**Hátizsák probléma optimalizálása természetből inspirált**

**algoritmusok segítségével**

**Mellau Márk-Máté**

**533-as csoport**

**A feladat kijelentése:**

A hátizsák probléma egy kombinatorikai optimalizációs probléma, mely kijelentése a következő: Adott egy tárgyak halmaza, melyeket két szám jellemez: érték és tömeg. Adjuk meg azokat a tárgyakat, amelyeket, ha beteszünk az állandó méretű hátizsákba, az össztömeg a lehető legkevesebb, a tárgyak összértéke a lehető legnagyobb.

A hátizsák probléma már több mint egy évszázada tanulmányozott probléma, az első munkák ezzel kapcsolatban egészen 1897-ig datálhatóak vissza. A név Tobias Dantzighoz köthető.

Az hátizsák problémának különböző változatai ismertek, ezek közül a leggyakrabban tárgyalt a 0-1 hátizsák probléma, ami leszűkíti a tárgyak másolatainak számát nullára vagy egyre, azaz nincs két teljesen egyforma tárgy az adathalmazban.

**Használt Algoritmusok:**

* **Klasszikus genetikus algoritmus (CGA):**

A klasszikus genetikus algoritmus genetikus operátorok segítségével optimalizál egy populációnyi egyedet, ezáltal (ideális esetben) minden új generáció esetén javítva a populáció egyedeinek a minőségét (Fitness érték).

1. Egyedek:

Az egyed egy genetikus algoritmus legkisebb része, kromoszómáit gének alkotják, és egy egyed rátermettségét a Fitness értéke adja. A fenti probléma esetén: a kromoszóma egy boolean vektor, amit logikai értékek tárolnak. Ezek a logikai értékek jelentik egy egyed esetén, hogy beraktunk-e egy adott elemet a hátizsákba, vagy nem (true/false).

1. Fitness érték:

Az egyedek rátermettségét, minőségét megadó érték, ezáltal hierarchiát állíthatunk fel egy populáció egyedei között. A hátizsák probléma esetén ez a fitness érték a tárgyak által elért profit, DE ha a tömeg túlhaladta a maximumot, az egyed fitness értéke nulla.

1. Genetikus operátorok: Kiválasztás, kereszteződés, mutáció:

A genetikus operátorok a genetikus diverzitás fenttartásást szolgálják, ami szükséges feltétele az evolúciónak. A kiválasztás néhány egyed reprodukcióra jelölése, akik létrehozzák a következő generációt. A mi esetünkben ez a kiválasztás randomizált, két egyed két leszármazottat hoz létre keresztezés által.

A keresztezés az egyik leggyakrabban használt reprodukciós folyamat, az utód egyedek mindkét szülőtől örökölnek tulajdonságokat.

A mutáció a genetikai diverzitás fenttartását szolgája, apróbb (vagy kevésbé apró) módosítások egy egyed szerkezetében.

Az implementált klasszikus genetikus algoritmus elitista. A következő generációra csak a legjobb egyedeket viszi tovább, igy a konvergencia alacsonyabb szinten gyors, de könnyedén beragadhatunk lokális minimumokba. Ennek elkerülése végett magas mutációs valószínűséget adunk meg, illetve ha az átlagos fitness érték túlságosan megközelíti a legjobb egyed fitness értékét egy grandiózusabb mutációt viszünk végbe, ahol az első néhány legjobb egyedet megtartunk, apróbb mutációt végzünk az egyedek 1/3-án, és a többi egyedet randomizáljuk.

Az algoritmus futási ideje alacsony és közepes méretű bemeneteken meglepően gyors, és a konvergencia sebessége és a szórás is kielégítő. Nagyobb bemenetek esetén már lassabb. Pontos adatok a későbbiekben.

* **Multi-Memetikus Algoritmus (Második generációs MGA):**

A memetikus algoritmusok olyan hibrid algoritmusok, amik ötvözik a lokális keresések előnyeit a genetikus algoritmusok előnyeivel. Lokális keresési metódusok a szimulált lehűtés, a nyaláb keresés, a tabu keresés, és a hegymászó algoritmus. A multi-memetikus algoritmusok több lokális keresést használnak, megjutalmazva azt, amelyik sikeresen jobb egyedet produkált, így adaptálva egy ideálisabb lokális keresés kiválasztást.

