

TDK- dolgozat

Gelle Kitti Erzsébet

Optimalizálási feladatok hálózata

Gelle Kitti Erzsébet
Programtervező Informatikus MSc I. évfolyam

Témavezető:
Dr. Vinkó Tamás
egyetemi adjunktus

Szegedi Tudományegyetem Számítógépes Optimalizálás Tanszék

Tartalomjegyzék

ABSZTRAKT	4
BEVEZETÉS	6
1. KAPCSOLÓDÓ DEFINÍCIÓK.....	8
1.1. A globális optimalizálás	8
1.1.1. Optimalizáló eljárások	18
1.1.2. Optimalizálandó feladatok	13
1.2. Hálózattudomány.....	13
2. AZ ÖTLET.....	16
2.1. Legmeredekebb lejtő	17
2.2. Basin Hopping.....	19
2.3. Optimumkeresés a különböző szinteken.....	22
3. EREDMÉNYEK.....	25
3.1. A tesztfeladatok	25
3.1.1. Lennard-Jones klaszterezési feladat	25
3.1.2. Schwefel	26
3.3. Eredmények.....	28
4. ÖSSZEFOGLALÁS	39
Irodalomjegyzék.....	37

ABSZTRAKT

A globális optimalizálás területén végzett számos kutatás az optimalizálási feladatok és módszerek valamilyen szempontból releváns jellemzésével foglalkozik. Számos javaslat született már mind a feladatok nehézségének, mind az azokat megoldó eljárások hatékonyságának mérésére.

Ebben a megközelítésben célom a hálózattudomány matematikai eszközeit felhasználva jellemzést adni a feladatokra, azoknak nehézségére, a hálózatok egyedi tulajdonságait vizsgálva.

Ebben a kontextusban megadható olyan eljárás, mely felépíti az optimalizálási feladatra generálható gráfot, vagy annak egy jó közelítését. Az ötlet inspirálója Stillingen és Weber egy korai munkája volt, melyben atomklaszterek helyzeti energia „térképét” állították elő hálózatként. Az ötletük az volt, hogy ezek feloszthatóak olyan medencékre, melyek egy-egy lokális minimumot vesznek körül. Ezt a megközelítést később sikeresen alkalmazták a Lennard-Jones klaszterezési feladat több különböző dimenziós esetében előállított hálózatok elemzésére. Ebben az esetben egy olyan hálózatot írnak le, melyben a pontok a klaszterezési feladat lokális optimumai, s ezek között akkor szerepel él, ha úgynevezett átmeneti állapotokkal közvetlen össze vannak kötve. S ugyanezt a gondolatot alkalmazták már kombinatorikus optimalizálási feladatok esetén is. Én pedig ennek az ötletnek egy olyan változatát szeretném szemléltetni, mely kizárólag folyamatos optimalizálási feladatokkal foglalkozik.

Szakedolgozatomban már feldolgoztam az ötlet egy olyan formáját, melyben a gráfot olyan módon építjük fel, hogy a keresési teret egy megfelelően kis lépésközzel bejárva, minden pontból lokális keresőt indítunk többször, és ha egy pontból több lokális optimumot is találunk, azok között lesz él. Ez azonban magasabb dimenziós feladatokban nem vezetett eredményre, hiszen a keresési tér végigpásztázása nagy tár- és számításigényű. Mivel a véletlenszerű pontválasztás sem vezetett volna eredményre – hiszen egy magas dimenziós térben szintén nehéz feladat olyan pontot kiválasztani, melyből két különböző lokális minimum is elérhető, így a további gráfok előállításához egy Locatelli munkájában alkalmazott eljárást, a Basin-Hopping optimalizálót használtam fel, mely magas dimenziószám esetén is egy jó közelítést ad. Dolgozatom hipotézise, hogy egy optimalizálandó feladat akkor lesz nehezen megoldható, ha az általa generált gráfban a jó lokális optimumok a globális optimumtól – több globális optimum esetén maguk a globális

optimumok különböző közösségekben helyezkednek el. Első eredményeink egy globális optimalizálással foglalkozó konferencia anyagában is szerepelnek.[13]

BEVEZETÉS

*„A matematika izgalmas paradoxonja: hogy művelői
akármennyire nem vesznek tudomást a világról, annak
megértéséhez következetesen a lehető legjobb
eszközöket adják kezükbe.”*

– John Tierney

Két olyan tudományág ismereteit használom fel a dolgozatomhoz, melyek manapság igen közkedveltek, számos könyv és cikk jelenik meg évtizedek óta mindkét témában, s nem csak tudományos szempontból érdekesek, hanem mindennapi életünkben is nagy jelentőséggel bírnak. Ez a két tudományág az optimalizálás és a hálózattudomány, a matematika két látszólag távol álló területe, melyeknek összekapcsolása már csak azért is érdekes, mert eddig nagyon kevesen foglalkoztak velük együtt ebben a kontextusban.

Hálózati problémák optimalizálásáról sok forrásból tájékozódhatunk, viszont magukat az optimalizálási módszereket és feladatokat gráfok alapján vizsgálni, mondhatni új keletű dolog.

Az optimalizálási problémák életünk számos területén jelen vannak. Az alkalmazott kutatási módszertan több területén - matematika, fizika, közgazdaságtan – foglalkoznak azzal, hogyan lehetne az egyes problémákra a legjobb megoldást adni. És nem csak a tudományban, hanem a mindennapjainkban is jelen vannak az optimalizálásra váró problémák. Hogyan osszuk be az időnket, hogy minden feladatunkat el tudjuk végezni? Melyik boltba menjünk? Milyen úton menjünk, hogy eljussunk valahová? Ezek tulajdonképpen mind – mind optimalizálási problémák.

Az általános optimalizálási problémát akár a következőképpen is megfogalmazhatjuk: „Válasszuk ki a lehetséges megoldások közül a legjobbat, ami a feltételeinknek megfelel!” Persze ez így egy elég tág megfogalmazás, mégis ez az alapja az összes optimalizálási problémának.

Optimalizálással kapcsolatban tehát két fontos szempontot kell megjegyeznünk: a „legjobb” megoldást szeretnénk megtalálni, viszont erről csak akkor tudjuk megmondani, hogy mit is jelent, ha az eredményeink jóságát valahogyan mérni tudjuk. Másrészt a „lehetséges” legjobb megoldást keressük, tehát vannak bizonyos korlátok, amiket nem léphetünk át.

Érthető, hogy mivel a technológia fejlődése lehetővé teszi számunkra, ezeknek a problémáknak a megoldására minél gyorsabb, hatékonyabb és persze minél jobb

eredményeket elérő módszereket szeretnénk kifejleszteni. Illetve ezeknek a módszereknek a hatékonyságát szeretnénk valamilyen megközelítésből mérni.

Az optimalizálók hatékonyságának, valamint optimalizálási feladatok nehézségének mérésére már számos ötlet született, dolgozatomban én is egy mérési technikát használok, amelyben felhasználom a hálózattudomány eszközeit.

S hogy mit is értünk hálózattudomány alatt? A hálózat szó napjaink egy divatos kifejezése: „a hálózatok mindenütt ott vannak”. A hálózatok között kiemelt szerepet játszanak a természet, a társadalom és a technika komplex hálózatai. Gondoljunk csak néhány egyszerű példára, maguk a társas kapcsolatok, a sejtjeink, a táplálkozási lánc a föld élőlényei között mind – mind hálózatokba rendezhetők, s ahogy a hálózatok neves magyar kutatója, Barabási Albert-László megfogalmazta: *„Még a nyelv is, amit gondolataink közvetítésre használunk, önmagában véve nem más, mint szintaktikai kapcsolatokkal összekötött szavak hálózata.”*[7]

A hálózatközpontú megközelítés a katonai szervezeteknél is egyre inkább előtérbe került az 1990-es évek végétől kezdve. Ezeknek az volt az alapja, hogy mind az információs technika hatalmas fejlődésen ment keresztül, a hálózatok világméretűvé váltak (például Internet), mind pedig, hogy a társadalomtudományokban is eredményesen alkalmazták, valamint hogy a hálózatközpontú szervezetek kialakítása vált általánossá.

A különböző minősítéssel ellátott hálózatok – különböző szakterületekben vett jelentésükben - jelentős szerepet játszanak a tudományos kutatásokban, valamint a gyakorlatban is. Különösen igaz ez az információs technikákat (információcsere, információhoz történő hozzáférés) támogató hálózatokra.

A hálózat fogalma a hálózattudomány fogalomrendszerében fizikai, biológiai vagy társadalmi fogalmak absztrakciója, tehát *"a hálózat mindig a megfigyelhető valóság egy reprezentációja, vagy modellje és nem a valóság maga"*. [7]

Mivel sok természettudományos területen előszeretettel alkalmazzák a hálózatokat, mint vizsgálati módszert, így számomra is ez adta az ötlet koncepcióját: építsünk hálózatokat optimalizáló eljárások adta eredményekből különböző feladatokra, s vizsgáljuk meg, hogy a kapott gráfok tulajdonságai alapján megállapítható-e valamilyen különleges eset a feladatok nehézségének függvényében.

1. KAPCSOLÓDÓ DEFINÍCIÓK

Dolgozatom témájának alapja a globális optimalizálás témaköre, a kapott eredményeket pedig a hálózattudomány eszközeivel dolgozom fel és elemzem. Ezért fontosnak tartom mindkét témakör azon alapvető fogalmainak megemlítését, amelyek a dolgozat megértéséhez szükségesek. A továbbiakban szót ejtek magáról a globális optimalizálásról, mint fogalom, valamint az ehhez kapcsolódó optimalizálendő feladatok és az őket megoldó eljárások osztályozására is rövid áttekintést adok.

A második alfejezetben pedig a hálózattudomány azon fogalmai sorolom fel, melyeket kutatási eredményeim elemzéséhez felhasználtam.

1.1. Globális optimalizálás

A globális optimalizálás az optimalizálásnak egy olyan ága, amelynek célja egy feladat legjobb megoldásának megtalálása, amely megfelel a feladatban adott feltételeknek. A cél egy általában nem konvex, több szélsőértékkel rendelkező célfüggvény legjobb megoldásának (globális minimumhelyének vagy maximumhelyének) megtalálása. Az ilyen feladatok megoldásának kiszámítása általában NP-nehéz feladat.

A különböző tudományágak – fizika, kémia, gazdaság, stb. – számos területén megjelenő modern problémák akár globális optimalizálási feladatokként is felfoghatóak és megadhatóak azok megoldásai számítógépes optimalizálási technikák segítségével.

