

Projecte de Final de Carrera  
Enginyeria Informàtica

ESTUDI COMPARATIU DE L'ALGORISME BASAT EN UNA  
COLÒNIA ARTIFICIAL D'ABELLES

*Autor:*

Joan T. Matamalas Llodrà

*Consultor:*

Dr. David Isern Alarcón



# TAULA DE CONTINGUTS

- Objectius
- Computació evolutiva (EC)
  - Conceptes
- Algoritmes Evolutius (EA)
  - Algoritme genètic (GA)
  - Evolució diferencial (DE)
- Intel·ligència d'eixam (SI)
  - Optimització amb eixam de partícules (PSO)
  - Algoritmes basat amb una colònia artificial d'abelles
- Experiment
- Funcions d'optimització
- Representació gràfica de les funcions
- Metodologia experimental
- Resultats de les optimitzacions
- Resultats de la comparativa
- Problemàtica detectada
- Millores proposades



# OBJECTIUS

- Exposar els fonaments de la computació evolutiva.
- Comparar l'eficàcia dels ABC envers els algoritmes GA, DE i PSO.
- Detallar les bases funcionals dels diferents algoritmes utilitzats.
- Raonar sobre la selecció del valor dels paràmetres configurables dels diferents algoritmes i els efectes que la seva elecció té sobre el seu funcionament.
- Implementar el prototipus que executarà l'experiment.
- Exposar les conclusions extretes de la realització de l'experiment



# COMPUTACIÓ EVOLUTIVA (EC)

Els algoritmes basats en el paradigma de la computació evolutiva intenten resoldre, mitjançant metàfores basades en l'evolució, problemes de complexitat elevada utilitzant per a tal fi mètodes d'origen estocàstic per guiar el procés de cerca.

Gregory Bateson postula que els organismes biològics s'adapten realitzant ajustos dins el que ell anomena els 2 grans sistemes estocàstics: L'evolució natural i la ment. Seguint aquesta divisió, els algoritmes basats en computació evolutiva poden classificar-se com:

- Algoritmes evolutius (basats en l'evolució natural)
  - Algoritmes genètics
  - Programació genètica
  - Estratègia evolutiva
  - ...
- Algoritmes d'intel·ligència d'examen (basats en comportament social)
  - Optimització utilitzant colònies de formigues
  - Optimització amb examen de partícules
  - Algoritmes basat amb una colònia artificial d'abelles



# CONCEPTES

Abans d'entrar en més detall en el funcionament dels algoritmes. Convé tenir clars uns quants conceptes sovint utilitzats en aquest camp:

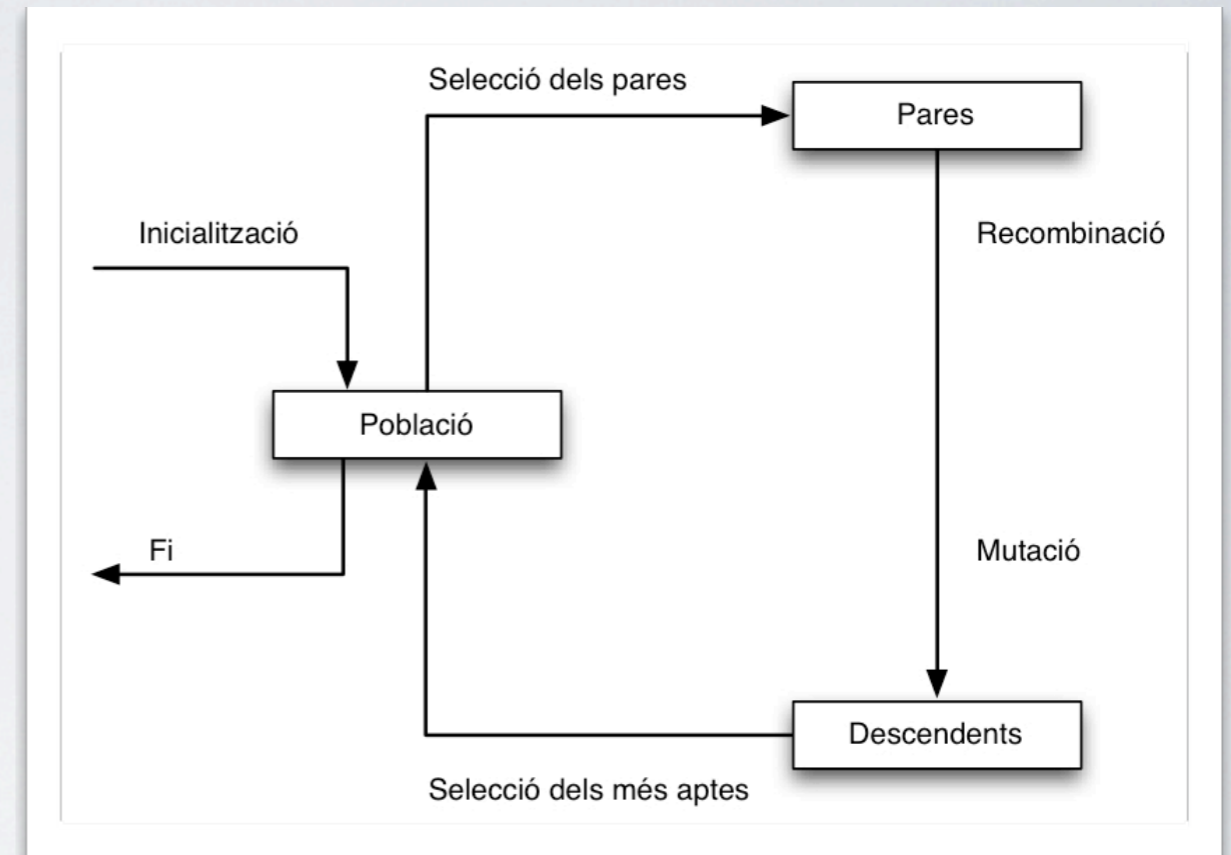
- **Individu:** l'individu representa una solució candidata de la funció que estam intentant optimitzar, i la seva representació és de quina manera estan 'codificats' els seus paràmetres.
- **Població:** Conjunt de totes les solucions candidates presents en un determinat moment a l'algoritme.
- **Funció d'avaluació:** Serveix per mesurar la qualitat de les solucions candidates, parlant en termes evolutius, determina el grau d'adaptabilitat d'un individu davant l'entorn.
- **Espai de solucions:** Domini de cerca d'una determinada funció.



# ALGORITMES EVOLUTIUS (EA)

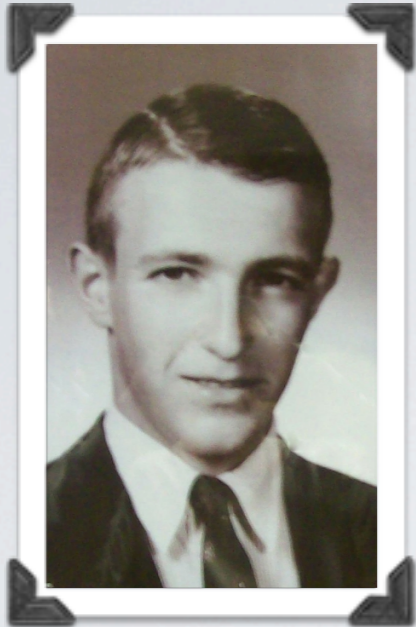
Basen el seu funcionament en alguns mecanismes de l'evolució biològica:

- ♦ **Reproducció:** Aparellament de diversos individus de la població per tal de produir descendents que compartesquin diferents característiques dels seus progenitors.
- ♦ **Mutació:** Canvis aleatoris en el procés de copia de l'ADN. Un dels principals introductors de diversitat a la població
- ♦ **Combinació:** Combinació de l'informació de dos individus per produir-ne un de nou
- ♦ **Selecció:** Selecció dels més aptes, ja que són aquests els que més probabilitats tenen de sobreviure





# ALGORITMES GENÈTICS (GA)



John Holland va ser el primer en introduir el concepte d'algoritme genètic l'any 1975 per descriure els algoritmes que, mitjançant operacions similars a les que s'utilitzen en processos genètics, afronten diferents tipologies de problemes.