Az algoritmus működése: Meghatározzuk a klasszikus genetikus algoritmushoz hasonlóan a következő generációt, ebből kiválasztunk meghatározott számú elemet, akiken lokális kereséssel javítunk. Miután elértük a maximális generációszámot, lokális kereséssel megpróbálunk még egy utolsót javítani a legjobb egyeden, majd ezt visszatérítjük. Ugyanúgy jelen vannak a genetikus algoritmusok operátorai, van kereszteződés és mutáció, ám az ezek által meghatározott egyedeken még javítunk a lokális keresések segítségével.

A használt lokális keresések:

1. A lokális keresések klasszikusa, a hegymászó algoritmus:

Az algoritmus alapötlete az, hogy kiindulunk egy véletlen megoldásból (ez a mi esetünkben a paraméterként megadott egyed), ezen változtatunk egy kicsit, ha a változtatott megoldás jobb, mint az elődje volt, akkor azt megtartjuk. Addig ismételjük ezt, amig elérjük a leállási feltételt.

Hogy javítsunk a hegymászó algoritmus hatékonyságán, az implementált változat minden iterációban egy random egy és tíz között generált számú szomszédot vizsgál meg, és ezek közül viszi tovább a legjobbat.

1. A Szimulált lehűtés algoritmusa:

Az algoritmus alapelve, hogy valamilyen kicsi valószínűséggel a ’rosszabb’ keresési irányba is elinduljon, ezáltal is kiküszöbölve a beragadást a lokális optimumba.

Az algoritmus egy véletlen lépést tesz, ha a lépés javítja a helyzetet, akkor az mindig végrehajtásra kerül. Ha nem javít a helyzeten, akkor az csak valamilyen kicsi valószínűséggel kerül megtételre, és ez a valószínűség exponenciálisan csökken a lépés ’rosszaságával’.

Az implementált algoritmus lineáris lehűtési folyamatot használ, melynek alfa paramétere eggyel egyenlő.

A változtatási metódus minden esetben egy logikai érték kicserélése az ellenkezőjére.

A multi-memetikus algoritmus lényegesen kevesebb generációval, lényegesen kisebb populációval is jobb konvergálást mutat nagy bemenetek esetén is. A lokális keresés miatt kisebb bemeneteken a futási idő lassúnak tűnhet, közepes és nagy bemeneteken azonban lényegesen jobb a teljesítménye.

**Az algoritmusok összehasonlítása:**

1. **Első összehasonlítás:**

Az előzetes feltételezés az, hogy a multi-memetikus algoritmus gyorsabban fog konvergálni, kevesebb generáció alatt, ugyanakkora populáció mérettel, mint a klasszikus genetikus algoritmus. Ennek a feltételezésnek a lokális keresés bevezetése adja ad létezési jogot. A gyakorlati eredmények viszont alá is támasztják a hipotézist: (Megj.: az eredmények átlageredmények, minden bemeneti fájlra 10x futott le **A képen asztal látható

Automatikusan generált leírás**az algoritmus, ezek eredményei átlagoltuk, és iktattuk be)

**A képen asztal látható

Automatikusan generált leírás**

fenti két táblázat összefoglalja az első gyakorlati összehasonlítását a két algoritmusnak. A klasszikus genetikus algoritmus 100-szor annyi iteráció után is lényegesen nagyobb szórást produkál, és ez a szórás 200 tárgy felett már túl nagy ahhoz, hogy releváns értéket kapjunk. Ezzel szemben a MGA 1000 tárgynál is négy százalékos szórást mutat, igy megbízható optimumközelítést adva. Az első összehasonlítás tehát egyértelműen az MGA nyerte.

