósított algoritmusnak az evolúciós stratégiához kapcsolódó részeit az előző fejezetben bemutatam. Külön alfejezetet szentelek a megjelenítési és kezelési felületnek, valamint a program egyéb összetevőinek és az alkalmazott módszereknek.

5.1 A megjelenítési és kezelés felület

Amint az előzőekben említettem, a program adatbeviteli és megjelenítési felülete egy Excel tábla. Ugyan az ez a felület egyben kezelői felületként is funkcionál. Ennek a felületnek egy részlete látható az 5.1 ábrán.



5.1 ábra A program kezelői, adatbeviteli és megjelenítési felülete

Az algoritmus meneti és egyben kimeneti felülete a *Populáció adatai* felirattal jelzett táblázatos rész az 5.1 ábrán. A kimeneti értékeket jeleníti meg az oszlopdiagram, melyen külön színnel jelöltem az egyes kútcsoportokat:

- Zöld: 1-es kútcsoport
- Kék: 2-es kútcsoport
- Piros: 3-as kútcsoport

A – bal oldali – értéktengely beosztása Nm^3/h -ban értendő. A vízszintes kategóriatengelyen a kútnevek vannak feltüntetve, bár a megjelenítési méret és az olvashatóság kompromisszumaként csak minden harmadik.

A táblázatos részben a cella zöld háttere azt jelzi, hogy az adott kút elérte a – rezsim utasítás szerinti – minimum hozamot. Amennyiben túllépi a maximum értéket – de jelenleg nincs ilyen – a háttér piros színűre változik.

Az 5.1 ábra alsó részén a már bemutatott populáció látható, csak most „üzem közben”.

Indulási információk

Az 5.2 ábrán kiemeltem a megjelenítő felületnek azt a részletét, amely - működés közben - információkat ad a kútcsoportok állapotáról, illetve az indulás módjáról.

Indulás	Üresjárat (nap)
CSÖKKENTETT	04
PAUSE	Töröl
I. indult	HAMIS
II. indult	IGAZ
III. indult	HAMIS
I. kész	IGAZ
II. kész	HAMIS
III. kész	HAMIS
	IGAZ

5.2 ábra Információk az indulásról

Az *Indulás* felirat alatt közvetlenül az indulás módja jelenik meg (CSÖKKENTETT vagy NORMÁL), a PAUSE gomb alatt pedig piros háttérrel az éppen felfuttatás alatt lévő kútcsoportok, zöld háttérrel pedig a felfuttatott és a – rezsim utasítás szerinti – pihenőidőn túl esett kútcsoportok lettek megjelölve.

Az *Üresjárat (nap)* felirat alatt az éppen pihentetési állapotban lévő csoport eltelt pihenőnapjainak száma olvasható le.

Megjegyzés: A *Töröl* gomb alatt segédváltozók információi láthatóak, az indulási állapot színezését vezérlik a MS Excel *feltételes formázás* szolgáltatásával.

PAUSE és Töröl gombok

A PAUSE gombbal fel lehet függeszteni a program futását. Ilyenkor program nem áll le, a változók állapota megmarad. Újabb megnyomása esetén a program ugyan onnan folytatódik.

A *Töröl* gomb egy törlési rutint indít el, a bemeneti értékeket nullázza le.

Megjegyzés: Erre a két funkcióra csak a program fejlesztésekor van szükség.

Office segéd

A program futásának mindenkor nyomon követéséhez az *Office segéd*et választottam, amely a VBA felületről vezérelhető. Az *Office segéd* folyamatos információkat szolgáltat az éppen aktuális állapotról, várható – figyelendő – eseményekről. A segéd a program indulásakor megjelenik, a végén eltűnik. Egyébként pedig az újabb üzeneti felülírják az előzőeket, mivel a segéd csak egy példányban tud egyszerre megjelenni [9].

A segéd elhelyezésére a diagram jobb felső sarkát választottam, mert közel van az információs felülethez, de lényeges nem zavarja a megjelenítést.

START és STOP gombok

A program indítását, leállítását és a kíván mennyiség változtatását a diagram mellett – jobbra – található, az 5.1 ábrán *Kezelési gombok* feliratú nyíllal jelölt gombok végzik.

A program a START gombbal indítható el, mely egy eseményvezérelt eljárást indít el, a STOP gomb a főciklus leállási feltételét adja.

Az előírt mennyiség növelése és csökkentése az előbbi nyomógombok mellett található két léptető nyíllal történik.

A program futása közben vannak olyan esetek, amikor ezeknek a léptető nyilaknak az utasításait a program felülbírálja. Ekkor a következő értéke érvényesülnek:

- a mennyiség kezdeti felfuttatása:
 - o CSÖKKENTETT indulás esetén 1.920.000,
 - o NORMÁL indulás esetén 4.800.000,
- teljes kútállomány esetén – amikor mind a három kútcsoporthoz üzemel -, a minimális mennyiség 6.000.000.

Információs felület

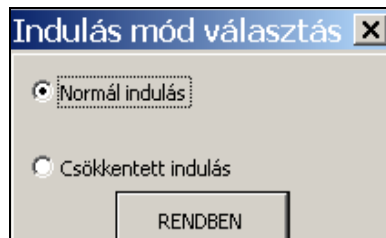
A diagramtól jobbra eső oszlopban információkat kapunk a következő állapotokról:

- | | |
|---|--------------------|
| • a fitness értékek átlaga | FitnessAverage, |
| • a fitness értékek szórása | FitnessDispersion, |
| • a beállított – tárolóra vonatkozó – mennyiség | SetValue(PV), |
| • a pillanatnyi – összes – mennyiség | ProcessValue(PV), |
| • a szabályozásban részt vevő kutak száma | Population, |
| • a pillanatnyi mennyiségváltoztatási sebesség | Direction. |

Párbeszéd ablakok

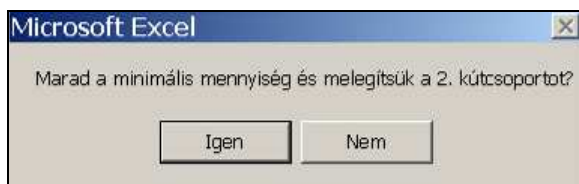
A program elindítása után kérdés jelenik arra vonatkozóan, hogy NORMÁL vagy CSÖKKENTETT indulási móddal indítjuk a tárolót.