Általánosan a következőképpen írhatunk fel egy globális optimalizálási feladatot:

$$\min_{x \in A} / \max_{x \in A} f(x)$$

$$\text{f.h. } g_i(x) \leq 0 \quad i \in I \quad (\text{egyenlőtlenség feltételek})$$

$$g_j(x) = 0 \quad j \in J \quad (\text{egyenlőség feltételek})$$

ahol:

- $f(x): A \rightarrow \mathbb{R}$ a célfüggvény: az a függvény, amelynek a minimumát vagy maximumát keressük,
- $A = \{x \in \mathbb{R}^n \mid a \leq x \leq b, a, b \in \mathbb{R}^n\}$ intervallum korlát,
- $\forall i \in I, j \in J \quad g_i, g_j : A \rightarrow \mathbb{R}$ korlátozó feltételek, g_i egyenlőtlenség, g_j egyenlőség feltétel,

- $X = \{x \in A | g_i(x) \leq 0 \ \forall i \in I, g_j(x) = 0 \ \forall j \in J\}$ a megengedett megoldások halmaza. [4]

A továbbiakban csak minimalizálási problémákkal foglalkozunk, hiszen bármely maximalizálási feladatot visszavezethetünk egy minimalizálásra a következő konverzió segítségével:

$$\max f(x) = -\min(-f(x))$$

Az adott feltételektől függően az alábbi 3 optimalizálási feladattípussal találkozhatunk:

- ha $X = \mathbb{R}^n$ akkor feltétel nélküli az optimalizálási feladatunk,
 - ha $I \cup J = \emptyset$, de $X = A \subset \mathbb{R}^n$ akkor intervallum korlátozott feladatunk van,
 - és ha $I \cup J \neq \emptyset$ akkor feltételes vagy korlátozott optimalizálási feladatról beszélünk.
- [3]

Tehát mint tudjuk, a globális optimalizálás annyit tesz, mint megkeresni a lehetséges legjobb megoldást egy adott problémára. Ahhoz, hogy meg tudjuk határozni, mit is jelent egy feladat optimális megoldása további fogalmak bevezetésére van szükség.

1.1. Definíció (Lokális minimum): f_i^* egy lokális minimum érték, vagyis $\exists x_i^* \in X$ és $\varepsilon > 0$, hogy $f_i^* = f(x_i^*) \leq f(x)$ minden $\|x - x_i^*\| < \varepsilon$, $x \in X$ esetén. Legyenek nagyság szerint rendezettek, azaz $f_1^* < f_2^* < \dots f_i^* < f_{i+1}^* < \dots$.

1.2. Definíció (Lokális minimumok halmaza): $X_i^* = \{x_i^* \in X | f(x_i^*) = f_i^*\}$ az f_i^* lokális minimumhoz tartozó lokális minimumhelyek halmaza.

1.3. Definíció (Globális minimum): $f^* = f_1^*$ a globális minimum, $f^* \leq f(x) \ \forall x \in X$.

1.4. Definíció (Szinthalmaz): $L_f(y) = \{x \in X | f(x) \leq y\}$ f egy szinthalmaza. [4]

1.5. Definíció (Nyeregpont): olyan pont, amelyben a célfüggvény gradiense zérus ($\nabla f(x) = 0$), de minden környezetében van nála kisebb és nagyobb célfüggvény értéket felvevő pont is. Stacionárius pontként is ismert.

1.6. Definíció Egy optimalizálási feladat megoldásának a globális (esetleg lokális) optimum függvényértékét, helyét, az összes globális (esetleg lokális) optimum halmazát, vagy ezek valamilyen kombinációját tekintjük.

Az, hogy egy globális optimalizálási feladatot megoldhatónak tekintünk-e nagyban függ annak tulajdonságaitól, és attól, hogy milyen megoldást keresünk.

Egy problémát akkor tekintünk megoldottnak, ha elérjük a következő szinthalmazt

$$X_\epsilon \equiv \{x \in X | f(x) \leq f^* + \epsilon\}$$

valamely $\epsilon > 0$ -ra vagy pontot találunk az alábbi tartományból:

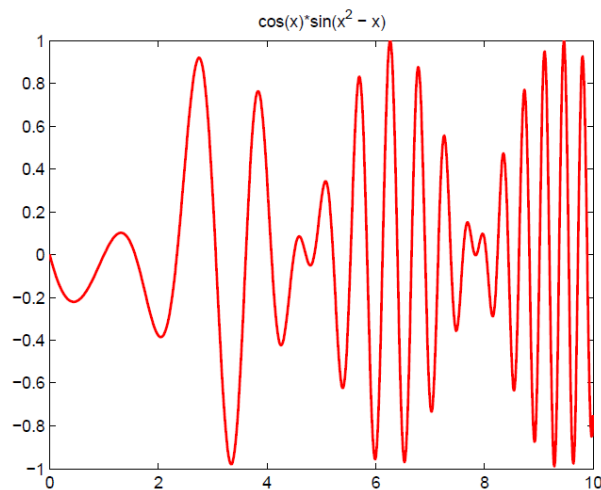
$$N_\epsilon = \{x \in X | \|x - x^*\| < \epsilon \text{ valamennyi } x^* \in X^* - \epsilon a\}$$

Ha a globális optimumot szeretnénk közelíteni, akkor egy folytonos függvény esetén nem garantált, hogy biztosan véges lépésben megtaláljuk az $f^* + \epsilon$ értéket.

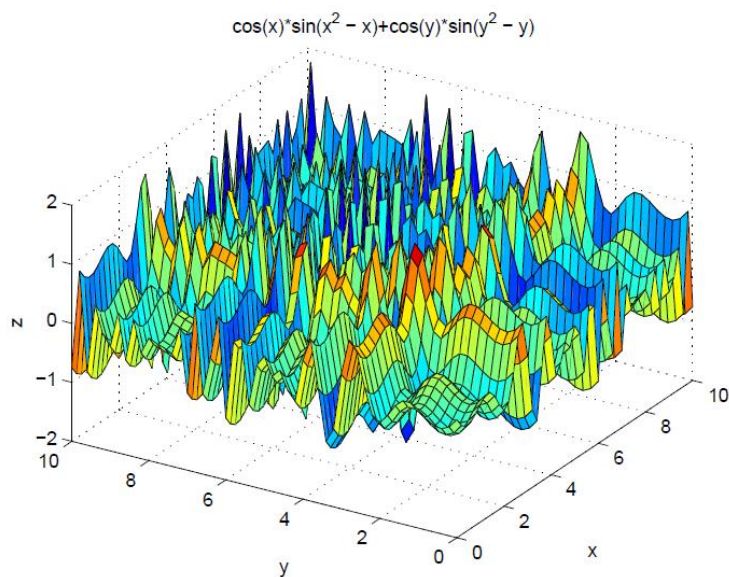
Amennyiben feladatunkhoz adottak a lehetséges megoldás intervallumok alsó és felső korlátai, akkor intervallumkorlátos feladatról beszélünk. Az ilyen feladatok esetén arról van szó, hogy a lehetséges megoldások teljes halmazában szeretnénk megtalálni célfüggvényünk abszolút minimumértékét. Ez egy egyébként is bonyolult feladat, amit tovább nehezít, hogy a probléma dimenziószámának növekedésével exponenciálisan növekszik a keresési tér. [11] (1.1. és 1.2. Ábra)

1.7. Definíció (Vonzási tartomány): Legyen f_i^* lokális minimum érték. Ekkor az f_i^* lokális minimum vonzási tartománya $attr(f_i^*) \subseteq X$ azon $x \in X$ pontok legnagyobb halmaza, amelyekből a legmeredekebb lejtő módszere végtelenül kicsi lépésközzel f_i^* -hoz konvergál $\forall x \in attr(f_i^*)$ esetén.

1.8. Definíció Egy f_i^* lokális minimum elérhető, ha $\mu(attr(f_i^*)) > 0$, ahol μ egy $\mathbb{R}^n$ -beli mérték.[4]



1.1. Ábra Egydimenziós, intervallumkorlátos feladat [11]



1.2. Ábra Kétdimenziós, intervallumkorlátos feladat [11]

1.1.1. Optimalizáló eljárások

Az optimalizáló eljárásokat szokás egyszerűen determinisztikus és sztochasztikus csoportokba sorolni, ám ez egy nem szerencsés csoportosítás, hiszen semmit nem tudunk meg általuk az eljárással kapott eredményről. Egy pontosabb kategorizálást ad a *Handbook of Global Optimization* [12], mely szerint az optimalizáló eljárásokat két fő csoportba sorolhatjuk: egzakt és heurisztikus eljárások.

A determinisztikus eljárásokat az egzakt eljárások közé soroljuk, mivel elméletben garantálják legalább egy, vagy az összes globális optimum elérését. A legtöbb folytonos

globális optimalizálási problémáról ismert, hogy megoldásuk NP-nehéz. Ezen eljárásoknak a számítási igénye pedig a probléma dimenziójával exponenciális arányban növekszik. Ezért a magasabb dimenziójú feladatokban célszerű sztochasztikus eljárásokat választani. Ezek az eljárások X^* egy eleméhez konvergálnak közel egy valószínűséggel (azaz majdnem biztosan). Az ilyen metódusok esetében fontos, hogy megadjunk egy valamilyen elfogadható numerikus célt, ebben az esetben a sztochasztikus eljárásunk biztosítja, hogy talál egy keresési pontot, mely egy „elfogadhatóan jó minőségű” megoldása a problémának. Azt is fontos megemlíteni, hogy sokszor nehéz különbséget tenni a determinisztikus és sztochasztikus módszerek között, illetve léteznek olyan módszerek is, amelyek mindkét típus elemeit tartalmazzák. Az egzakt eljárásoknak azonban szükségük van valamilyen plusz információra a feladatról, teljesen általános esetre nem létezik algoritmus ezekből az osztályokból.

Heurisztikus eljárásról beszélünk abban az esetben, ha az eljárás képes közelítő megoldását adni egy problémának, de nem bizonyítható sem az eljárás helyessége, sem valamely korlát a közelítés hibájára. Ezek az eljárások ugyan nem biztosítják a pontos eredményt az egzakt típusúakkal ellenben, ugyanakkor sok nehéz globális optimalizálási problémára jó közelítő megoldásokat adnak. A heurisztikus eljárásokat tehát szintén a magas dimenziójú, modell struktúra nélküli problémák esetén érdemes alkalmazni, illetve kedvezőek a nagy számítási erőforrások hiánya esetén is, valamint általában gyorsabbak egzakt típusúaknál.

Ezekben belül a módszereket a velük elérhető megoldás minősége alapján további két-két alcsoportba oszthatjuk be:

- **Heurisztikus eljárások:**

- **Nem teljes eljárás:** Lehetséges, hogy nem a globális optimumhoz konvergál.
- **Aszimptotikusan teljes eljárás:** Olyan eljárás, mely biztosan eléri a globális optimumot, ha végtelen ideig futhat, de soha nem tudhatjuk, hogy az aktuális megoldásunk milyen messze van a globális optimumtól.