1. **Második összehasonlítás:**

**A képen asztal látható

Automatikusan generált leírásA képen asztal látható

Automatikusan generált leírás**Beigazolódott a hipotézis, hasonló paraméterezés esetén az MGA hatékonyabb és pontosabb közelítést ad a klasszikus genetikus algoritmusnál. Második összehasonlítás esetén megpróbáltunk javítani a második algoritmus hatékonyságán:

Mint látjuk, a klasszikus algoritmus a következő paraméterezéssel sem teljesített túl jól. Megnöveltük háromszorosra a populációt, és megtartottuk az előző összehasonlítás generációszámát. Az eredmény leheletnyivel jobb, mint az előző kísérletben, azonban nem jelentős a javulás.

Ezzel szemben az MGA esetén a megkétszerezett generációszámmal és ugyanakkora populációval is jelentősét javított az amúgy is kicsi szórásán.

Ebben a kísérletben más jelentős szerepet játszik a futási idő. Erősen megnövekedett a klasszikus genetikus algoritmus futási ideje, és enyhén az MGA ideje is, azonban ez utóbbinál nem annyira érzékelhető ez, mint az elsőnél. Futási idő összehasonlítása a következőkben.

1. **Harmadik összehasonlítás:**

**A képen asztal látható

Automatikusan generált leírás**A harmadik összehasonlításban megpróbáltunk még jobb javulást elérni mindkét algoritmus esetén a paraméterek növelésével.

**A képen asztal látható

Automatikusan generált leírás**

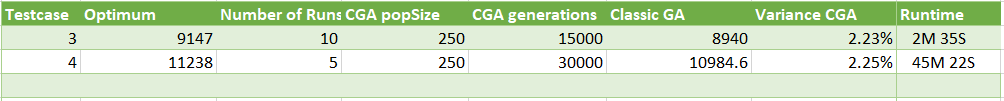
Mindkét esetben sikerült picit több és több százalékot lefaragni a szórásból, ez bizonyítja, hogy fennáll a konvergencia.

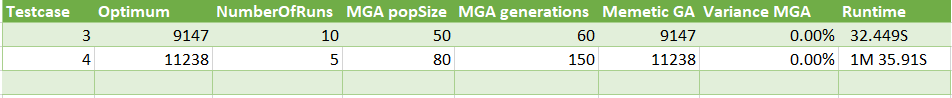
**A klasszikus genetikus algoritmus konvergenciájának a vizsgálata:**

Láthattuk a gyakorlati összehasonlítás vizsgálata közben, hogy a Memetikus Algoritmus konvergenciája vitathatatlan még nagy bemenetek esetén (5,6. testeset) is. Azonban a klasszikus genetikus algoritmus 500 tárgynál már elfogadhatatlanul nagy a szórási százalék. Felmerül a kérdés, hogy hogy csupán helytelen a paraméterezés, vagy egyszerűen eléri az algoritmus a limitet ekkora elemszámnál? A következő adatokkal ezt a kérdést próbáljuk meg vizsgálni:

A paraméter tuning következtetései: Elvégeztünk egy paraméter tuningot a 3. bemenet tárgyhalmazára, amire a CGA 5%-os szórást adott. A következő következtetést vonhatjuk le: A drasztikusan növelt iterációszám sokat segít a konvergencián. Például 20000 iteráció és 300 populáció esetén a szórás lecsökken 0.2%-ra, ezáltal megnövelve a futási időt, de lényeges javulást tudunk felmutatni a konvergenciában. (Ehhez képest a MGA például 60 generációra 50 fős populációval 0.00%-ot hozott, tehát mind a 10 esetben elértük az optimumot)

Futási eredmények:

****

****

Mint láthatjuk a fenti adatok alapján, amelyeket a 3. és a 4. teszteseteken végeztünk, a klasszikus algoritmus futási ideje a tárgyak számának növekedésével exponenciálisan nő. Ezt tetézi az is, hogy a helyes konvergencia fenttartása miatt a generáció és/vagy a populációszámot is növelnünk kell, így egy nem túl hatékony, azonban helyes klasszikus genetikus algoritmust láthatunk.