Ezt a kérdés egy űrlap teszi fel, mely akkor jelenik meg, ha a tároló még nem lett elindítva. Az 5.7 ábrán egy kétgombos választócsoporth bejelölésével választhatunk.



5.3 ábra Az indulási mód választása

A többi – a teljesítményemelésre vonatkozó - üzenet és választási lehetőség üzenetdobozokban (MsgBox) jelenik meg, melyeket a program futása közben hív meg, s nem fut tovább a válaszáig. Az üzenetdoboz látható az 5.8 ábrán.



5.4 ábra Az Üzenet Panel

5.2 Módszerek

PAUSE gomb

A program futásának felfüggesztését a VBA eseményvezérelt eljárásával oldottam meg, a nyomógomb megnyomásához kötötten.

```
Private Sub PAUSE_Click()  
    If blnPAUSE = False Then  
        blnPAUSE = True  
    ElseIf blnPAUSE = True Then  
        blnPAUSE = False  
    End If  
    Do While blnPAUSE = True  
        DoEvents  
    Loop  
End Sub
```

Tulajdonképpen a program egy ciklusba kerül, melynek belsejében csupán egy eseményfigyelés történik, a ciklusból való kilépés feltétele pedig a gomb újbóli megnyomása.

Töröl gomb

Ez a rutin a START gomb megnyomásával is elindul, hogy a bemeneti értékek alapállapotában induljon a program. A rutin semmi mást nem csinál, mint a *Share (SV)* és *ready* oszlopokat üres értékekkel feltölti.

```
Sub Töröl()  
    Range("C27:C135") = ""  
    Range("J27:J135") = ""  
End Sub
```

Office segéd

A segéd megjelenítése és elrejtése érdekében a programban a központi ciklust alkotó Do Loop While előtt és után áll a következő két sor:

- Assistant.Visible = True
- Assistant.Visible = False

A segéd üzeneteit a Message paraméterezett eljárás vezérli.

```
Sub Message(X As String)  
    Set blnMess = Assistant.NewBalloon  
    With blnMess  
        .Text = X  
        .Mode = msoModeModeless  
        .Callback = ""  
        .Button = 0  
        .Show  
    End With  
End Sub
```

Az eljárás az *X* paraméterben kapott szöveget jeleníti meg a segéd üzenetbuborékjában. A segédet a következőképpen állítottam be:

- A paraméterként megkapott sztringet jelenítse meg üzenetként,
- ne várjon választ,
- ne végezzen műveletet,
- ne legyen rajta gomb,
- jelenjen meg.

Összesen 12 szöveges üzenetet fogalmaztam meg a segéd számára.

- Text_1 = "NORMÁL indítás. Indult az 1. és 2. kútcsoport. Elérik a 4,8M-t és várnak öt napig"
- Text_2 = "Az első és második kútcsoport szabályozható. 10M-nál indulnia kell a 3. kútcsoportnak"
- Text_3 = "Elértük a 10M-t, elindult a 3. kútcsoport, és vár 5 napot"
- Text_4 = "Mindhárom kútcsoport szabályozható, 6M alá nem lehet menni"
- Text_5 = "CSÖKKENTETT indulás. Indult az 1. kútcsoport. Eléri az 1,92M-t és vár öt napig"
- Text_6 = "CSÖKKENTETT indulás. Indult a 2. kútcsoport. Eléri az 1,92M-t és vár öt napig, miután az 1. csoport lezárt"
- Text_7 = "CSÖKKENTETT indulás. Indult a 3. kútcsoport. Eléri az 1,92M-t és vár öt napig, miután a 2. csoport lezárt. Ezután az 1. és 3. helyet cserél, az első 12 órát pihen"
- Text_8 = "Az első kútcsoport szabályozható. 4M-nál indulnia kell a 2. kútcsoportnak"
- Text_9 = "Elértük a 4M-t, elindult a 2. kútcsoport, és vár 12 órát"
- Text_10 = "Elértük a 10M-t, elindult a 3. kútcsoport, és vár 12 órát"
- Text_11 = "Elértük a 4M-t, elindult a 2. kútcsoport, és vár 5 napot"
- Text_12 = "Az 1. és 2. helyet cserél, az első 12 órát pihen."

A program a futása során a megfelelő helyeken a kívánt üzenetekkel paraméterezve hívja meg a Message eljárást. Például: Message Text_2

A kezelő gombok

A cmdStart_Click() eseményvezérelt eljárás a STOP gombot kezeli - hogy elkerüljük a futó program újbóli elindítását -, elindítja a Töröl majd a GasWells eljárást.

```
Private Sub cmdStart_Click()  
    cmdStop_Click  
    Call Töröl  
    Call GasWells  
End Sub  
Private Sub cmdStop_Click()  
    blnStop = True  
End Sub
```

A STOP megnyomása a fő ciklusból való kilépés feltétele.

```
Loop While Not blnStop
```

A léptető nyilak objektum elnevezése *SpinButton2*. Az értéket az M8 jelű cellába küldik ki (LinkedCell M8).

