

Dokumentáció

Az 1. fázis

INVERZ SZŰRÉS:

A GUI-ból származó input adatok beolvasása után végrehajtunk egy inverz szűrést (`decfilter.m`), ami a vágási hely egy approximációjával tér vissza. A szűrést Wiener-szűrővel hajtjuk végre. Ez az érték az inverz szűrésből származó megoldás maximum helyének, illetve ez ettől balra eső zérushelynek a számtani közepe. Ez utóbbi természetesen létezik a Gibbs-jelenségnek köszönhetően.

SZÉTSZÓRÁS:

Az inverz szűréssel meghatározott megoldással és vágási helyével elégedetlenek vagyunk és azt sem tudjuk hogy megfelelő-e, ezért ezen csatorna körül veszünk egy szimmetrikus környezetet. A környezeten belüli összes csatornához létrehozunk egy olyan populációt, amelynek egyedeit az adott csatornában vágjuk le.

A továbbiakban a populációkat és azok tulajdonságait, vagy egyedek tulajdonságait azonosítjuk a csatornával. Tehát, ha például egy csatornához tartozó legjobb egyedről beszélünk, akkor valójában annak a populációnak a legjobb egyedéről beszélünk aki amely az adott csatornában van levágva.

TEREMTÉS:

Az első fázis populációi 100 fősek. Az egyedek paraméterei a GUI-ból származó adatok által meghatározott egyenletes eloszlásokból származik, kivételt képez ez alól a maximum érték, aminek alsó és felső korlátját az objektum maximumának segítségével definiáljuk. A teremtés során amennyiben egy egyed maximum helye nem egyezik a vágási helyével, végzünk egy exponenciális extrapolációt a maximum és az öt követő négy függvényérték segítségével. (Vesszük az értékek logaritmusát, legkisebb négyzetek módszerével illesztünk egy egyenest, majd vesszük ennek az exponenciálisát, ld. `exp_decrease.m` és az így kapott függvényt illesztjük a vágási hely és a maximum közé).

GENETIKUS LÉPÉSEK:

Az egyes csatornákhöz tartozó populációkra külön-külön, 50 iterációig futtatjuk a genetikus algoritmust. Az egyes csatornákhöz tartozó populációkból kivesszük a győzteseket és megnézzük, hogy ezek közül melyik a legfittebb, avagy melyiknek legkisebb a hibája. A vágási hely következő approximációjának azt a csatornát választjuk, amelyik ezen egyednek a vágási helye.

A 2. fázis

SZÉTSZÓRÁS ÉS TEREMTÉS: Ezzel a vágási hellyel továbbra sem vagyunk elégedettek, mert a tesztek során azt tapasztaltuk, hogy a vágás az optimálistól jobbra tolódott. Ebben a fázisban veszünk egy, az előzőnél kisebb környezetet a második approximáció körül, annak reményében, hogy az optimális vágási hely ebben a környezetben van. A módszert adó ötlet az, hogy amennyiben a vágási hely rossz, a hozzá tartozó megoldás nagy valószínűséggel „beragad” a keresési tér egy lokális minimumhelyébe, míg az optimális vágási hellyel rendelkező egy idő után jobb lesz mint bármely más megoldás.

Az első fázisban tárolt paraméterekkel létrehozunk újabb populációkat (a környezeten belül), amit a későbbiek miatt el is tárolunk. A teremtségi paraméterek ez esetben már normális eloszlásból származnak.

GENETIKUS LÉPÉSEK ÉS EGYÜTTES ITERÁCIÓK:

A fent létrehozott populációra újabb 50 generációig futtatjuk a genetikus algoritmust és azt mondjuk, hogy azt a csatornát választjuk vágási helynek, amelyik a legtöbbet fejlődött a két fázis között. Gyakorlatilag azt a felugrási helyet keressük, amelyben a legnagyobb a fejlődési potenciál.

A tapasztalatok azt mutatták, hogy a véletlenek használata miatt nem biztos, hogy jó vágási helyet kaptunk, így az eltárolt populációra (20-szor) megismételjük a futtatást, azaz a fenti utolsó bekezdést. Vágási helynek azt választjuk, amely a 20 ismétlés során a legtöbbször a legjobbnak bizonyult.