A Memetikus algoritmus lokális keresései azonban jól teljesítettek ezekben a tesztesetekben is. A futási idő töredéke a CGA futási idejének, és a pontosság is jóval nagyobb (ezen két teszt esetén el is éri minden futás alkalmával a globális optimumot az algoritmus).

Tehát, a CGA sokkal nagyobb futási idővel és populáció mérettel ugyan konvergál a globális optimum felé, azonban a hibrid MGA algoritmus felsőbbrendűsége megkérdőjelezhetetlen.

**Az MGA konvergenciájának a vizsgáltat:**

A fentiekben bizonyosodott be, hogy az MGA algoritmus gyors, jól konvergál, és minimálist szórást produkál, közepes, és nagyobb bemenetek esetén is. Azonban térjünk ki ennek a konvergenciának a felső határaira is. Az eddig látott legnagyobb teszt eset ezer tárgyat tartalmaz, vizsgáljuk meg hogyan teljesít az MGA ebben a próbában:

A futási idő ismét kielégítő, a pontosság sem túlságosan rossz, azonban figyelembe véve hogy csak 180 egyedet tartalmaz a populáció, és csupán 400 generációra volt szükség hogy 97.77%-ban megközelítsük a globális optimumot, elmondhatjuk azt hogy a MGA kiállta a próbát.

Most azonban vizsgáljunk meg egy extra-nagy teszt esetet, a 7. bemenetünk egy 5000 tárgyat tartalmazó szöveges állomány, lássuk mennyire teljesít jól ebben a jóval nagyobb keresési térrel rendelkező feladatban az algoritmus:



A 8.45%-os szórást már nem mondanánk ideálisnak, azonban a futási idő nagysága miatt limitált populációval és generáció számmal indultunk. Ez a közelítés még igy is meglepően jó, tekintve a klasszikus genetikus algoritmus szórását, ami lényegesen kisebb bemenetekre is exponenciálisan nagyobb hasonló paraméterekkel. Ezzel bizonyítást nyert, hogy nagyon nagy bemenetek esetén is jól működik és egész megbízható a multi-memetikus algoritmus. A szóráson javíthatunk nagyobb generáció és/vagy iteráció számmal, azonban ez a futási idő rovására menne, ezért ezt ebben a kísérletben nem tettük meg.

**Következtetések:**

Összefoglalva, és a fenti eredményekkel alátámasztva tehát, bebizonyosodott, hogy a lokális keresések kombinálása a genetikus algoritmusokkal nagyban javítja a konvergenciát, csökkenti a futási időt, a szükséges memóriaigényt és kisebb közelítési szórással is rendelkezik, mint a klasszikus genetikus algoritmusok.

**A továbbiakban:**

A tanulmányozott algoritmusok javítására még volna lehetőség. A klasszikus genetikus algoritmusban más genetikus operátorok használata, például egy szerencsekerekes kiválasztás reprodukcióra vagy kifinomultabb kereszteződés alkalmazása javíthat a konvergencián, ám erre a kísérletünk nem terjedt ki. A multi-memetikus algoritmus esetég lehetséges lenne jobb és gyorsabb konvergenciát elérni más lokális keresések bevezetésével, akár a szimulált lehűtés és a hegymászó primitív algoritmusok mellett, akár helyettük. Egy megfelelő jelölt lehetne a Kvantum-lehűtés algoritmus, amely működése szempontjából hasonló a szimulált lehűtéshez, azonban még kevesebb eséllyel ragadunk lokális optimumba. Egy másik javítási mód lehetne a klasszikus genetikus algoritmus más genetikus algoritmusokkal, például evolúciós stratégiákkal ötvözni, ezzel is felgyorsítani a konvergenciát.

**Források:**

* Bemeneti adatok: <http://artemisa.unicauca.edu.co/~johnyortega/instances_01_KP/>
* Probléma leírása és elemzése: Martello, Silvano; Toth, Paolo [Knapsack problems: Algorithms and computer implementations](https://archive.org/details/knapsackproblems0000mart/mode/2up)
* Mitchell, Melanie (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press
* Moscatto, Cotta, Mendes: [Memetic Algorithms](https://www.researchgate.net/publication/267695201_Memetic_Algorithms)