(Name)	SpinButton2
AutoLoad	False
BackColor	 &H8000000F&
Delay	50
Enabled	True
ForeColor	 &H80000012&
Height	103,5
Left	838,5
LinkedCell	M8
Locked	True
Max	23090000
Min	0
MouseIcon	(None)
MousePointer	0 - fmMousePointerDefault
Orientation	-1 - fmOrientationAuto
Placement	3
PrintObject	True
Shadow	False
SmallChange	100000
Top	216,75
Value	4800000
Width	45

5.5 ábra A léptető nyilak tulajdonságai

A léptető nyilak a mennyiséget:

- Minimum nulla (Min 0),
- maximum 4.800.000 (Max 4800000) értékek között tudják beállítani, 100.000-es lépésekben (SmallChange 100000).

Az 5.5 ábrán az aktuálisan beállított érték 4.800.000 (Value 4800000).

Információs felület

Az információs felületen kapott különböző adatok közül kettő számítás eredménye, a többi az algoritmus kimeneteként képződik.

A fitness átlag értékét és a szórását az Excel beépített függvényeivel számoltattam ki, mivel a számításhoz szükséges bementi értékek a táblázatban megtalálhatóak, valamint a kiszámított értékre is csak a táblázatban van szükség.

HA(I138<>0;AB.ÁTLAG(I26:I135;1;N1:N2);"")
 HA(I138<>0;AB.SZÓRÁS(I26:I135;1;N1:N2);"")

Az Excel az átlag számítására az $\bar{x} = \frac{\sum x}{n}$ képletet használja, ahol:

- $\bar{x}$ a minta átlaga,
- x a fitness értékek,
- n pedig a minta értékek számát jelenti.

A szórás kiszámítása pedig a következő képlettel történik: $s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}}$ ahol az „s” a szórási jele.

A jelen esetben viszont csak azoknak az értékeknek az átlaga, illetve a szórása az érdekes, melyek részt vesznek a mennyiség előállításában. Ezért a feltételeket úgy adtam meg, hogy az AB függvényekben csak az üzemelő kutak fitness értéke szerepeljenek.

Űrlap

Az indulási mód kiválasztása a *Selection* nevű űrlapon történik. Az űrlap két választógombot, és egy nyomógombot tartalmaz. Az űrlap a következő utasítássor második sorára jelenik meg, majd a választás után el is tűnik. Az űrlap az 5.3 ábrán látható.

```
If blnSTARTed = False Then
    Selection.Show
    If Cells(12, 16) = True Then
        Cells(27, 12) = "NORMÁL"
        Cells(8, 13) = 4800000
    ElseIf Cells(13, 16) = True Then
        Cells(27, 12) = "CSÖKKENTETT"
        Cells(8, 13) = 1920000
    End If
End If
```

A csoport gombjai különböző segédcellákat jelölnek meg True értékkel, majd e segédcellák állapotától függően folytatódik tovább a program.

```
Private Sub UserForm_Initialize()
    Normál.Value = Cells(12, 16)
    Csökkentett.Value = Cells(13, 16)
End Sub
```

MsgBox

Az MsgBox a VBA-ban meghívható eljárásként, vagy függvényként. Eljárásként meghívva csak tájékoztatásra használható, míg függvényként visszatérési értéke van. Ebben a programban a függvényszerű meghívást alkalmaztam, a visszatérési értéket pedig elágazásban használtam fel.

A lenti programrészlet szerint történik - CSÖKKENTETT indulás közben -, az első kútcsoport felfutása után az MsgBox meghívása. A hívás függvényként történik, a kérdésre adott választól függően a függvény visszatérési értéke 6 vagy 7 lesz. A program további futása a bytFlagFirst változó értékétől függ.

```
If bytFlagFirst = 0 And blnFirstReady = True Then  
    bytFlagFirst = MsgBox("Marad a minimális mennyiség és melegítsük a 2. kútcsoportot?", vbYesNo)  
End If
```

Amennyiben *Igen* választ adtunk, akkor a változó értéke 6, amennyiben *Nem* választ, akkor pedig 7. A változó értéke egy elágazás feltételében szerepel. Az MsgBox az 5.4 ábrán látható. A paraméterként szereplő vbYesNo vezérli a kétgombos megjelenést, valamint azt, hogy kötelező választ adni a kérdésre.

Az egyedek

Az algoritmus alapja a kutak adatainak tárolása egy struktúrában. A VBA *Felhasználói adat-típus* néven említi azt az adatformát, amelyben különböző típusú adatokat tudunk tárolni, és a Type és End Type sorok között kell deklarálni az elemeit, de a két utasítás között csak deklaráló utasítások lehetnek [9].

```
Type GasWell  
    strWellName As String * 7  
    lngRezsiMin As Long  
    lngShareValue As Long  
    lngRezsiMax As Long  
    bytGroup As Byte  
    lngProcessValue As Long  
    sngOpen As Single  
    sngFitness As Single  
    strReady As String * 5  
End Type
```

Az egyes elemek deklarálásakor igyekeztem a változó nevekben utalni azok típusára illetve funkciójára is. Ennek megfelelően a kutak adatai a GasWell típusban:

- strWellName As String * 7 (az adott kút neve, 7 elemű szöveggént tárolva)
- lngRezsiMin As Long (a rezsim minimum százezres nagyságrendje csak a 4 bájtos Long típusban fér el)

- lngShareValue As Long (a pillanatnyi megosztási érték szintén százezres nagyságrendű)
- lngRezsMax As Long (a rezs maximum értéke is százezres nagyságrendű)
- bytGroup As Byte (a csoporttagság értéke eléri a legkisebb egész értékű tárolóban)
- lngProcessValue As Long (a pillanatnyi kimeneti érték szintén százezres nagyságrendű)
- sngOpen As Single (a szelep nyitás százalékos értéke lehet tört is, ezért valós adatként kell tárolni, de nem kell nagy pontosság)
- sngFitness As Single (a fitness érték szintén lehet tört)
- strReady As String * 5 (a „READY” szöveg tárolása 5 elemű szöveggént)

A mennyiség felfuttatása

Külön eljárás szolgál arra az esetre, amikor egy kútcsoportot fel kell futtatni egy minimális értékre, vagy a rezs minimumra. Ezekben az esetekben az adott kútcsoport mennyiség növelése független attól, hogy a célfüggvény éppen teljesült vagy sem. A felfuttatást végre kell hajtani, még annak árán is, hogy ezzel más kútcsoportok mennyiségét csökkenteni kell. A felterhelést a GrownUp paraméterezhető eljárás végzi.