- **Egzakt eljárások:**

- **Teljes eljárás:** Olyan eljárás, mely egzakt aritmetikát feltételezve biztosan eljut a globális optimumhoz, ha végtelen ideig futhat, valamint véges sok lépést követően tudja, hogy egy közelítő megoldást talált a globális minimumra (előre megadott tűrésen belül).
- **Megbízható eljárás:** Olyan teljes eljárás, amely kerekítési hibák mellett is teljes.

1.1.2. Optimalizálандó feladatok

Attól függően, hogy mennyire ismerjük a megoldandó optimalizálási problémát, ha annak egy adott típusú megoldását keressük, könnyebben választhatunk megoldási módszert. Ezért ezeket a problémákat több szempont szerint szokás osztályozni.

Osztályozási szempontok:

- **Dimenziószám** (egydimenziós, többdimenziós)
- **Függvénytulajdonságok** (konkáv, konvex, kvadratikus, Lipschitzes, stb.)
- **Korláttulajdonságok** (feltétel nélküli, intervallum korlátos, egyenlőség/egyenlőtlenség feltételes, stb.)
- **Egyéb információk**: gradiens, analitikus függvény megadás, Hesse-mátrix számíthatósága, f értékkészlete X -ben, lokális és globális optimumok száma.

Ezen tulajdonságok ismeretében tehát könnyebben választhatunk optimalizáló eljárást. Egy egydimenziós feladatra például sokkal több egyszerű eljárást találhatunk, mint többdimenziós társaira. Az optimum helyek számának ismerete a megállási feltétel megadásában van segítségünkre.

Persze olyan eset is előfordulhat, hogy semmilyen információnk nincs az optimalizálандó függvényről, csak a kiértékelő eljárás. Ekkor fekete doboz problémáról beszélünk.

Dolgozatomban felhasználok a Locatelli által adott osztályozást, mely az optimalizálандó feladatokat azon szempontból osztályozza, hogy mennyire egyszerűen képes egy több véletlen pontból induló eljárás elérni a globális optimumot, vagy esetleg benn ragad egy-egy lokális optimumban. Ezekből a csoportokból adok példákat dolgozatom utolsó fejezetében.[9]

1.2. Hálózattudomány

Manapság a tudomány számos területén használják a hálózatokat, mint elemzőeszközöket. Egy kapcsolatrendszeréről, melyből hálózat építhető számos érdekes tulajdonság deríthető ki, ha az épített hálózatot megvizsgáljuk. Épp ezért én is ezt az elemzési módszert választottam, hogy az optimalizálási feladatok nehézségét vizsgáljam.

Magának a hálózatnak a fogalma a hálózattudományi értelmezésben matematikai, ezen belül gráfelméleti alapokra épül. Ennek értelmében a hálózat csúcspontok halmaza, melyeket

páronként élekkel kapcsolunk össze. Ezzel lényegében megegyezik a gráf fogalma, ám ezzel csak az egyszerűbb hálózatok jellemezhetők, hiszen ezeknek számos más tulajdonsága is lehet - például lehetséges, hogy a hálózat kapcsolatai nem egyenműek. [7]

Ha másképp nem definiáljuk a hálózatunknak n csúcsa és m éle van, és a csúcspárok között csak egyetlen él fut. Ekkor egyszerű hálózatokról beszélünk. Munkám során az optimalizáló eljárások eredményeit ilyen egyszerű hálózatokkal – irányítatlan gráfokkal - reprezentálom.

A hálózatokat szokás szomszédsági mátrixszal megjeleníteni. Egy szomszédsági mátrix elemei a következő szerint kapnak értéket:

- megcímkézzük a hálózat csúcsait:

$$A_{i,j} = \begin{cases} 1 & \text{ha van (i,j) él} \\ 0 & \text{különben} \end{cases}$$

- Irányítatlan egyszerű gráfok esetében ez egy főátlóban zérus értéket felvevő, szimmetrikus mátrix.
- Szokás a gráf éleit súlyozni is. Ebben az esetben a gráf minden éléhez hozzárendelünk egy értéket, ami az él költsége, súlya vagy hossza alkalmazástól függően.

A továbbiakban a gráfok elemzéséhez felhasznált néhány hálózattudományi fogalmat definiálok, valamint azokhoz a lokális optimumok hálózatának kontextusában vett magyarázatot adok.

1.9. Definíció (Méret): A hálózat méretét a csomópontok száma határozza meg.

1.10. Definíció (Fokszám): Egy csúcs fokszáma a ráilleszkedő csúcsok száma:

$$k_i = \sum_{j=1}^n A_{i,j}$$

Ezzel együtt érdemes megvizsgálnunk a csomópontokban a fokszámok eloszlását. Egy hálózat $P(k)$ fokszám eloszlásfüggvénye azt mutatja meg, hogy mekkora a k fokszámú csúcsok aránya.

1.11. Definíció (Úthossz): Az átlagos úthosszt a hálózatban található legrövidebb utak hosszának átlagaként határozzuk meg.

1.12. Definíció (Átmérő): A hálózat átmérője az összes legrövidebb út közül a leghosszabb.

1.13. Definíció (Betweenness centrality): Egy központisági érték, melyet a csúcsokra úgy határozunk meg, hogy azt mérjük, hogy az adott csúcs hányszor szerepel a csomópontok közötti utakon. Általános definíciója

$$x_i = \sum_{s,t} \frac{n_{st}^i}{g_{st}},$$

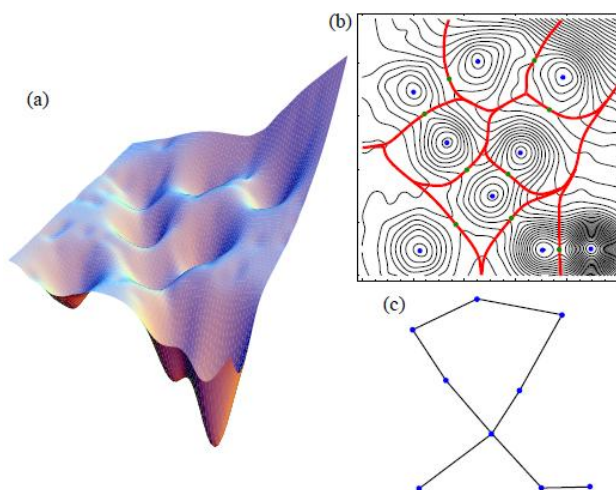
ahol:

- $n_{st}^i = 1$, ha i rajta van az s,t legrövidebb úton
- g_{st} = összes legrövidebb s,t utak száma

2. AZ ÖTLET

Mint a bevezetésben említettem, kevesen foglalkoznak a hálózattudomány és az optimalizálás ilyen értelemben vett együttes vizsgálatával. Dolgozatom témájának alapjául az egyik ilyen vizsgálati eset szolgált, Jonathan P. K. Doye *The network topology of a potential energy landscape: A static scale-free network* [6] című cikke, melyben a Lennard-Jones klaszterezési feladat különböző dimenziós eseteire hozott létre hálózatokat, és vizsgálta ezeket. A feladat röviden a következő: vegyük egy rendszer atomjainak elhelyezkedését, s ezeknek az atomoknak a koordinátáit állítsuk olyan pozícióra, amelyben a rendszer energiája a legalacsonyabb. Ebben a cikkben Doye egy olyan hálózatot ír le, melyben a pontok a klaszterezési feladat lokális optimumai, s ezek között akkor szerepel él, ha úgynevezett átmeneti állapotokkal közvetlen össze vannak kötve. Az megérthetőség kedvéért ezt a 2.1-es ábra szemlélteti.

Ebben a cikkben Doye a kapott hálózatokat két központi fogalom mentén vizsgálja. Megállapított, hogy ezek a hálózatok rendelkeznek az úgynevezett 'kis világ' tulajdonsággal, valamint skálafüggetlenek.



2.1. Ábra (a) Egy két dimenziós feladat energiaszíne. (b) A felszín kontúros ábrája, melyen a minimum pontokat körülvevő medencék jól elkülöníthetők. A piros vonalak a medencék határait, a kék és zöld pontok pedig a minimumok és átmeneti pontokat jelölik. (c) Az ez alapján kapott hálózat. [6]

Doye cikke Stillinger és Weber egy korábbi munkájának ötletét használja fel, ugyanis ők fogalmazták meg az atomklaszterek helyzeti energiatérképét hálózat formájában. S azt írják

le, hogy egy ilyen hálózat felosztható olyan medencékre, melyek egy-egy lokális optimumot vesznek körül. Doye mellett szintén felhasználták ezt az ötletet olyan kutatásokban is, ahol kombinatorikus optimalizálási feladatokat vizsgáltak. Én ennek az ötletnek a folyamatos optimalizálási feladatokra vett változatát dolgozom fel.

Szakedolgozatomban már feldolgoztam az ötlet egy olyan formáját, melynek működése a következőképpen írható le:

Egy feladatra egy választott, elfogadhatóan finom lépésközzel bejárjuk a teljes keresési teret, minden pontból többször elindítva egy lokális optimalizálási eljárást. Minden iterációban megnézzük, hogy milyen megoldást kaptunk, és eltároljuk.

Az így kapott eredményekből egy hálózat építhető. Ebben a hálózatban két pont között akkor van él, ha az optimalizálási eljárásunkat ugyanabból a kiindulási pontból indítva találtuk meg őket.

A keresési tér módszeres bejárása azonban magas dimenziószámú feladatoknál nem volt megfelelő a gráf megépítésére, ugyanis a keresési tér végigpásztázása nagy idő- és tárigényű feladat. A keresési teret adott lépésközzel eltárolni nagy tárhelykapacitást igényel, majd minden pontból egyesével többször futtatva az adott lokális optimalizálót, mely esetemben egy legmeredekebb lejtő eljárás volt rendkívül költséges, így nem volt hatékony. Az, hogy esetleg a keresési térből véletlen válasszunk kezdőpontokat, szintén nem vezetett volna megoldásra, hiszen az, hogy egy stacionárius pontot válasszunk ki, melyből a legmeredekebb lejtő eljárás több lokális optimumba is képes eljutni szintén nehéz feladat.

2.1. Legmeredekebb lejtő

A kezdetekben használt eljárás egy egyszerű legmeredekebb lejtő algoritmus volt. Erről az algoritmusról fontos tudni, hogy csak differenciálható függvényeken képes megoldást biztosítani, hiszen felhasználja számításaihoz a gradiens értékét. Egy hatásos lokális keresőről van szó, hiszen ha létezik a gradiens, akkor az mindig egy alacsonyabb szint felé fog mutatni, az algoritmusunk pedig a negatív gradiens irányában „ereszkedik” tovább.