Ez a gondolatmenet a meteorológiában már régóta ismert és elfogadott, ott *ensemble*-nek hívják.

EREDMÉNYEK, FOLYTATÁS: Az első két fázis eredményeit ábrázoljuk, ezek ismeretében pedig eldönthetjük, hogy szeretnénk-e áttérni a harmadik fázisra avagy sem.

A 3. fázis

TEREMTÉS: A harmadik fázis során ismételten létrehozunk egy kezdeti populációt, amelynek vágási helyét és az egyedek maximumának várható értékét az előző két fázis szolgáltatja. A többi paraméter a GUI-ból származik. A maximum eloszlása normális, míg a többi paraméteré egyenletes. Továbbá fontos még, hogy a vágási helyet nem változtatjuk!

Ha az összes paramétert az előző fázisokból származtatnánk, akkor a beszűkülés miatt lehetséges, hogy rossz megoldást kapnánk.

GENETIKUS LÉPÉSEK:

A genetikus műveletek iterációs számát egy kis ablakban adhatjuk meg, majd lefut a megadott számú iteráció. Ezúttal a genetikus lépések magja azonos, ám beépítettünk néhány a konvergenciát segítő eljárást: a leftshift automatikus számítását, a nem-egyenletes mutációt, illetve a mutációs valószínűség növelését. (Ezekről bővebben ld. később.)

EREDMÉNYEK, ÚJABB GENETIKUS MŰVELETEK:

Ha lefutott a harmadik fázis, ismét ábrázoljuk az eredményeket és az eddigiekhez hasonlóan ismét dönthetünk a folytatás és megállás mellett.

Amennyiben folytatjuk az algoritmust lehetőségünk van relatív módon újra beállítani a legfontosabb paramétereket: a kezdeti populáció egyedeinek vágási helyét, maximumuk várható értékét és szórását, valamint az iterációs számot.

Amint az újabb iteráció is lefutott ismét a fenti döntéshelyzetbe kerülünk.

Leftshift

A **leftshift**, vagy balra tolás, az az eltolási szám amellyel minden egyed rekonvolváltját a lehető legjobban fedésbe hozunk a mért jellel, úgy hogy ezzel az értékkel balra tolunk.

A **leftshift** számítása is automatikussá vált a harmadik fázis során. Úgy tűnik, a két legfontosabb paraméter (maximum, vágási hely) mellett ez volt a probléma harmadik kulcsa.

A **leftshift**-et úgy számoljuk, hogy kiszámítjuk az mért jel és a rekonvolvált maximumának helyét és képezzük a két érték különbségét, minden egyedre külön-külön.

Joggal vetődne fel a kérdés, hogy ha ez kulcsfontosságúnak bizonyult, akkor miért nem alkalmazzuk már az első fázis során? A válasz nagyon egyszerű: mert ez esetben az algoritmusnak nem lenne lehetősége arra, hogy egymáshoz viszonyítsa a különböző vágási helyeket, így azt nem tudnánk meghatározni a valódi vágási helyet.

Finomabb genetikus módszerek (adaptivitás)

Beépítettünk néhány olyan módszert, amely nagyban képes javítani a program hatékonyságát. Ezek a következők.

Non-uniform mutation: A mutáció egy véletlen Gauss-függvény hozzáadásával történik, melynek paraméterei input adatok. Az algoritmus harmadik fázisa során a mutáló függvény paraméterei közül az amplitúdót a hiba függvényében csökkentjük vagy növeljük, precízen egy f faktorral szorozva módosítjuk. (Csak a harmadik fázis során.)

A faktort a következő képpen számítjuk:

$$f = 1 - \exp\left(1 - \frac{m}{E}\right),$$

ahol m a hibák átlaga, míg E a tapasztalati hiba (*experimental error*, most $E = 0.17$).

Mutációs valószínűség növelése: Tegyük fel, hogy kezdetben a mutáció valószínűsége kezdetben p . Amennyiben a GUI-ban bekapcsoljuk ezt a lehetőséget (**Adaptive**

increase of mutation probability), akkor a 3. fázis során a mutáció valószínűségét az $\frac{n}{2}$ -edik iteráció után egyenletesen p -ről 0.9-re növeljük.