El funcionament teòric d'aquests algoritmes està basat en el *schemata theorem*. Aquest explica perquè augmenten dins la població les representacions genètiques de les solucions candidates més aptes, així com les menys aptes van decreixent.

```
begin
  S'inicia la població
  while ¬Condició de fi do
    Avaluació de les solucions candidates que formen la població
    Selecció dels pares que generaran la nova generació
    Encreuament dels genotips dels pares per generar un nou individu
    Procés de mutació per els nous individus
    Selecció dels individus que formaran la nova població
  od
end
```

Pseudocodi de l'algoritme GA



# EVOLUCIÓ DIFERENCIAL (DE)

Storn i Price proposaren aquest sistema com una millora respecte els mètodes de cerca existents, ja que aquest oferia una més ràpida convergència cap a l'òptim.

Comparat amb el GA que acabam de veure aquest presenta algunes diferències:

- Els DE només s'apliquen quan s'utilitzen en conjunció amb funcions d'avaluació reals.
- Tot el pes de la introducció de diversitat a la mostra recau sobre l'operador de mutació.
- No requereix la generació del conjunt de progenitors.

```
begin
  Iniciam la població
  while ¬condicions_fi do
    for i = 0 to len(poblacio) step 1 do
      individu = poblacio[i]
      Cream un vector mutant en base a 3 membres de la població
      Encreuam aquest vector amb l'individu i cream un descendent
      Avaluam l'individu i el descendent
      Afegim a la futura generació el que tengui una millor fitness
    od
  od
end
```

Pseudocodi de l'algoritme DE



# INTEL·LIGÈNCIA D'EIXAM (SI)

Els algorismes que s'han mostrat fins ara estan basats en l'evolució genètica o natural, però l'adaptació també es pot mirar des del punt de vista social, on la intel·ligència emergeix de la interacció social dels individus. En concret es diferencien dos nivells d'adaptació:

- Components d'alt nivell:
  - Formació de patrons entre individus
  - Habilitat de resoldre problemes
- Components de baix nivell (inherents al comportament de cada individu):
  - Avaluacions dels components de l'entorn
  - Comparació de l'estat propi amb els d'individus pròxims
  - Imitació del comportament dels individus que posseeixen qualitats desitjables.



Hi ha moltes estructures socials que a la naturalesa segueixen aquests patrons: les formigues, les abelles, les termites...



# OPTIMITZACIÓ AMB EIXAM DE PARTÍCULES (PSO)



Concebut per Kennedy i Eberhart per simular el comportament social com una representació estilitzada del moviment d'organismes dins un estol d'ocells o un banc de peixos.

Aquí, cada individu es mou per l'espai de solucions en relació el seu coneixement sobre la millor posició aconseguida fins el moment i l'influència de les millors posicions obtingudes per altres membres del seu cercle social.

```
begin
  Generam una població inicial
  Definim el veïns per a cada individu
  while  $\neg$ condicions_fi do
    for  $i = 0$  to  $len(poblacio)$  step 1 do
      individu = poblacio[ $i$ ]
      Desplaçam l'individu tinguen en compte el seu millor fitness passat
      i el millor fitness d'entre els seus veïns
    od
  od
end
```