Ha $x = \bar{x}$ egy kezdőpont, a d pedig az úgynevezett keresőiránya az f függvénynek $\bar{x}$ -nél, ha létezik $\delta > 0$ érték, hogy $f(\bar{x} + \lambda d) < f(\bar{x}) \forall \lambda \in \{0, \delta\}$ esetén. A λ értéket lépéshossznak nevezzük.

Ahhoz, hogy ezt az értéket minimalizáljuk, a λ szerint egyváltozós iránymenti deriváltat 0-nál kisebbre kell, hogy kiértékeljük. Tehát ha

$$\nabla f(\bar{x})^T d = \lim_{\lambda \rightarrow 0^+} \frac{f(\bar{x} + \lambda d) - f(\bar{x})}{\lambda} < 0,$$

akkor d a keresőirány.

A legmeredekebb lejtő algoritmus tehát minden pontban d irányba lép (úgy, hogy $\|d\| = 1$), így minimalizálva a fenti értéket.

Az eljárás a következő 4 lépésben felírható: [5]

- I. Adjuk meg x^0 -t, valamint legyen $k \leftarrow 0$
- II. $d_k = -\nabla f(x_k)$. Ha $d_k = 0$ megáll.
- III. Legyen λ_k a k . iteráció lépéshossza, melynek értékét a következő alapján számítjuk:
 $\lambda_k = \arg \min_{\lambda > 0} f(\bar{x} + \lambda d)$
- IV. Legyen $x_{k+1} \leftarrow x_k + \lambda_k d_k$, $k \leftarrow k + 1$ és ugorjunk a II. lépésre.

Megállási feltételként szokás még a gyakorlatban azt is megadni, hogy ha két egymást követő iterációban a megoldás alig javul, akkor az eljárás befejeződik. Pontosabban legyen $\varepsilon > 0$ egy tetszőlegesen kicsi szám, az általunk kívánt pontosság. Ha az

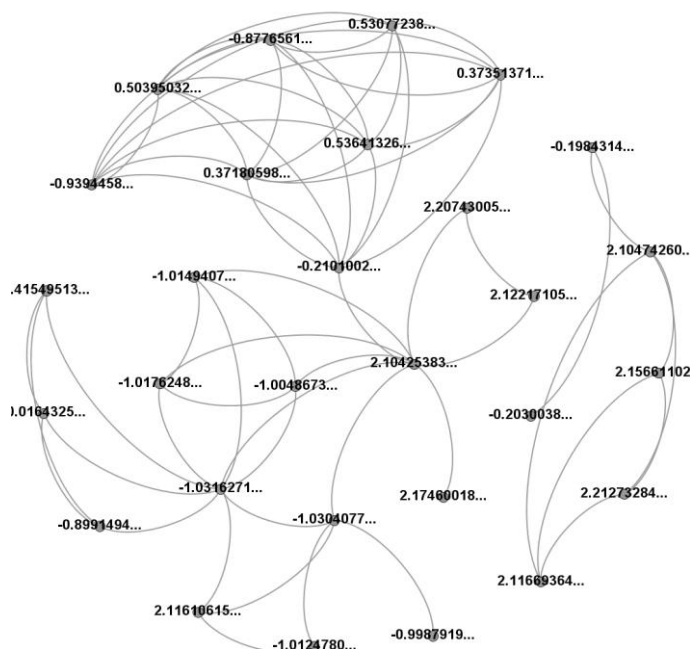
$$e = \frac{\|x_{k+1} - x_k\|}{1 + \|x_k\|} < \varepsilon$$

egyenlőtlenség teljesül, akkor az algoritmus megáll, és az utoljára kapott megoldást tekintjük a probléma közelítő megoldásának. Az ε értékét a kívánt pontosság mértékétől függően 10^{-10} és 10^{-6} között érdemes megválasztani.

Belátható, hogy ebben az esetben ez egy teljesen determinisztikus futású algoritmus, mely semmilyen randomizált értéket nem tartalmaz. Ebben a formában tehát nem volt alkalmas gráfok építésére. A megoldást a következők jelentették. Vizsgáljuk meg azt az esetet, ha egy x_s kezdőpont gradiensenek értéke zérus (ezt egy $\|\nabla f(x_s)\| < \varepsilon$ feltétel kiértékelésével ellenőrizzük), akkor ez azt jelenti, hogy például stacionárius ponton álluk. Amennyiben ténylegesen egy ilyen pontból indítjuk a keresést, az x_s pont koordinátaiban pici változtatást teszünk. Ettől a minimális elmozdulástól a stacionárius pontokon az algoritmus más-más irányban fog elindulni a „lejtőn” felépítve a gráf éleit.

Ez az eljárás sajnos nem volt hatékony egyrészt amiatt, hogy sok olyan pontban megállt, mely nem volt optimum (egy ezzel az eljárással előállított gráf látható a 2.2. ábrán) másrészt pedig amiatt, hogy ezt felhasználva a keresési téren megfelelően kellett felderíteni a

stacionárius pontokat, és mivel ezeknek a pontoknak a kiválasztására nem ismerünk hatékony eljárást, így feltétlen szükség volt egy a teljes keresési tér módszeres bejárásához. Magasabb dimenzióban pedig ez az eljárás nem volt megvalósítható, így szükségem volt egy olyan eljárásra, mely képes a gráfnak egy megfelelően jó közelítését adni magas dimenziós feladatok esetén is úgy, hogy nincs szükségük arra, hogy előre ismerjék a teljes keresési teret. Tehát szükségem volt egy globális optimalizáló eljárásra, ehhez pedig egy olyan változatot használtam fel, melyről már említették könyvben, hogy egyes verziói épp az optimalizálandó feladat által adott gráf pontjain kellene végig „ugráljon”.



2.2. Ábra Az SHCB feladatra a legmeredekebb lejtő algoritmus által gyártott gráf

2.2. Basin hopping

Az 1.1.3. fejezetben már összefoglaltuk, hogy az optimalizálási eljárások milyen csoportokba sorolhatók. Azonban ezekben a módszerekben mégis van néhány közös vonás működésüket illetően.

Egy globális optimalizáló eljárásoknak alapvetően 3 fontos fázisát különböztetjük meg:

- Globális fázis: feltérképezzük a teljes keresési régiót.
- Lokális fázis: az ígéretes részeket – melyek várhatóan egy optimumhoz közel lesznek – behatóbban vizsgálja.

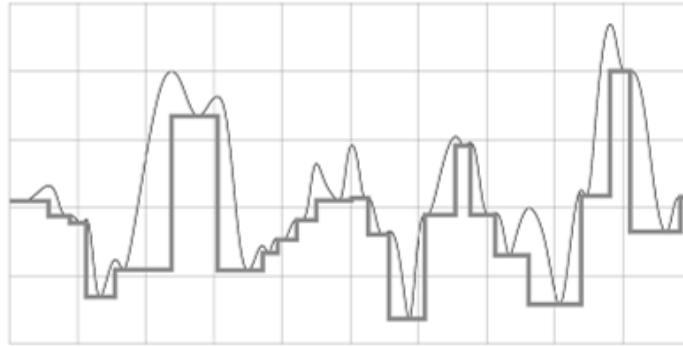
- Adaptáció: az ígéretes részeket sűrűbben mintavételezi, ez egy köztes lépés a globális, és a lokális fázis között.

Ezekon a fázisokon alapszik az a módszer, melyet a kutatásom későbbi fázisaiban használtunk, melyet a *Global Optimization: Theory, Algorithms, and Applications*[10] című könyvben úgy definiáltak, hogy ezen eljárás egy optimalizálási feladat gráfjának pontjain – azaz a lokális optimumokon – „ugrál”. Ez az eljárás a Basin Hopping. Az eljárás a következőképpen működik:

```
while (GlobalStoppingRules == false){
    x = GloballyGenerate();
    x =  $\Theta$ (x);
    while (LocalStoppingRules == false){
        z =  $U(N(x))$ ;
        y =  $\Theta(z)$ ;
        if(IsAcceptable(x,y)){
            x = y;
        }
    }
}
```

Ez az algoritmus kicsit hasonló egy multistart verzióhoz, azonban azzal ellentétben itt csak a globális fázisokban generálunk kezdőpontot a teljes keresési térben, a lokális fázisban azonban kicsit másképp működik: x -nek egy bizonyos környezetéből ($N(x)$) generálunk következő kezdőpontot. Másrészt viszont úgy is tekinthetünk rá, mintha egy tiszta multistart típusú eljárás lenne, külön-külön indítva a lokális fázisokat, melyben egy-egy lokális optimumot találunk meg.

Az eljárás nevét arról kapta, hogy amennyiben a feladat függvény grafikusán reprezentálható, az út, amit a Basin Hopping eljárás követ úgy írható le, mintha az egyik lokális minimumból ugrana át egy másik közeli lokális optimumra, tehát „átugrik” egyik medencéből a másikba. Egy másik nézőpontból úgy is tekinthetünk rá, mint egy standard globális optimalizálási eljárásra, melyet nem konkrétan egy tesztfüggvényre alkalmazunk, hanem a lokális keresés és a tesztfüggvény egy egyesítésére. Ebben az esetben az optimalizálandó függvény leírható úgy is, mint a Basin Hopping által szakaszonként talált optimumértékeinek konstans függvénye. Ezt írja le a 2.3. ábra.



2.3. ábra. Egy függvény és a lokális keresések eredményeként kapott transzformációja [10]

Az eljárásnak különböző implementációi léteznek. Az alapvető elemek azonban mind közősek, és az implementációban én is ezeket alkalmaztam:

- **Megállási feltételek:** általában egyszerűen egy maximális iterációs szám. Az iterációk pedig úgy alakulnak, hogy lokális esetben, ha az elfogadási kritérium teljesül, akkor lenullázzuk a számlálónkat, egyébként pedig növeljük azt, globális esetben csak növeljük a számlálót.
- **Elfogadási kritérium:** alapvetően két változatot különböztetünk meg.
 - *Monotonic Basin Hopping:* ebben a változatban az $\text{IsAcceptable}(x, y)$ metódus az x és y megoldásokra akkor lesz igaz, ha $f(y) < f(x)$, tehát ha az y helyen a függvényérték alacsonyabb, mint x -en. Tehát a monoton verzió csak abban az esetben vesz fel új megoldást, ha az optimalizálás szempontjából jobb.
 - *Generic Basin Hopping:* ebben az esetben $\text{IsAcceptable}(x, y)$ igazat ad vissza, ha teljesül a következő feltétel - mely a szimulált hűtésnél is használatos: $U[0,1] \leq \exp(-(f(y) - f(x))/T)$, ahol T egy nemnegatív paraméter, az úgynevezett „hőmérséklet”, melyet általában monotonon csökkentünk az iterációk alatt. Ebben az esetben az iterációk folyamán egyre kisebb valószínűséggel elléphetünk „rossz” irányba is egy jobb megoldás megtalálása érdekében.
- **Lokális kereső (Θ):** általában egy egyszerű lokális keresési metódus, mely valamilyen gradiens alapú keresési stratégiát használ. Ez persze függhet a feladat típusától is.