Sub GrownUp(X As Boolean, Y As Byte, Z As Boolean)
--

- Az X paraméter visszajelzést ad – az eljárás végén – arról, hogy éppen felterhelés folyik valamelyik kútcsoporton,
- az Y paraméterben a felhozandó kútcsoport számának kell lennie,
- a Z paraméter pedig azt határozza meg, hogy mennyiségre (Z=True), vagy rezs minimumra (Z=False) történjen a felterhelés.

If Z = True Then bytControlN = 5 Control Y, bytControlN, True, False
--

A programrészleten látható, hogy Z=True esetén az eljárás meghívja a Control eljárást:

- átadja neki az Y paraméterben a felhozandó kútcsoport számát,
- a bytControlN paraméterben hogy hány darab kút vegyen részt a szabályozásban,
- a harmadik True paraméter jelzi, hogy mennyiségre kell szabályozni,

- a negyedik False pedig azt, hogy szimpla, azaz csak egy kútcsoportot kell felfuttatni (erre a CHOOSE eljárásnak lesz szüksége).

```

For bytI = 1 To bytPopulationPIECE
    With Population(bytI)
        If .bytGroup = Y And .IngShareValue + 1000 <= .IngRezsMin Then
            .IngShareValue = .IngShareValue + 1000
            bInFlag = True
            bytGrownPIECE = bytGrownPIECE + 1
        End If
    End With
Next bytI

```

Ez a kódrészlet Z=False esetén fut le és 1.000-es lépésekben emeli a kijelölt kutak mennyiségét, természetesen csak akkor, ha ezzel rezsimez értéket nem sért. A bytGrownPIECE változóban visszaadja, hogy hány kúton történt emelés.

Egy kútcsoport lezárása

Egy kútcsoport teljes lezárására olyankor van szükség – a jelen rezsimez utasítás mellett – amikor CSÖKKENTETT indulási szisztémát alkalmazunk, és az első (vagy később a második, kérdésre is) *Igen* választ adunk a párbeszéd panelen. Ilyenkor az algoritmus úgy működik, hogy elindítja a soron következő (második vagy harmadik) kútcsoportot és a mennyiségértéktartás érdekében a nyitott kútcsoportot lezárja.

Megjegyzés: a felfuttatandó kútcsoport pihentetési ideje akkor indul, mikor a lezárandó kútcsoport jelzi a teljes lezárási állapotát a

- BlnFirstZERO,
- BlnSecondZERO,
- BlnThirdZERO változók True értékével.

Egy kútcsoport lezárását a GrownDown eljárás végzi. Az eljárás működése megegyezik a GrownUp eljárásával, azzal a különbséggel, hogy itt a növelés negatív előjelű.

A pihentetési időzítés

Amikor egy kútcsoport lezárt állapotából eléri a minimális mennyiségét, akkor – a rezsimez utasításnak megfelelően – 5 napot vagy 12 órát pihentetni kell a minimális mennyisége közelében. Az időzítést – természetesen a szimulációs környezetnek megfelelő értékekkel – a Boole típusú Timing függvény végzi. False értékkel tér vissza, ha végzett.

A függvény a következő paraméterekkel rendelkezik:

- X Öntartást biztosít a függvény számára. Csak akkor indul el az időzítés, ha az értéke False. A függvény a Timing változóban vissza is adja az értékét.
- Y A függvény hívásakor nullázódik, majd visszaadja az időzítésből eltelt időt, és megjeleníti az indulási információk között. (Az 5.2 ábrán az *Üresjárat (nap)* felirat alatt.)
- Z Ebben a változóban jelzi, hogy az adott kútcsoport elkészült.
- V Az aktuális kútcsoport számát kell benn megadni.
- W Visszajelzés arról, hogy az adott kútcsoport éppen felfuttatásban van-e még.
- T Itt kell megadni a szükséges késletetés idejét.

```
Function Timing(X As Boolean, Y As Date, Z As Boolean, V As Byte, W As Boolean, T As Date)
```

```
    If X = False Then
```

```
        datNow = Now
```

```
        X = True
```

```
    Else
```

```
        Y = Now - datNow
```

```
        Cells(27, 13) = Y
```

```
    End If
```

```
    If Y >= T Then
```

```
        For byt1 = 27 To 135
```

```
            If Cells(byt1, 6) = V Then
```

```
                Cells(byt1, 10) = "READY"
```

```
            End If
```

```
        Next byt1
```

```
        Range("L27:M27") = ""
```

```
        X = False
```

```
        Y = "00:00:00"
```

```
        Z = True
```

```
        W = False
```

```
    End If
```

```
    Timing = X
```

```
End Function
```

A program törzse

Az előzőekben bemutatott algoritmusok a program törzséből kerülnek meghívásra. A programtörzs hasonló egységekből áll, mint az itt következő. Maga az szerkezet – elágazások felhasználásával – a 2.4 ábrán bemutatott logikát követi.

Az itt bemutatandó programrészlet akkor fut le, amikor az első párbeszédpanel kérdésére *Igen* választ adtunk és a második kútcsoport még nincs kész.

```
If bytFlagFirst = 6 And blnSecondReady = False Then
```

Majd az első kútcsoport lezárását és a második kútcsoport felhozatalát indítja.

```
GrownDown (1)  
GrownUp blnSecondSTART, 2, True
```

Amikor a GrownUp a blnFlag False értékével jelzi, hogy a második kútcsoport elérte a mennyiséget, akkor indul a késleltetés, ami ebben az esetben 5 másodpercig tart.

```
If blnFlag = False And blnFirstZERO = True Then  
    blnTimeNULL = Timing(blntimeNULL, datTime, blnSecondReady, 2, _  
                        blnSecondSTART, "00:00:05")  
End If
```

Azon kívül ez a programrészlet gondoskodik arról is, hogy az 5.2 ábra felületén az *Indulás* felirat alatt megjelenjen a CSÖKKENTETT felirat és 1.920.000 legyen az előírt mennyiség.