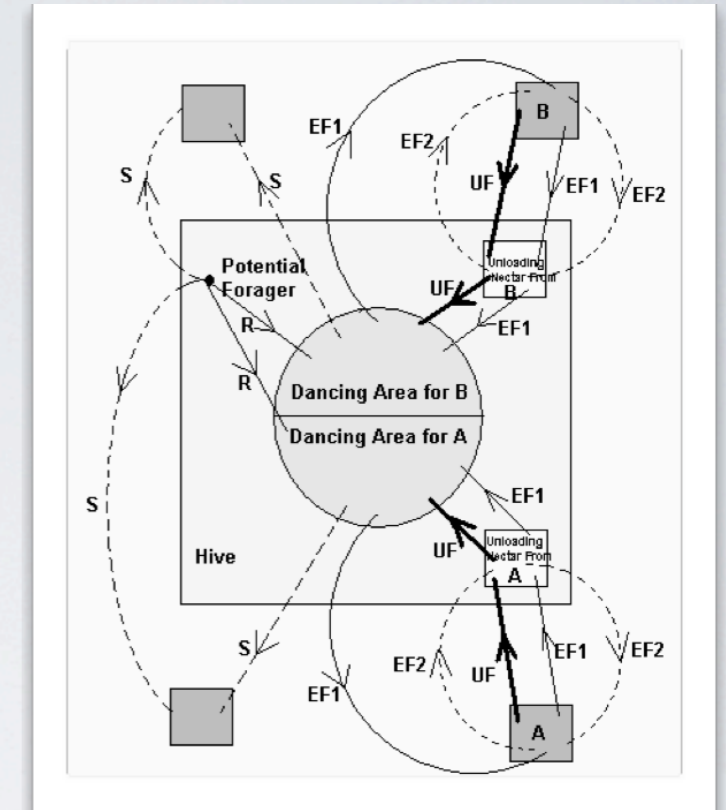
Pseudocodi de l'algoritme PSO



# COLÒNIA ARTIFICIAL D'ABELLES

Dervis Karaboga proposa un algoritme que recree el comportament de l'abella de la mel a l'hora de compartir la informació sobre la localització de fonts d'aliment. Per fer-ho utilitza diferents tipus d'abelles:

- Abelles ocupades (EF): Abelles que actualment estan explotant una font d'aliment i en transmet l'informació.
- Abelles no ocupades (UF): no tenen assignada cap font d'aliment i per tal de trobar-ne alguna poden:
  - Cercar fonts d'aliments aleatòriament per l'entorn (abelles escolta S)
  - Escollir una font d'aliment basant-se en l'informació que les abelles ocupades transmeten quan arriben al rusc



Esquema del funcionament de l'algoritme

```
begin
  Iniciam la població de possibles solucions
  while  $\neg$ condicions_fi do
    Posicionar les abelles ocupades a cada una de les possibles solucions
    Distribuir les abelles no ocupades entre les solucions proporcionalment a la seva valoració
    Enviar a les escoltes a cercar noves fonts d'aliment
    Memoritzar la millor posició trobada fins el moment
  od
end
```

Pseudocodi de l'algoritme ABC



# FUNCIONS A OPTIMITZAR

Els algoritmes han estat provats optimitzant 15 funcions cada una d'elles amb diferents propietats: nombre de dimensions, rang de cerca, modalitat, separabilitat... Aquestes defineixen l'adaptabilitat d'una determinada solució.

La modalitat fa referència al nombre de òptims que té la funció. Aquest és un paràmetre important ja que un dels principals problemes dels algoritmes d'evolució computacional és la convergència prematura a un mínim no global. En base a la modalitat, una funció pot ser:

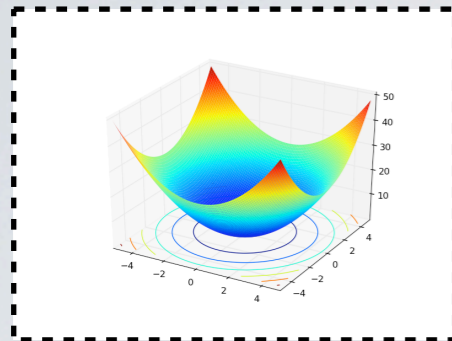
- Unimodal
- Multimodal

La separabilitat parla de si podem dividir una funció de  $p$  variables en  $p$  funcions d'una sola variable. En aquest cas, les funcions poden ser:

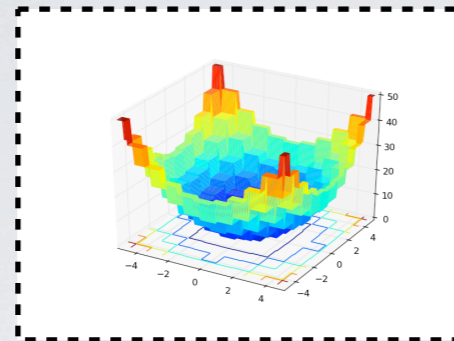
- Separable
- No separable



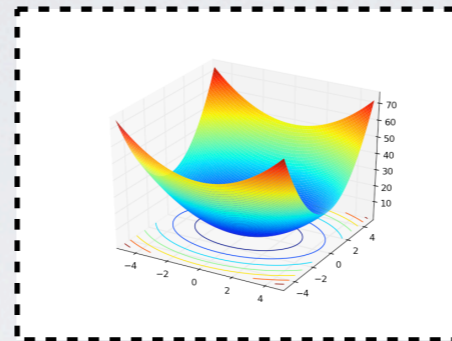
# REPRESENTACIÓ GRÀFICA DE LES FUNCIONS D'OPTIMITZACIÓ



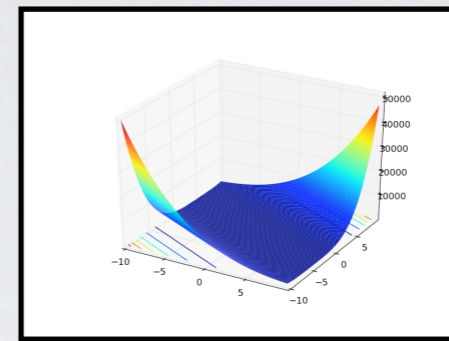
Sphere



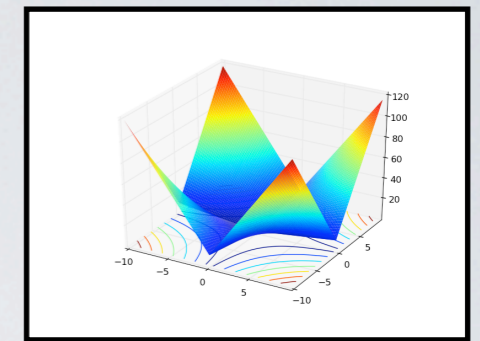
Step



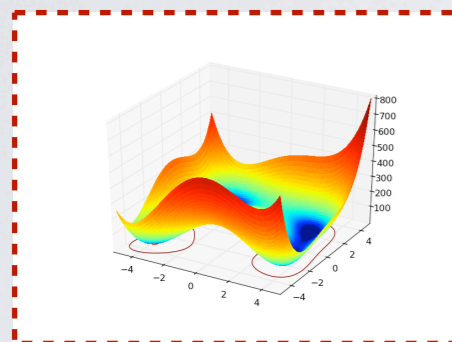
SumSquares



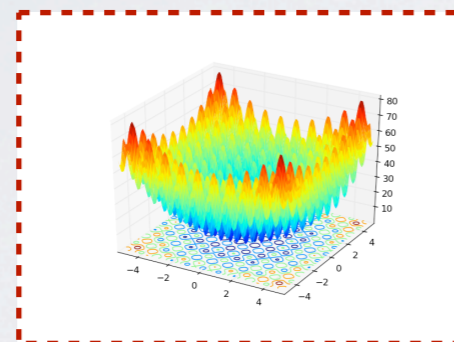
Zakharov



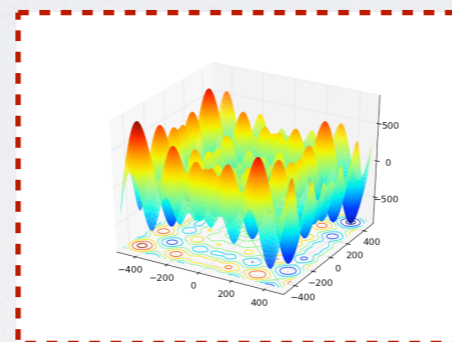
Schwefel 2.22



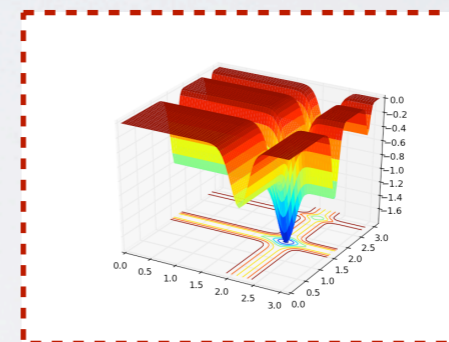
Himmelblau



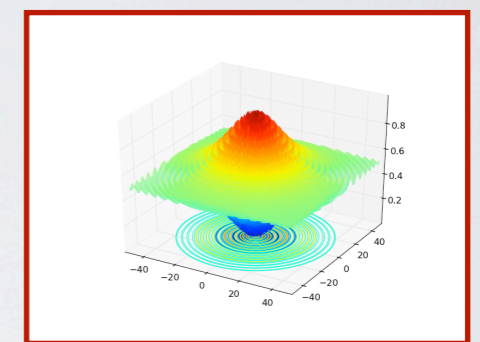
Rastrigin



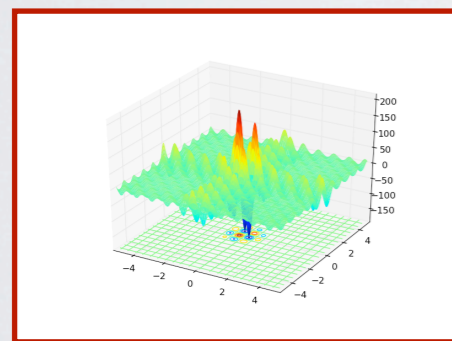
Schwefel



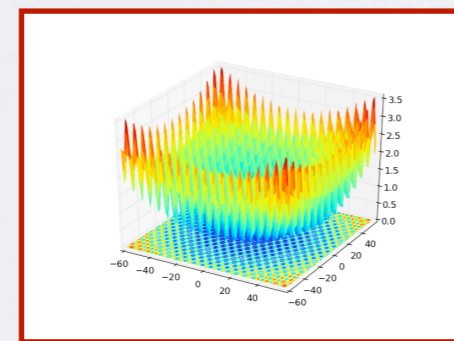
Michalewicz5



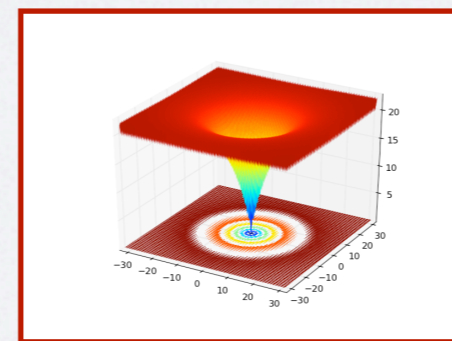
Schaffer



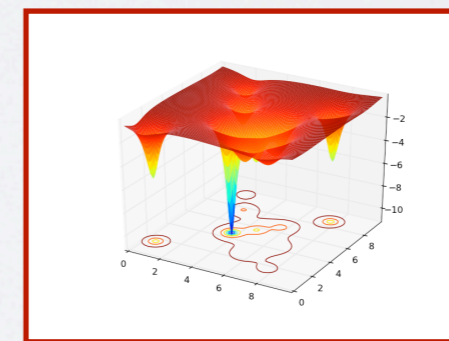
Shubert



Griewank



Ackley



Shekel10

**Unimodals**

**Multimodals**

----- Separables

———— No separables

\*Falta la representació de la funció unimodal i no separable *Colville* ja que aquesta té 4 variables i per tant no pot ser representada gràficament



# METODOLOGIA EXPERIMENTAL

- Es comparen els valors algoritmes tenguen present l'eficàcia d'aquests. S'entén per eficàcia la precisió amb la que un algoritme pot apropar-se al mínim teòric.
- Un algoritme pot acabar la seva execució per un dels dos motius següents:
  - Ha trobat l'òptim amb la precisió desitjada i per tant aquest finalitza amb èxit
  - Ha arribat a un valor màxim d'iteracions sense trobar l'òptim. Fi no satisfactori.
- Donat el funcionament no determinista dels algoritmes, per tal de poder fer una comparativa s'han d'utilitzat eines estadístiques.
- Utilitzant el teorema del limit central, sa repetit el procés d'optimització 31 vegades per cada parella algoritme/funció amb la finalitat de aproximar la distribució a una normal.
- Les comparatives entre els diferents resultats s'han realitzat mitjançant un contrast d'hipòtesi *t-test* per a mostres independents amb una significació 0.05.
- Ja que s'han realitzat diverses comparatives per a cada parell d'algoritmes, s'han de vigilar els falsos positius que es puguin derivar de la *desigualtat de Boole*. S'ha utilitzat la correcció de Bonferroni per aconseguir aquest objectiu.