Eleinte egy Java-ban implementált egyszerű legmeredekebb lejtő eljárással próbálkoztam, ám ez sok esetben nem volt elég pontos, és beragadt olyan

pontokban, melyek nem voltak optimumhelyei a függvénynek. Így a későbbiekben a feladatok és a Basin Hopping eljárás AMPL-ben lett implementálva. Ennek előnye, többek között az is, hogy számos ingyenesen használható hatékony lokális optimalizáló eljárást alkalmazhatunk lokális keresőként.

2.3. Optimumkeresés a különböző szinteken

Locatelli volt az, aki a Basin Hopping eljárást használta fel optimalizálási feladatok vizsgálatára.[9] Cikkében a globális optimalizálási feladatokon történő optimumkereséseket különböző szintekbe sorolta. A szinteket az úgynevezett ékszer probléma alapján sorolta be. Az ékszer probléma röviden a következő: van egy nagy dobozunk, nevezzük X -nek. Ebben a dobozban van T db kisebb doboz, legyenek $X_i, i = 1, \dots, T$. Minden X_i dobozban van további t_i db kisebb doboz, $X_{i,j}, j = 1, \dots, t_i$, végül ezek is tartalmaznak $\tau_{i,j}$ még kisebb dobozt, $X_{i,j,k}, k = 1, \dots, \tau_{i,j}$. Minden ilyen $X_{i,j,k}$ doboz tartalmaz egy ékszert, melynek van egy értéke. Az ékszer probléma megtalálni az egyes ékszerek közül a legmagasabb értékűt. Ezt a problémát definiálhatjuk $X_{i,j}$ és X_i dobozok felett is, ahol $X_{i,j}$ értéke a benne található legnagyobb értékű ékszer, míg X_i értéke a benne található $X_{i,j}$ -k értékei közül a legmagasabb. Továbbá feltételezzük, hogy ezek a dobozok nem véletlenszerűen vannak elrendezve. Ezt megfogalmazhatjuk úgy is, hogy adott $X_{i,j}$ -hez definiálunk egy $G_{i,j}$ gráfot, melynek pontjai a benne található $X_{i,j,k}, k = 1, \dots, \tau_{i,j}$ dobozok az egyes értékeikkel, és két pont között van él, ha az egyes kapcsolódó dobozok egymáshoz érnek $X_{i,j}$ -ben. A gráf struktúrája olyan, hogy minden $G_{i,j}$ -ben ($i = 1, \dots, T; j = 1, \dots, t_i$) van egyetlen maximális értékű ékszer. A G_i és G gráfokat hasonló módon definiálhatjuk.

Ekkor a következő szinteket különböztetjük meg:

0. szint: $X_{i,j,k}$ dobozokat tartalmazza
1. szint: $X_{i,j}$ dobozokat és $G_{i,j}$ gráfokat tartalmazza
2. szint: X_i dobozokat és G_i gráfokat tartalmazza
3. szint: X dobozt és G gráfot tartalmazza

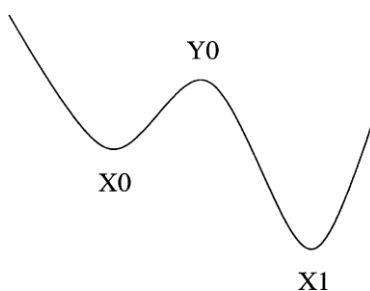
Minden szinten lokális keresést alkalmazva egy csomópontot a közvetlen felette lévő szinten értékelünk ki. Így az egyes szintekkel növekszik a csomópontok kiértékelésének költsége is.

Ebben a formában látható a többszintű szerkezet problémája: a 3-tól az 1. szintig ugyanaz a szerkezet, csupán a pontok kiértékelésének költsége változik.

Ezt a problémát egyszerűen vissza tudjuk vezetni egy globális optimalizálási problémára, melyben az $X_{i,j,k}$ dobozok megfelelnek az egyes pontoknak a keresési térben az adott helyen felvehető értékkel, míg az $X_{i,j}$ -k felfoghatóak lokális optimumokként, $G_{i,j}$ -k pedig mint a lokális optimumok vonzáskörzetei. Egy $X_{i,j}$ doboz kiértékelése megfelel egy benne végrehajtott lokális keresésnek. Egy-egy X_i doboz lokális minimumok egy halmaza. X pedig az összes lokális minimum a keresési térben.

Az 1. szinten a lokális keresés, mint említettem megfelel egy egyszerű lokális keresésnek, így ebben az esetben nincs nehéz dolgunk, egy gradiens alapú keresési eljárás könnyen megtalálja az optimumot. A 2. és 3. szinten viszont ennél bonyolultabb a helyzet.

A 2. szinten definiálnunk kell lokális minimumok szomszédságát. Többféleképpen is megadhatjuk, egy egyszerű eset, ha a lokális minimumok távolságára alapozunk: feltételezzük, hogy minden lokális optimum fix távolságra van a kijelölttől. Ez azonban bár egyszerű, de nem használható a gyakorlatban, ugyanis nem garantálható, hogy egy függvény olyan lesz, hogy bármely lokális optimumtól adott fix távolságra lesz az összes többi. Vethetünk egy olyan szomszédságot is, melyet az alapján definiálunk, hogy azokat a pontokat vesszük szomszédosnak, melyeket az adott lokális optimumtól egy stacionárius pont választ el (2.3. ábra). Locatelli egy olyan szomszédságfüggvényt alkalmaz, mely viszonylag könnyen számolható. Legyen a lokális minimum x_0 és egy doboz $C[x_0, r_1] = [x_0 - r_1 e, x_0 + r_1 e]$, ahol $r_1 > 0$ az élhossz fele, és x_0 a középpont (e pedig az a vektor, melynek minden eleme 1). Így x_0 szomszédja lesz minden olyan lokális minimum, melyek vonzáskörzetének metszete a C halmazzal nem üres.



2.3. ábra. Két lokális minimumot (x_0 és x_1) egy stacionárius pont y_0 választ el [9]

Ez alapján definiálhatjuk a keresést a 2. szinten:

1. lépés: legyen y_0 egy lokális minimum. Legyen $k = 0$.
2. lépés: válasszunk véletlenszerűen egy pontot w_k az $X \cap C[y_k, r_1]$ -ben, Indítsunk egy lokális keresést w_k -ből és legyen z_k az így kapott lokális minimum.
3. lépés: ha $f(z_k) < f(y_k)$, legyen $y_{k+1} = z_k$, egyébként $y_{k+1} = y_k$.
4. lépés: Ha egy megállási feltétel teljesül, álljunk meg és térjünk vissza y_{k+1} -el, egyébként legyen $k = k + 1$ és menjünk a 2. lépésre.

Lokális keresés a 3. szinten:

1. lépés: Legyen ξ_0 egy lokális minimum a 2. szinten. Legyen $k = 0$.
2. lépés: Legyen $d_k, \|d_k\| = 1$ egy véletlen irány és r_2 a pozitív fix lépésköz. Legyen: $\alpha_k = \xi_k + r_2 d_k$ és legyen φ_k egy lokális minimum a 2. szinten melyet α_k -ból indított lokális kereséssel találtunk a 2. szinten.
3. lépés: Ha $f(\varphi_k) < f(\xi_k)$, legyen $\xi_{k+1} = \varphi_k$, egyébként $\xi_{k+1} = \xi_k$.
4. lépés: Ha egy megállási feltétel teljesül, álljunk meg és térjünk vissza ξ_{k+1} -el, egyébként legyen $k = k + 1$ és menjünk a 2. lépésre.

3. EREDMÉNYEK

3.1. A tesztfeladatok

Évtizedek óta foglalkoznak egyre jobb, hatékonyabb, gyorsabb optimalizáló módszerek fejlesztésével, így elengedhetetlenné vált olyan teszt problémákat összegyűjteni, amelyeken mérhetjük ezeknek az eljárásoknak a hatékonyságát. Dolgozatomban én is ezeken a standard tesztfüggvényeken futtattam az optimalizáló eljárásokat.

A továbbiakban néhány – valamely tulajdonsága miatt kiemelkedő – teszt példát mutatok be.

3.1.1. Lennard-Jones klaszterezési feladat

A Lennard-Jones klaszterek az optimalizálás, a kémia és a fizika területén is jelentős modellek. Az általános Lennard-Jones potenciál alakja a következőképpen írható le:

$$v_{t,e}(r) = 4\varepsilon \left(\left(\frac{t}{r} \right)^{12} - \left(\frac{t}{r} \right)^6 \right)$$

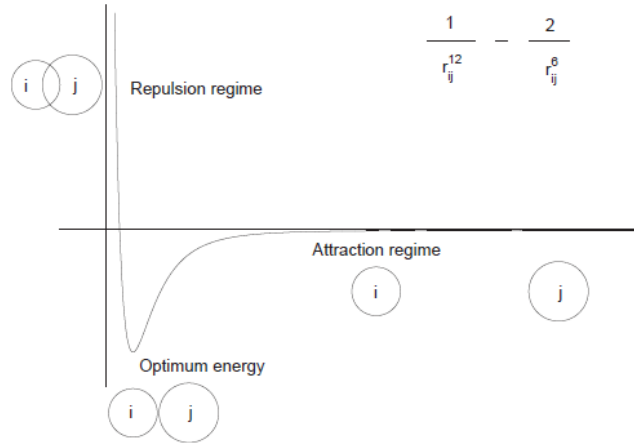
A globális optimalizálás esetében a függvénynek két általánosítása használandó: az egyik az $\varepsilon = t = 1$ és $s = 2^{1/6}$ eset

$$v_{1,1}(r) = \left(\frac{4}{r^{12}} - \frac{4}{r^6} \right)$$

másik pedig az ún. skálázott LJ potenciál, melyet ($\varepsilon = 1, t = 2^{-1/6}$ és $s = 1$)

$$v_{2^{-1/6},1}(r) = \left(\frac{1}{r^{12}} - \frac{2}{r^6} \right).$$

A skálázott Lennard-Jones potenciál függvény egy semleges atompárnak egy egyszerű unimodális függvénye. (Az unimodális f függvény az x^* minimumhelytől balra szigorúan monoton csökkenő, tőle jobbra pedig szigorúan monoton növekvő.) Ezt ábrázolja a 3.1. ábra.



3.1. ábra Egy atompár skálázott LJ potenciálja [3]

Általában azonban nem egy atompár lép egymással kapcsolatba, hanem több, így azt az esetet kell vizsgálnunk, amikor minden atom minden másikkal interakcióba lép. Egy adott N atomos LJ klaszterezési probléma tehát, hogy megtaláljuk az atomok olyan relatív pozícióját a háromdimenziós Euklideszi térben, hogy azok az energia potenciál minimumát adják.