A ciklusidő

A felfutási meredekség mértékegysége m^3/h^2 . Ennek állandó értéken, illetve előre meghatározott értéken tartását, csak akkor lehet biztosítani, ha a ciklusok azonos idő alatt futnak le. Ez a következő kis programrészlettel oldottam meg.

```
count = 0  
Do  
    count = count + 1  
Loop While Timer < TimeNow + sngT  
Cells(27, 11) = count
```

A programrészlet a Do Loop While ciklus végén található. A TimeNow a ciklus elején - közvetlenül a Do után - eltárolja a ciklus indulásának idejét, az sngT pedig – kísérleti alapon – egyenlő 0,2-vel. Tehát, amíg a program indulásától számított 0,2 másodperc el nem telik, addig nem következhet a ciklus vége. A számláló értékét nyomon lehet követni a megjelenítési felületen. Ha az értéke nulla, akkor az azt jelenti, hogy a ciklus hosszabb, mint 0,2 másodperc, a futása nem determinálható.

Megjegyzés: Amennyiben DCS felületen kerül az algoritmus megvalósításra, erre a funkcióra nincs szükség. A DCS rendszerekben a Task-ok futási ideje előre beállítható.

Az eredmények kiírása

A ciklusok végén az eredményeket ki kell íratni a forrásul szolgáló táblázatba, hogy a diagram mindig az aktuális eredményeket jelenítse meg.

A kép villódzásának, ugrálásnak elkerülése érdekében minden ciklus elején letiltottam a képernyő frissítését a következő utasítással.

```
Application.ScreenUpdating = False
```

A kiírás gyorsítása érdekében csak azokat a cellákat kell frissíteni a táblázatban, amelyek értéke változik.

```
If Population(bytJ).IngShareValue <> Cells(bytI, 3) Then  
    Cells(bytI, 3) = Population(bytJ).IngShareValue  
End If
```

Végül – minden ciklus végén – frissíteni kell a képernyőt.

```
Application.ScreenUpdating = True  
DoEvents
```

A DoEvents a VBA beépített eszköze. Használatával eseményfigyelést lehet elindítani. Jelen esetben a STOP gomb megnyomását kell figyelgetni, mert annak megnyomása a ciklus kilépési feltétele.

Hibakezelés

A hibák futás közbeni kezelésére a VBA beépített hibakezelőjét használtam. Az On Error GoTo Error kifejezés a központi eljárás első sora. A vezérlés hiba esetén ugrik a következő programrészletben található Error sorra.

```
Exit Sub  
Error:  
    Debug.Print Time & "-HIBA-" & Err.Description & " " & Err.Source & " " & Err.Number  
    Resume Next  
End Sub
```

Az Exit Sub sorra azért van szükség, hogy rendes esetben – amikor nincs hiba – itt kilépjen a szubrutin [9].

Amikor viszont hiba van, akkor az *Immediate* ablakba kiírja – a Debug.Print hatására – a hiba: időpontját, szöveges leírását, forrását (azt a vezérlő elemet, amely a hibát kezdeményezte) és azonosító számát.

5.3 Kezelői Utasítás

Az utasítás hatálya: Az „EFS HFGT befutósori kútjainak szabályozása, a 2009/2010 kitermelési ciklusra készített egyedi Rezsim Utasítás alapján” feladatot megvalósító szimulációs program kezelése.

1. Az egyedi rezsim utasítás rezsim minimumra és maximumra vonatkozó értékeit a *Befutósor.xls Data* munkalapjának megfelelő oszlopaiba manuálisan be kell írni. A rezsim utasítás részlete az 5.6 ábrán látható, a *Data* munkalap részlete pedig az 5.7 ábrán.

e-on

Földgáz
Storage

Tárolói Műszaki Tervezés

+

	Egyedi rezsimutasítás a hajdúszoboszlói földalatti gáztároló kútjaira																																								
2009.10.14. Kitermelés/1	1. A földalatti gáztároló kezdő és maximális rezsimei kutanként: <u>Első csoport:</u> <table><tr><th></th><th>Kezdő hozam m³/nap</th><th>Maximális hozam m³/nap</th><th>Cascad kutak</th><th>Elmúlt két cik- lusban homokolt</th></tr><tr><td>Hsz -100</td><td>100 000</td><td>400 000</td><td>*</td><td></td></tr><tr><td>-113</td><td>100 000</td><td>400 000</td><td></td><td></td></tr><tr><td>-115</td><td>70 000</td><td>200 000</td><td></td><td></td></tr><tr><td>-117</td><td>100 000</td><td>400 000</td><td>*</td><td></td></tr><tr><td>-126</td><td>80 000</td><td>270 000</td><td></td><td>#</td></tr><tr><td>-134</td><td>80 000</td><td>250 000</td><td></td><td></td></tr><tr><td>-137</td><td>100 000</td><td>500 000</td><td></td><td>#</td></tr></table>		Kezdő hozam m ³ /nap	Maximális hozam m ³ /nap	Cascad kutak	Elmúlt két cik- lusban homokolt	Hsz -100	100 000	400 000	*		-113	100 000	400 000			-115	70 000	200 000			-117	100 000	400 000	*		-126	80 000	270 000		#	-134	80 000	250 000			-137	100 000	500 000		#
	Kezdő hozam m ³ /nap	Maximális hozam m ³ /nap	Cascad kutak	Elmúlt két cik- lusban homokolt																																					
Hsz -100	100 000	400 000	*																																						
-113	100 000	400 000																																							
-115	70 000	200 000																																							
-117	100 000	400 000	*																																						
-126	80 000	270 000		#																																					
-134	80 000	250 000																																							
-137	100 000	500 000		#																																					

5.6 ábra A rezsim utasítás részlete

Azokat a kutakat, amelyeket az utasítás tartósan nem üzemelőnek jelez, nem kell beírni a táblázatba, de beírásuk gondot nem okoz.