# RESULTATS DE LES OPTIMITZACIONS

	<b>GA</b>	<b>DE</b>	<b>PSO</b>	<b>ABC</b>
<i>Sphere</i>	—	●	●	●
<i>Step</i>	●	●	●	●
<i>SumSquares</i>	—	●	●	●
<i>Colville</i>	—	○ (84%)	●	—
<i>Zakharov</i>	—	●	●	●
<i>Schwefel 2.22</i>	—	●	●	●
<i>Himmelblau</i>	○ (68%)	●	●	●
<i>Rastrigin</i>	—	—	—	●
<i>Schwefel</i>	—	—	—	●
<i>Michalewicz5</i>	○ (35%)	○ (94%)	○ (97%)	●
<i>Schaffer</i>	○ (3%)	○ (58%)	●	●
<i>Shubert</i>	○ (61%)	●	●	—
<i>Griewank</i>	—	○ (90%)	○ (87%)	●
<i>Ackley</i>	—	●	●	●
<i>Shekel10</i>	○ (19%)	○ (97%)	●	—

- Totes les execucions completades amb èxit
- Optimització parcialment completada amb èxit
- Optimització no finalitzada de manera satisfactòria



# RESULTATS DE LES COMPARATIVES

<b>ABC vs</b>	<b>GA</b>	<b>DE</b>	<b>PSO</b>
<i>Sphere</i>	ABC	—	—
<i>Step</i>	—	—	—
<i>SumSquares</i>	ABC	—	—
<i>Colville</i>	ABC	—	PSO
<i>Zakharov</i>	ABC	—	—
<i>Schwefel 2.22</i>	ABC	—	—
<i>Himmelblau</i>	—	—	—
<i>Rastrigin</i>	ABC	ABC	ABC
<i>Schwefel</i>	ABC	ABC	ABC
<i>Michalewicz5</i>	—	—	—
<i>Schaffer</i>	ABC	ABC	—
<i>Shubert</i>	GA	DE	PSO
<i>Griewank</i>	ABC	ABC	—
<i>Ackley</i>	ABC	—	—
<i>Shekel10</i>	GA	DE	PSO

— No mostra diferència significativa d'eficàcia



# OBSERVACIONS SOBRE ELS RESULTATS

- **GA:** No mostra indicis de convergència a òptims locals però si mostra una lenta convergència cap a l'òptim. Aquest fet l'hi ha impedit en molts de casos trobar el mínim desitjat amb la suficient precisió.
- **DE:** Mostra convergència prematura a mínims locals sense que és pugui restablir el procés de cerca un cop ha convergit.
- **PSO:** Bon comportament a l'hora d'optimitzar funcions unimodals. Mostra indicis de convergència prematura en funcions multimodals amb alt nombre de dimensions. Al contrari que en el cas de l'algoritme DE, el PSO mostra cert grau de recuperabilitat després de convergir a un mínim local.
- **ABC:** No mostra resultats parcials: o optimitza la funció a totes les execucions o no l'optimitza a cap. A més, les tres funcions que no arriba a optimitzar (*colville*, *shubert* i *shekel10*) són funcions amb un baix nombre de dimensions.



# MILLORES PROPOSADES

- Canviar l'operador de mutació de l'algoritme GA per tal d'afavorir una més ràpida convergència a l'òptim.
- En el cas dels DE, introduir noves solucions candidates, generades aleatòriament dins el rang de cerca, quan no es detecti una millora de la solució després de  $n$  iteracions.
- Canviar la topologia de les relacions socials de les partícules a l'algoritme PSO per una topologia Von Neumann. Implementar un mecanisme de repulsió si 2 partícules s'atraquen molt dins l'espai de solucions.
- A l'ABC, revisar el generador de desplaçament per tal d'intentar minimitzar els efectes negatius que aquest té a l'hora d'optimitzar funcions amb poc nombre de dimensions.
- Utilitzar un sistema *híbrid*: generar el conjunt inicial de solucions candidates utilitzant un algoritme de cerca de mínims locals. Després, utilitzar algun dels algoritmes descrits per a finalitzar el procés de cerca.