A Lennard-Jones potenciál függvény a következő alakban definiálható: [3]

$$LJ_{t,\varepsilon}(x) = \sum_{1 \leq i < j \leq n} v_{\sigma,\varepsilon}(\|x_i - x_j\|)$$

Ezek a klaszterezési feladatok skálázhatóak attól függően, hogy hány atomos klasztert vizsgálunk. Ezekről a feladatokról tudjuk, hogy közülük sok, 2-30 atomszám között viszonylag könnyűek – ezek közül a 13 atomos változatot fogjuk elemezni. Míg például a 38 atomos változat egy kifejezetten nehéz optimalizálási feladatnak tekinthető.

3.1.2. Schwefel

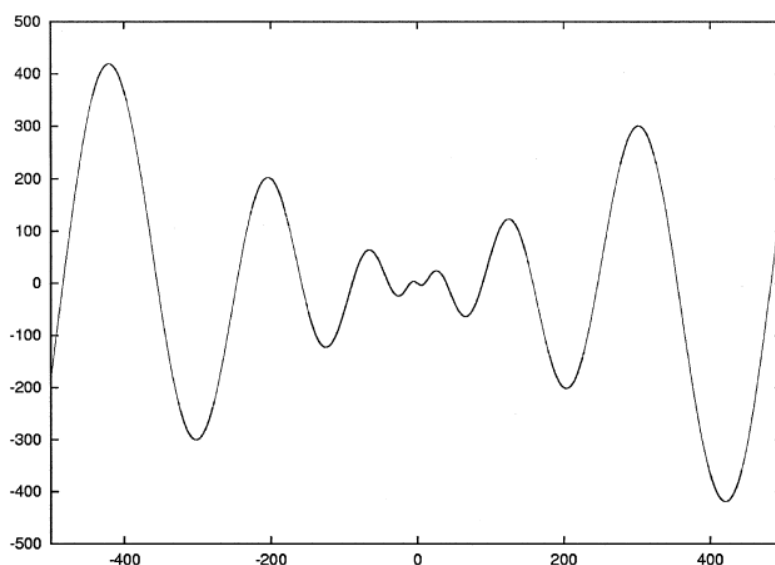
Szintén egy N dimenzióban skálázható függvényről van szó, melynek általános képlete a következőképpen adható meg:

$$Schwefel(x) = \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|})$$

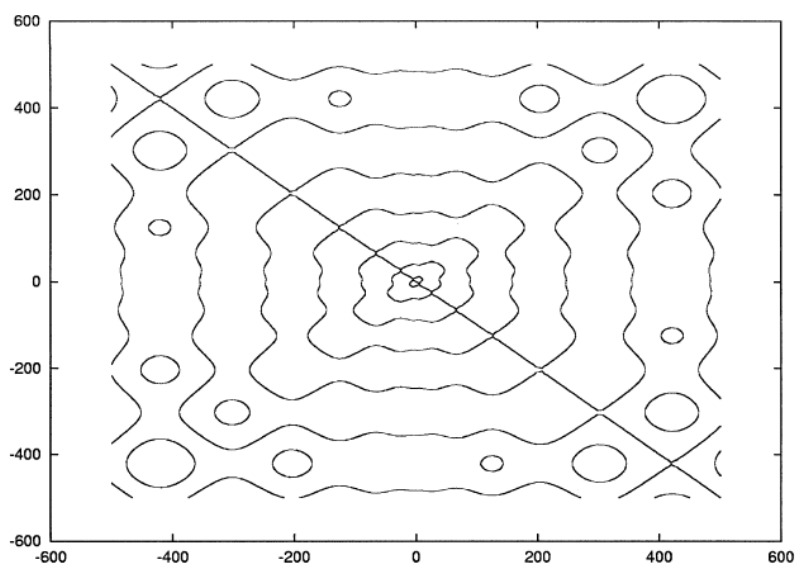
ahol $x_i \in [-500, 500]$.

A feladatnak számos lokális minimuma van, illetve egyetlen globális optimuma, mely az $x^* = (1, \dots, 1)$ pontban az $f(x^*) = 0$ értéket veszi fel.

Ez a függvény azért különleges, mert egy optimalizálás szempontjából elég nehéz feladatról van szó, hiszen a lokális optimumok messze találhatók egymástól, illetve a jó lokális optimumok (globális értékhez közeliek) szintén nagyon messze találhatók a globális optimumtól. Ezt ábrázolja a 3.2. és 3.3. ábra.



3.2. ábra Az 1 dimenziós Schwefel függvény [9]



3.3. ábra Az 2 dimenziós Schwefel függvény [9]

3.2. Futtatási eredmények

Jonathan P. K. Doye *The network topology of a potential energy landscape: A static scale-free network* [6] című cikkében a Lennard-Jones klaszterezési feladatra kapott hálózatokat vizsgálta, s engem is ez a publikáció inspirált arra, hogy további kutatásokba kezdjek az optimalizáló módszerek hatékonyságáról úgy, hogy az általuk generált eredményekből hálózatot építsek. Mint említettem Doye két fontos tulajdonságot vélte felfedezni ezekben a hálózatokban. Az egyik, hogy megvan bennük a 'kis világ' tulajdonság. Ahhoz, hogy meg tudjuk magyarázni, mit is nevez a hálózattudomány kis világnak, példának Stanley Milgram amerikai pszichológus 1960-ban végzett kísérletét szeretném bemutatni. A kísérlet célja az volt, hogy felrajzolhassák a személyközi kapcsolatok egyes embereket közösséggé szervező hálóját. Ezért Nebraska és Kansas államban élő embereket választott ki véletlenszerűen, s küldött levelet, azzal a kéréssel, hogy továbbítsák a levelet egy Bostonban élő tőzsdeügynök barátjának, akinek a címét viszont elmulasztotta mellékelni. Csupán arra kérte őket, küldjék olyan személyes ismerősüknek, akiről úgy vélik, társadalmi kapcsolatai révén közelebb lehetnek a tőzsdeügynökhöz. A levelek többsége végül tényleg megérkezett az említett bostoni baráthoz, de a meglepő az volt, hogy a levelek milyen gyorsan érkeztek meg – nem több száz, csupán átlagosan hatszori postázással annak ellenére, hogy mind Kansas, mind Nebraska állam társadalmilag távol van Bostontól. Milgram felfedezése közkeletűvé vált, s *hatlépéses távolság* néven be is vonult a folklórba. [8] A kis világ elmélet azonban 1998-ig ellenállt a kutatásoknak, amikor abban az évben a Cornell University két matematikusa, Duncan Watts és Steve Strogatz felfedezte a kis világ tulajdonsággal rendelkező gráfokat. Olyan rendezett, körkörös gráfokból indultak ki, amelyeknek minden pontját csak néhány legközelebbi szomszédjával kötötték össze. Majd véletlenszerűen kiválasztottak egy pontpárt és élt rajzoltak közéjük. A két professzor több ilyen gráfot is megvizsgált számítógépes módszerekkel, s a felfedezésük az volt, hogy a véletlenszerűen hozzáadott élek gyökeres hatást gyakoroltak az összekötési távolságra. Tegyük fel, hogy a kezdeti kör 1000 pontot tartalmaz, s minden pontot a 10 legközelebbi szomszédjával kötünk össze. Ekkor a hálózat 5000 kapcsolatot számlál, s a véletlen utak nélkül az összekötési távolság nagyjából 50 körüli. Azonban a mindössze néhány véletlen utat a hálózatba téve ez a szám 7-re zuhan. [8] Az ilyen kevés véletlen útvonalat tartalmazó, egyébként szabályos gráfokat nevezzük tehát kis világnak, melyekben az út két egyed között igen rövid. Doye másik felfedezése a skálafüggetlenség volt, egy hálózatról akkor mondjuk, hogy skálafüggetlen, ha foksám eloszlásuk hatványfüggvény logaritmikus skálán ábrázolva.

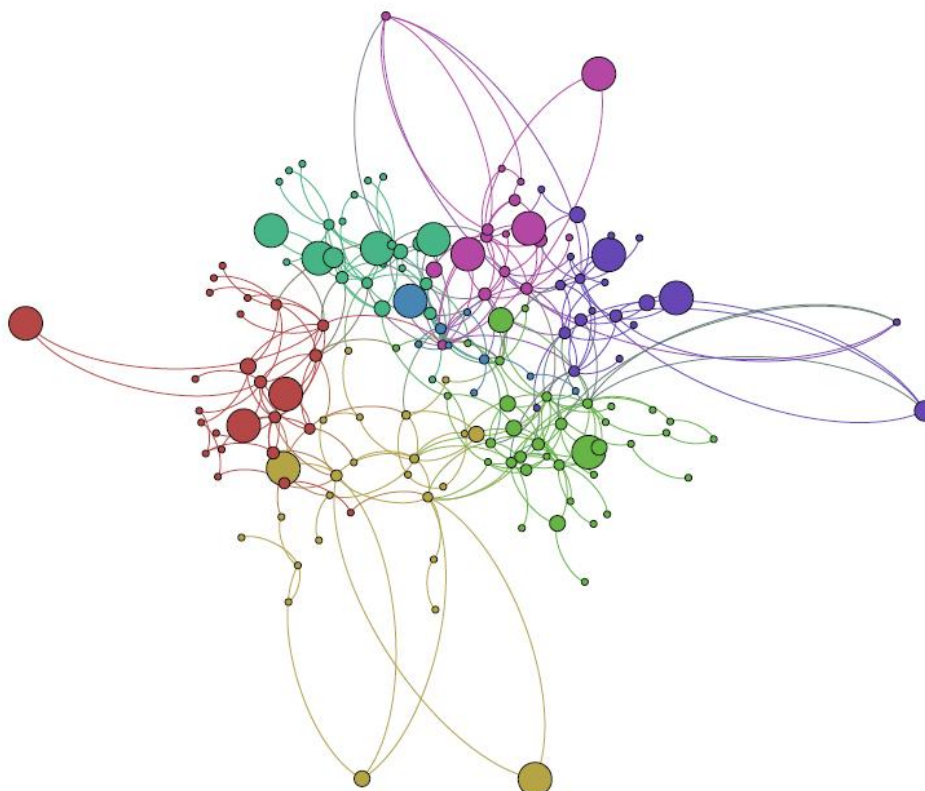
Emellett volt még néhány feltételezésem az egyes gráf tulajdonságokkal kapcsolatban, melyeket a heurisztikákat figyelembe véve igaznak is találtam:

- A gráf mérete várhatóan az optimalizáló által talált lokális minimumok száma lesz.
- Az egyes csúcsok fokszáma előre láthatóan a szomszédos stacionárius pontok számát fogják megadni.
- Az átlagos úthosszból azt fogjuk tudni kiolvasni, hogy várhatóan hány ugrás fog kelleni ahhoz, hogy egy medencéből a másikba jussunk, elérve a globális optimumot.
- A hálózat átmérője megfelel annak a legrosszabb esetnek a nem lokális ugrások számát tekintve, amelyekkel elérjük a globális optimumot.
- Végül pedig, hogy a globális optimum fog minden hálózatban a legmagasabb Betweenness centrality értékkel bírni.