Microsoft Excel - Befutósor			
Fájl Szerkesztés Nézet Beszúrás Formátum Eszközök			
Arial 10 F D A			
F17	fx		
A	B	C	D
1 kútszám	minimum	share (SV)	maximum
2 Hsz-100	100 000		400 000
3 Hsz-101			
4 Hsz-102	90 000		180 000
5 Hsz-103			
6 Hsz-104			
7 Hsz-105	80 000		280 000
8 Hsz-106			

5.7 ábra A Data munkalap részlete

2. A megjelenítési és kezelési felület *tiltás* oszlopában az üzemelésre kijelölt kutakat 1-es értékkel kell megjelölni. Erre mutat példát az 5.8 ábra.

maximum	tiltás	group
400 000	1	1
	0	
180 000	1	3
	0	
	1	

5.8 ábra Az üzemelő kutak megjelölése

Egy kút menet közben is kivehető az üzemelők közül – például homok kihordás miatt -, ilyenkor elegendő ugyanitt nulla értéket adni. Ilyenkor az így jelölt kút mennyisége rövid időn belül nullára csökken.

3. Ugyancsak ezen a felületen a *group* oszlopban fel kell tüntetni az adott kút csoporttagságát is. Ez az érték is legördülő menüből állítható 1-től 3-ig.
4. Ezek után az előkészületek után a programnak csak a *GasWells* lapját kell kezelni. Az algoritmus ezt a lapot teszi aktívvá elindulásakor. Az itt található zöld START gombbal indítható a szimuláció.
5. Ki kell választani az indulás módját: CSÖKKENTETT vagy NORMÁL

A program automatikusan felemeli a mennyiséget 1.920.000 illetve 4.800.000 m³/nap ±0,5%-os környezetébe. Az előírt pihentetési idő után csökkentett módban választani kell, hogy a csökkentett mennyiség marad, vagy mennyiség emelés következik. A program az előírt meredekségeket – a 7.4 ábra a rezsím utasítás részlete – automatikusan tartja.

A kezdeti 5 nap egyenletes kitermelést követően az indulási változatok termelési tartományainak függvényében a következő terhelésváltoztatásokat kell alkalmazni:

- 1,92-4,0 x 10⁶ m³/d termelési ütem tartományban a maximális változtatás mértéke: 3750 m³/óra negyedóránként.
Amikor a gázigény eléri a 4,0 x 10⁶ m³/nap ütemet, akkor a második csoportba sorolt kutakat is be kell indítani, amelyek hozama öt nap kezdő hozamon történő termeltetés után emelhető.

- 4,0-10,0 x 10⁶ m³/d termelési ütem tartományban a maximális változtatás mértéke: 7500 m³/óra negyedóránként, amennyiben a második csoport is túl van az 5 napos kezdő termeltetésen és mindkét kútcsoport együttesen üzemben van.

5.9 ábra A mennyiség emelési sebességek

6. Mennyiség emelés a kezelőgombok mellett található nyilakkal lehetséges. A teljes kútállomány működése esetére a rezsím utasítás külön rendelkezik, lásd 7.5 ábra.

2. A tárolói maximumok és minimumok, valamint a terhelésváltoztatás körülményeit bemutató táblázat a tároló normál (teljes kútállomány termel) működése során

Kitermelési kapacitások				Terhelésváltási adatok		
minimum		maximum		alapjárat	felleterhelés min->max-ra	
Mm ³ /nap	em ³ /óra	Mm ³ /nap	em ³ /óra	nap	em ³ /óra/óra	óra
6,0	250	20,8	867	5	45*; 36**	7,4*+7,9**=15,3

* 6,0-14,0 Mm³/nap termelési tartományban
 ** 14,0-20,8 Mm³/nap termelési tartományban

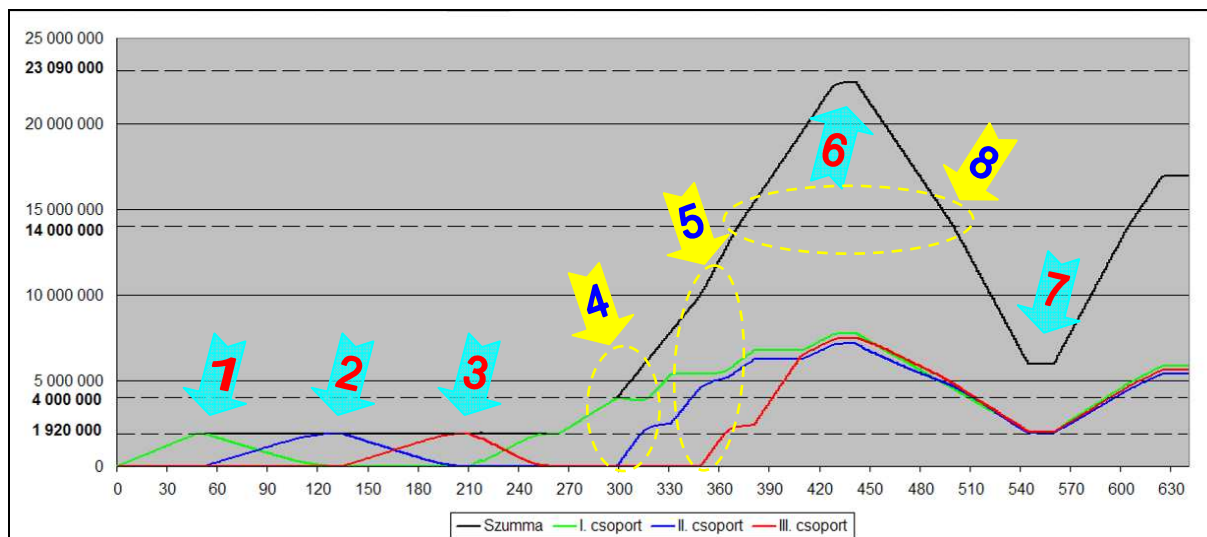
5.10 ábra A változtatási sebességek és mennyiségi határok

Ezeket a változtatási sebességeket az algoritmus automatikusan tartja.