Dolgozatomban a fent bemutatott két feladattal foglalkoztam részletesen. A Lennard-Jones klaszterezési feladat gráfjai elérhetőek online, melyeket Doye kutatásai során állított elő. Így ezek közül használtam fel a 13 atomos változatot vizsgálataimhoz, mint „könnyű” feladatra példát, ugyanis ez a feladat olyan típusú, hogy az optimumok elrendeződése olyan, mintha egy csőben lépkednénk lefelé, melynek alján a globális optimum áll. Ez alapján egy Monoton Basin Hopping eljárás bármely pontból indítva könnyen eljut a globális optimumba.

A másik általam vizsgált feladat a Schwefel volt, melynek egy 2D-s esetére készült gráf a saját AMPL alapú implementációval. Ez a gráf egy megfelelő közelítése az „eredeti” gráfnak, s a 3.4. ábrán látható.

Dolgozatom célja igazolni, hogy egy optimalizálási feladat abban az esetben lesz nehezen megoldható, ha az egyes lokális optimumai különböző közösségekben helyezkednek el a feladat hálózatában. A közösségek alatt, pedig olyan hálózat részeket értünk, ahol a tranzitivitás értéke magas adott pontokon belül, tehát az egy közösségben lévő csúcsok valamilyen szempontból összetartoznak. A közösségek szerepe a hálózattudomány számos területén jelentős. Sok esetben vizsgálunk olyan gráfokat, ahol a közösségek központi szerepet töltenek be, a legegyszerűbb példát véve gondoljunk csak a szociális hálózatokra. A közösségeknek és a kis világ gráfoknak azonban egészen sok közülük van egymáshoz, nem véletlen, hogy optimalizálási szempontból Doye fontosnak tartotta, hogy ha egy optimalizálási feladat gráfja bír a kis világ tulajdonsággal, az megkönnyíti annak megoldását.



3.4. ábra A 2D Schwefel függvény gráfja a közösségek jelölésével

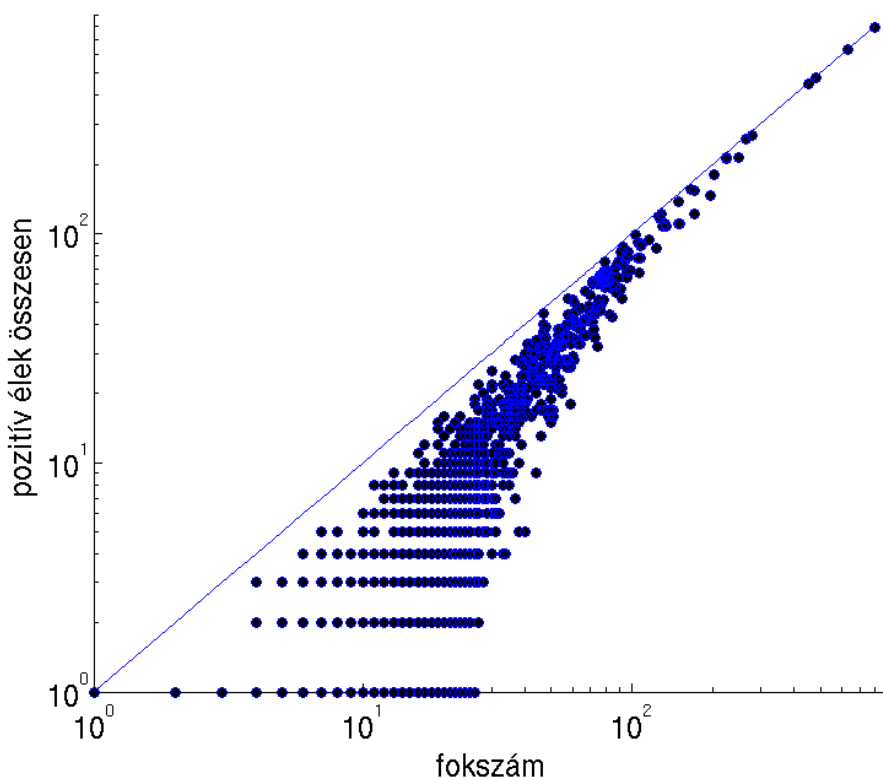
Vegyük sorra a számunkra fontos elemeket ebben a kontextusban. Tudjuk, hogy amennyiben a hálózatban vannak közösségek, akkor a hálózatunk „fürtös” formát fog felvenni, ami azt jelenti, hogy vannak egymással jobban összekapcsolt csúcsok – *a barátom barátja az én barátom is*. Úgy is tekinthetünk az ilyen közösségekre, hogy itt erős összeköttetések vannak a csúcsok között. Ezek az erős összeköttetések azt is jelentik, hogy ezeken keresztül könnyen eljuthatunk a közösségen belül bármely csúcsból bármely másik csúcsba. Amennyiben van egy ilyen gráfunk, de nem rendelkezik a kisvilág tulajdonsággal, az azt fogja jelenteni, hogy a hálózatban vannak hosszú legrövidebb útjaink, tehát költséges lehet a hálózat egyik elemétől egy másikig eljutni. Optimalizálási szempontból nézve tehát ez azt jelenti, hogy ha egy hálózatban különböző közösségekben vannak a lokális optimumok, akkor egy Basin Hopping típusú eljárás egyikből a másikba csak nehezen, vagy egyáltalán nem is tudna eljutni, tehát benn ragadna egy-egy közösségben. Azonban, ha feltételezzük a gráfról, hogy kis világ, mely azt fogja jelenteni, hogy az egyes közösségek pontjai között léteznek gyenge összeköttetések, melyeken viszont könnyen átugorhatunk egyikből a másikba. Márpedig ha ez így van, az azt fogja jelenteni, hogy az optimalizálónknak könnyebb dolga lesz, ha egyik optimumból egy másik olyanba ugrana át, mely másik közösségben van. Tehát ez azt jelenti, hogy míg az erős összeköttetések esetleges megszüntetése nem lenne jelentős hatással a legrövidebb utakra,

úgy egy gyenge összeköttetés már jelentősen befolyásolná azokat. Doye megmutatta a Lennard-Jones klaszterekről, hogy azok gráfjaira minden esetben teljesül a kis világ tulajdonság, [6] én pedig szakdolgozatom írása során szintén olyan gráfokat állítottam elő különböző optimalizálási feladatokból – köztük a Schwefelről, melyek szintén kis világok.

Vegyük Doye másik vizsgált elméletét, azaz hogy miért lesz optimalizálási szempontból jó az, ha egy feladat gráfjának fokszám eloszlása hatványfüggvényt követ. A matematikusok a görbe „farkának” nevezik azt a részt, ahol a csomópontok száma a nullához közelít. A görbe farka arról tanúskodik, hogy kevés csomópontnak van nagy számú kapcsolata. A tudósok évszázadok óta ismerik a normális eloszláshoz társítható ún. „haranggörbét”. A haranggörbe és a hatványgörbe farka ugyan nagyon hasonló, mégsem azonos. A hatványgörbéknek úgynevezett „kövér farkuk” van. Vagyis a haranggörbéhez képest a hatványgörbe sokkal lassabban csökken nullára. Ebből pedig következik az, hogy az ilyen hálózatokban néhány csomópontnak olyan sok kapcsolata van, hogy a hálózat összes kapcsolatának 80-90 százaléka a csomópontok egészen kis töredékére esik. A hatványösszefüggés tehát matematikai megjelenési formája egy speciális architektúrának, amelyben dominálnak a különösen sokszorosán összekapcsolt centrumok.[8] Egy ilyen hálózatban tehát nagy az esélyünk arra, hogy könnyen találunk olyan csúcsot, mely nagy számú összeköttetéssel rendelkezik. Ezt az optimalizálásban kihasználhatjuk, ha a hálózatunkban ezek a nagy fokszámú csomópontok a lokális optimumok, hiszen így könnyen találhatjuk meg őket bármilyen optimalizáló eljárással.

Másrészt viszont azt is meg kell magyaráznunk, hogy miért is fontosak a közösségek az optimalizálási feladatok hálózataiban. Feltételezésem az, hogy azért, mert ha egy feladat hálózatában annak jó – tehát nagyon alacsony értékű - lokális optimumai különböző közösségben helyezkednek el a globális optimumtól, vagy több globális optimum esetén maguk a globális optimumok szintén különböző közösségekben vannak, akkor az azt fogja jelenteni, hogy egy ilyen optimalizálási feladat nehéznek bizonyul. Ennek vizsgálatához a fentebb bemutatott két tesztfeladatnak egy új gráfját építettem fel az alapján, hogy egy monoton Basin Hopping eljárás hogyan lenne képes bejárni a gráfot. A gráfot egy olyan Basin Hopping eljárással építettük, melyben a lokális ciklusokban az elfogadási kritérium mindig igaz, tehát minden esetben átlépünk a következő talált pontba annak ellenőrzése nélkül, hogy jobb függvényértéket kapunk-e ott. Ez azért volt jó, mert ebben az esetben az eljárásnak a gráf minél hatékonyabb felderítése volt a cél, nem pedig a globális optimum megtalálása. Azonban arra is kíváncsi voltam, hogy a monoton verzió milyen utakat tudna ezen a gráfon bejárni, így súlyokat rendeltem a gráf éleihez. Abban az esetben, ha olyan pontba ugrunk, mely a

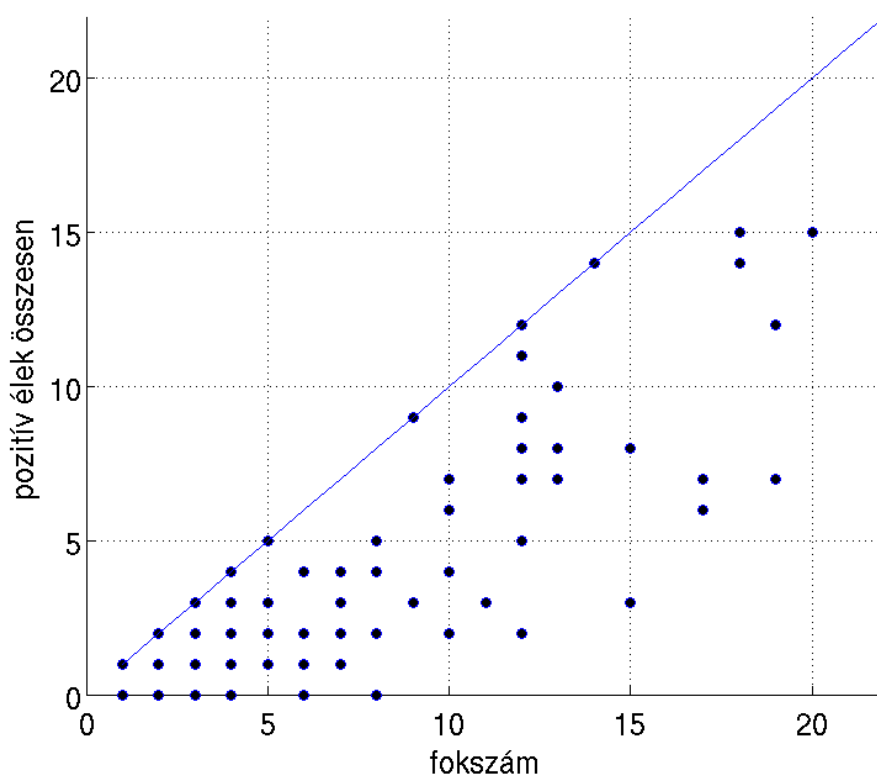
monoton eljárás elfogadási kritériumának is megfelelne, az él 1-es értékű súlyt kap, ellenkező esetben pedig -1 értékűt. A gráf legyártása után, amennyiben nem találta meg az „oda-vissza” utat az építéshez használt verzió, éleit megkettőzttem. Tehát ha egy él csak 1 súlyal szerepelt a vissza irányt is szerepeltettem -1 súlyal, mivel a nem monoton eljárás találhatna utat mindkét irányban, illetve, ha egy él csak -1 súlyal szerepelt az azt is jelenti, hogy a monoton eljárás az ellenkező irányba képes lenne haladni, így ezt is fel kell vennünk az 1 súlyú élek közé. Ez alapján szerettem volna vizsgálni, hogy a feladat mennyire lehet nehéz egy monoton Basin Hopping eljárásnak. Az ötlet az volt, hogy hasonlítsuk össze a kapott gráfban a csúcsok összfokszámát és ugyanazon a csúcsoknak csak a pozitív súlyú éleket figyelembe véve a fokszámait. Az eredményeket a 3.5 és 3.6 ábra szemlélteti.



3.5. ábra A 13 atomos Lennard-Jones klaszterezési feladatra kapott eredmények

Az ábrákon két fontos dolgot érdemes megfigyelnünk. Az egyik, hogy jól látható, hogy egyik esetben sincsenek pontok az egyenlőséget mutató vonal felett, ez azt jelenti, hogy nem lehetséges az a verzió, hogy a monoton eljárást használva többször megtalálhatnánk a pontot, mint a felderítő eljárás esetén. Ellenben vannak olyan pontjaink, melyek pont ezen az egyenesen találhatóak, azaz olyan pontok, melyekben egy monoton Basin Hopping eljárás be fog ragadni – mivel nem lesz belőle kifelé mutató él. A 13 atomos Lennard-Jones

klaszterezési feladat esetén azt is tudjuk, hogy a legmagasabb fokszerű csűcs     a glob  lis optimum, emellett a f  ggv  nynek m  s, kiugr  an j     rt  k   lok  lis optimuma nincsen. A k  t dimenzi  s Schwefel feladat esetén tudjuk, hogy a glob  lis optimum mellett sz  mos m  sik, hozz   f  ggv  ny  rt  kben k  zeli lok  lis optimuma van, ellenben a LJ feladattal.



3.6.   bra A 2D-s Schwefel feladatra kapott eredm  nyek

Egy jelent  s k  l  nbs  g a k  t feladat k  z  tt, hogy a Schwefel eset  ben sokkal t  bb olyan pontunk van, melynek a pozit  v   ls  lyokat figyelembe vett foksz  m   sszege maxim  lis, r  ad  sul ezek sokszor olyan cs  csok, melyeknek alacsony a foksz  ma, m  g a LJ feladatn  l ez csak az utols   pontra igaz. M  sr  szt azt is   szrevehetj  k, hogy m  g a LJ feladat eset  ben eredetileg nagy foks  r  m   cs  csok k  z  tt nincsenek olyanok, melyekbe monoton esetben is sokszor megtal  lhatunk, addig a Schwefel feladat esetén sz  mos ilyen pontunk van. Ezek alapj  n pedig levonhatunk n  h  ny k  vetkeztet  st a fentiekb  l. Az alapj  n, hogy a Schwefel esetén t  bb olyan pontunk van, mely magas foks  r  m     s pozit  v s  ly     leit figyelembe v  ve is viszonylag magas   rt  ket kapunk felt  telezhetj  k, hogy ezekben a pontokban egy monoton Basin Hopping benn ragadhatna, f  leg, ha az adott pont a hozz   hasonl  akkal egy  tt egy k  z  ss  gben helyezkedik el. M  sr  szt a 13 atomos LJ feladatr  l ismert, hogy tulajdonk  ppen

az optimalizáló szempontjából olyan, mintha egy szűkülő „csőben” kellene leugrálnia, melynek végén a globális optimum van. Ezt támasztja alá az ábránk is, mi szerint a globális optimumhoz közeledve már nincs olyan másik pont, amely szintén olyan magas foksámú, mint a globális optimum, így abba könnyen bele fogunk találni, hiszen sokan kapcsolódnak hozzá, és könnyen el is lehet oda jutni bárhonnán. Ez alapján is feltételezhetjük tehát, hogy míg a Lennard-Jones klaszterezési feladat 13 atomos esete egy viszonylag könnyű feladatnak számít, addig a 2D-s Schwefel függvény sokkal nehezebb.

Emellett az így előállított gráfokon vizsgáltam, hogy hogyan alakulnak a hálózat átmérőjének és átlagos úthosszának értékei, ha csak azokat az éleket vesszük figyelembe, melyek a monoton Basin Hopping esetén jöhetnek szóba, tehát pozitív súlyúak. Ezeket jelöltem meg, mint „Irányított” esetet. Ez alapján a következő értékeket kaptam:

	13 atomos LJ		Schwefel	
	Átmérő	Átlagos úthossz	Átmérő	Átlagos úthossz
Irányítatlan	5	2.39	7	3.56
Irányított	15	3.54	7	2.98

Előzetes feltételezésem az volt, hogy ezekről az adatokról is következtethetünk a feladat nehézségére például abból, ha az átmérő megnövekedik irányított esetben, akkor az nehezebb feladatot fog feltételezni. Azonban ezen adatok alapján ez a feltételezésem nem igazolódott be. A Lennard-Jones klaszterezési feladat esetén pedig jelentősen megnövekedik a hálózat átmérője, ami azt jelenti, hogy egy monoton Basin Hopping eljárással van olyan pontpár, melyek között ilyen hosszú ugrássorozattal tudunk eljutni. Habár a kis világ tulajdonságot nem „rontottuk el”, hiszen a hálózatunkban továbbra is 1508 a csúcsok száma, ennek ellenére még is háromszorosára növekedett a maximális úthossz. Erre magyarázat lehet, hogy a hálózat 1508 erősen összefüggő komponensre esett szét, tehát a pontok tulajdonképpen nem alkotnak közösségeket ebben az esetben. Így tehát meg kellett állapítanom, hogy önmagukban a fenti felsorolásból bármelyik tény – kis világ gráf, skálafüggetlenség, monoton BH hatékonysága, átlagos úthossz és átmérő változása – nem ad kielégítő választ olyan

tekintetben, hogy az adott optimalizálási feladat milyen nehéz, azonban ezeket a tényeket együtt vizsgálva már közelebb juthatunk egy jó sejtéshez.

Végül nézzük meg, miért is lehetnek fontosak egy optimalizálási feladat szempontjából, hogy milyen nehéz és hogyan helyezkednek el benne a közösségek. Ha létezne egy olyan optimalizáló, mely „látja”, fel tudja térképezni ezeket a közösségeket, akkor lenne egy hatékony eljárásunk ezeknek a nehéz feladatoknak a megoldására. Locatelli cikkében úgy érvel, hogy a Monotonic Basin Hopping megfelelően paraméterezve egy ilyen hatékony eljárás lehet, mely a lokális ciklusaiban egy közösségen belül lépked, míg a globális fázisokban képes átlépni másik közösségbe.[10]

4. ÖSSZEFOGLALÁS

Dolgozatomban kifejtettem, hogy a hálózattudomány és a globális optimalizálás életünk szerves részei. Az olvasó megismerhette azokat a fogalmakat, és definíciókat, melyek szükségesek a feladatom megértéséhez, ismertettem a globális optimalizálási feladatok és eljárások osztályozási lehetőségeit. Rövid áttekintést adtam arról, milyen előzményei vannak munkámnak, illetve milyen nehézségekbe ütköztem munkám során. Bemutattam néhány standard tesztproblémát, melyeket vizsgáltam, illetve azon az optimalizáló módszerek működének alapjait, melyeket felhasználtam munkám során. Megfogalmaztam, melyek voltak azok a feltételezések, amiket vizsgálni terveztem dolgozatomban, s példákat adtam arra, hogy a futtatási eredmények alapján ezekből mik teljesültek.

Irodalomjegyzék

- [1] C. Jansson and O. Knüppel. A global minimization method: The multi-dimensional case. Bericht, 1992.
- [2] Csendes, T.: Nonlinear parameter estimation by global optimization - efficiency and reliability. Acta Cybernetica 8(1988) 361-370.
- [3] Ellen Fan. Global optimization of Lennard-Jones atomic clusters. McMaster University, 2002
- [4] G.-Tóth Boglárka. Globális optimalizálás. BME, 2011-2006.
- [5] Jan A. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing. (2005).
- [6] J.P.K. Doye, The network topology of a potential energy landscape: A static scale-free network, Phys. Rev. Lett. 88, 238701 (2002).
- [7] Munk Sándor. Hálózatok fogalma, alapjai. In: Hadmérnök, 2010. V. Évfolyam 3. szám
- [8] Mark Buchanan. Nexus, avagy kicsi a világ – A hálózatok úttörő tudománya. Typotex Kiadó, 2003
- [9] M. LOCATELLI. On the Multilevel Structure of Global Optimization Problems. Computational Optimization and Applications, 30, 5–22, 2005
- [10] M. Locatelli, Fabio Schoen. Global Optimization: Theory, Algorithms, and Applications. SIAM, 2013.
- [11] Pál László. Global optimization algorithms for bound constrained problems. Szeged, 2010.
- [12] R. Horst and P.M. Pardalos. Handbook of global optimization. Kluwer Academic Publishers, Dordrecht, 1995.
- [13] Tamás Vinkó, Kitti Gelle. Networks of Optimization Methods and Problems. In Proceedings of the XII. Global Optimization Workshop, Malaga, Spain, 2014, pp. 141-143.