7. A program leállítása a piros STOP gombbal lehetséges.

6 Eredmények

A program kimeneti értékeiről készült trend látható az alábbi, 6.1 ábrán. A függőleges tengelyen a kimeneti mennyiség látható Nm³/nap értékben, míg a vízszintes tengelyen a program indítása óta eltelt idő másodpercben.



6.1 ábra A kimeneti értékek az idő függvényében

A trend demonstrációs céllal készült és egy olyan szimulációs esetet ábrázol, amikor

- CSÖKKENTETT indulás volt,
- mindhárom kútcsoport elő lett melegítve (az MsgBox-ok *Igen* válaszai).

A diagram

- zöld színnel ábrázolja az 1-es kútcsoport,
- kék színnel a 2-es kútcsoport és
- piros színnel a 3-as kútcsoport mennyiségét.
- Fekete színnel az összegzett tárolói mennyiség látható.

A diagramon számozott nyilakkal megjelöltem a jellegzetes futási pontokat.

1-es számú nyíl: azt az időszakot mutatja, amikor az első kútcsoport elért az előírt 1.920.000 érték közelébe, és 5 napot (jelképesen 5 másodpercet) vár. Ez után elindul a késsel jelölt második kútcsoport, az első mennyisége pedig csökken.

2-es számú nyíl: az előzőhöz hasonlóan a 2. kútcsoport felfutását és 5 másodperces várakozását mutatja. A várakozás után a második kútcsoport mennyisége csökken, a harmadik csoport pedig elindul.

- 3-as számú nyíl:* az előző kettőhöz hasonló szituáció, annyi különbséggel, hogy a harmadik kútcsoporthoz várakozási ideje után automatikusan átveszi helyét az első kútcsoporthoz.
- 4-es számú nyíl:* sárga ellipszisben mutatja a következőket:
- Amikor az 1. kútcsoporthoz elérte a $4.000.000 \text{ Nm}^3/\text{nap}$ értéket, akkor elindult a 2. kútcsoporthoz.
 - A mennyiség-növelési sebesség megtartásának érdekében az 1. kútcsoporthoz mennyisége egy ideig csökkent. Ez azért következett be, mert a 2. kútcsoporthoz mindenképpen el akarja érni a minimális mennyiségét, de a meredeksége nagyobb, mint a megengedett.
- 5-ös számú nyíl:* sárga ellipszisében azt a pontot jelöltem meg, amikor
- az összmennyiség eléri a $10.000.000 \text{ Nm}^3/\text{nap}$ értéket,
 - a tervnek megfelelően elindul a 3. kútcsoporthoz.
 - Ugyanakkor az emelkedési sebesség $30.000 \text{ Nm}^3/\text{h}^2$ -ről $45.000 \text{ Nm}^3/\text{h}^2$ -re változik.
- 6-os számú nyíl:* a szállított maximum elérését mutatja. A maximálisan beírható mennyiség $23.090.000 \text{ Nm}^3/\text{nap}$. Ez megegyezik a rezsimumok összegével. A program azonban megállt $22.431.200 \text{ Nm}^3/\text{nap}$ értéknél. Ez ezért történik, mert minden kút közel van saját rezsimumához, annyira, hogy egy mutációs lépéssel már túllépné azt. Az Egyedi rezsimumasítás $20.800.000 \text{ Nm}^3/\text{nap}$ mennyiségben határozza meg a maximumot, és az ezekkel a működési paraméterekkel teljesíthető.
- 7-es számú nyíl:* a teljes üzemben elérhető minimális szállítási mennyiség 0,5%-os megközelítési pontját jelzi. Ezen a ponton a beállított mennyiség $6.000.000 \text{ Nm}^3/\text{nap}$, a görbe aktuális értéke pedig $6.000.800 \text{ Nm}^3/\text{nap}$.
- 8-as számú nyíl:* ellipszise a $45.000 \text{ Nm}^3/\text{h}^2$ / $30.000 \text{ Nm}^3/\text{h}^2$ sebességváltozásokat jelzi $14.000.000 \text{ Nm}^3/\text{nap}$ értéknél.

A trendről leolvasható, hogy bár kezdetben a kútcsoporthoz szállítási teljesítménye eltérő, a normál üzemi állapotban, amikor mindhárom kútcsoporthoz termel, az egyes csoportok teljesítménye együtt változik. A teljes kútállományra nézve ez az ideális állapot.

7 Összefoglalás

Evolúciós stratégiai elemek felhasználásával sikerült létrehoznom egy olyan egy olyan szimulációs környezetet, amely valós időben tudja végrehajtani a mennyiségváltoztatási feladatokat. A tesztek során a célfüggvényt 0,5%-os pontossággal közelítettem.

A futtatás során rezsim határérték túllépést nem tapasztaltam, erre a program felépítése nem ad lehetőséget. A felterhelési meredekségek az előre tervezett értéket tartották.

A kutak állapotáról a legszemléletesebb információt a fitness értékek átlagából kaptam. A felfuttatási szakaszban ez az érték elérte a 0,4-et. Egy mennyiség emelés, majd csökkentés után viszont a teljes üzem alatt állandósult 0,006...0,008 értéken. Ez a gázkutak, illetve szabályozó szelepek szempontjából azt jelenti, hogy a tároló réteg terhelése egyenletesen eloszló.

Annak érdekében, hogy az algoritmus használható legyen – a jelenlegi helyett –, a DCS rendszerbe lehessen implementálni, jelentős fejlesztéseket kell még végrehajtani rajta.

1. A 0,5%-os pontosság 20M Nm^3 földgáz esetén $\pm 100.000\text{ Nm}^3$ bizonytalanságot jelent naponta. A célfüggvény megközelítésekor csökkenteni kell a mutációs lépésközt, és a résztvevők számát. Ekkor sokkal pontosabb megközelítése lehetséges a célfüggvénynek.
2. Az algoritmust ki kell bővíteni a gázszáritókra és a kitároló kompresszorokra vonatkozó résszel.

Megjegyzés: Ez legalább ekkora, de inkább nagyobb feladat, mint az eddig megvalósított.

3. A jelen algoritmus nem általános, csak egy adott rezsim utasítás végrehajtását tudja elvégezni. Általánossá kell tenni. Meg kell határozni – és megvalósítani - azokat az interfészeket, amelyekre szükség lehet annak érdekében, hogy bármely rezsim utasítás esetén alkalmazható legyen.

Meg kell vizsgálni, hogy az alkalmazott eljárások és függvények, speciális VBA eljárások miként implementálhatóak a YOKOGAWA DCS SEBOL programozási nyelvén.

Sikeres implementáció és próbaüzemek után érdemes megvizsgálni, a

- Kardoskúti MODICON PLC,
- Zsanai SIEMENS PCS7 DCS,
- Pusztaedericsi EMERSON DELTA-V DCS

irányítástechnikai rendszereken történő alkalmazás lehetőségét.

Alkalmazott rövidítések jegyzéke

EFS	E.ON Földgáz Storage Zrt.
DCS	Distributed Control System - Elosztott irányítási rendszer
MTBF	Mean time between failures - Meghibásodások között átlagosan eltelt idő
FCS	Field Control System - Terepi vezérlő
SCS	Safety Control Station - Biztonsági állomásvezérlő
HIS	Human Interface Station - Kezelői állomás
VBA	Visual Basic for Application
EA	Evolúciós algoritmus
ES	Evolúciós stratégia
GA	Genetikus algoritmus
EP	Evolúciós programozás
PV	Process Value – Folyamatváltozó értéke
SV	Set Value – Beállítási érték
SEBOL	Sequence and Batch Oriented Language
VLSI	Very Large Scale Integration - Nagy mértékben integrált elemek

Irodalomjegyzék

1. Mankovics – Tornyi *Gáztárolói folyamatirányítás YOKOGAWA CENTUM CS3000 DCS-el* (konferenciai előadás) DCS-6 Folyamatirányító Rendszerek VI. Találkozó, Lillafüred 2000. október
2. Jászberényi Zoltán (az EFS igazgatósági tagja) *Földgázraktárak a föld alatt* E.ON Világ (A munkatársak lapja) 2009/2
3. Dr. Ajtonyi István – Dr. Gyuricza István *Programozható irányítóberendezések, hálózatok és rendszerek* Műszaki Könyvkiadó, Budapest 2002
4. A Nagyalföldi Kőolaj- és Földgáztermelő Vállalat története (1851-1989) Készült az NKFV Nyomda Üzemében 5000 példányban²
5. Álmos Attila, Győri Sándor, Horváth Gábor, Várkonyiné Kóczy Annamária *Genetikus algoritmusok* TYPOTEX Kiadó, Budapest 2003
6. Bevezetés a genetikus algoritmusokba
<http://prog.hu/cikkek/909/Bevezetes+a+genetikus+algoritmusokba.html>
7. Borgulya István *Evolúciós algoritmusok* DIALÓG CAMPUS Kiadó, Budapest-Pécs 2004
8. Aszalós László *Algoritmusok* mobiDIÁK könyvtár
9. Kovalcsik Géza *Az Excel programozása* ComputerBooks Kiadó, Budapest 2009
10. Farkas Csaba – Szabó Marcell *A programozás alapjai Visual Basicben* Jedlik Oktatási Stúdió, Budapest 2005

² Fellelhető az NKFV jogutód szervezeteinek nyilvános helyiségeiben és irodáiban

Ábrajegyzék

1.1 ábra Egy földalatti gáztároló sematikus rajza.....	3
1.2 ábra A földalatti gáztárolás sémája.....	5
1.3 ábra Egy gázkút Hajdúszoboszló körzetében	5
1.4 ábra DCS struktúra redundáns busszal	6
1.5 ábra A SystemView részlete.....	7
2.1 ábra A jelenlegi komplex algoritmus elvi ábrája.....	9
2.2 ábra A Hsz-154-es kút vezérlésének funkcióblokkos sémája	10
2.3 ábra A 172-es kút megjelenítése a HIS állomáson.....	10
2.4 ábra A megvalósítandó folyamat.....	13
4.1 ábra A kezdeti populáció részlete.....	16
4.2 ábra A normál eloszlás függvényei	19
5.1 ábra A program kezelői, adatbeviteli és megjelenítési felülete	25
5.2 ábra Információk az indulásról	26
5.3 ábra Az indulási mód választása.....	28
5.4 ábra Az Üzenet Panel	28
5.5 ábra A léptető nyilak tulajdonságai	31
5.6 ábra A rezsim utasítás részlete.....	39
5.7 ábra A <i>Data</i> munkalap részlete	39
5.8 ábra Az üzemelő kutak megjelölése	39
5.9 ábra A mennyiség emelési sebességek	40
5.10 ábra A változtatási sebességek és mennyiségi határok.....	40
6.1 ábra A kimeneti értékek az idő függvényében	41

Köszönetnyilvánítás

Ezúton szeretném megköszönni Dr Aszalós László segítségét és türelmét a dolgozat elkészítésében, valamint feleségem és családom támogató hozzáállását az iskola elvégzéséhez.