بسم الله الرحمن الرحیم

دانشگاه تربت حیدریه

پردازش سیگنال دیجیتال

فصل چهارم :

الگوریتم ژنتیک

مجتبی قربانی

هدف این فصل ...

مقدمه :

هنگامی که لغت تنازع بقا به کار می‌رود اغلب بار ارزشی منفی آن به ذهن می‌آید. شاید هم‌زمان قانون جنگل به ذهن برسد و حکم بقای قوی‌ترها!

مقدمه :

البته همیشه هم قوی‌ترین‌ها برنده نبوده‌اند.

مثلاً دایناسورها با وجود جثه عظیم و قوی‌تر بودن در طی روندی کاملاً طبیعی بازیِ بقا و ادامه نسل را واگذار کردند در حالی که موجوداتی بسیار ضعیف‌تر از آن‌ها حیات خویش را ادامه دادند.

ظاهراً طبیعت، بهترین‌ها را تنها بر اساس هیکل انتخاب نمی‌کند! در واقع درست‌تر آن است که بگوییم **طبیعت مناسب ترین‌ها را انتخاب می‌کند نه بهترین‌ها**.

مقدمه :

قانون انتخاب طبیعی بدین صورت است که تنها گونه‌هایی از یک جمعیت ادامه نسل می‌دهند که بهترین خصوصیات را داشته باشند و آنهایی که این خصوصیات را نداشته باشند به تدریج و در طی زمان از بین می‌روند.

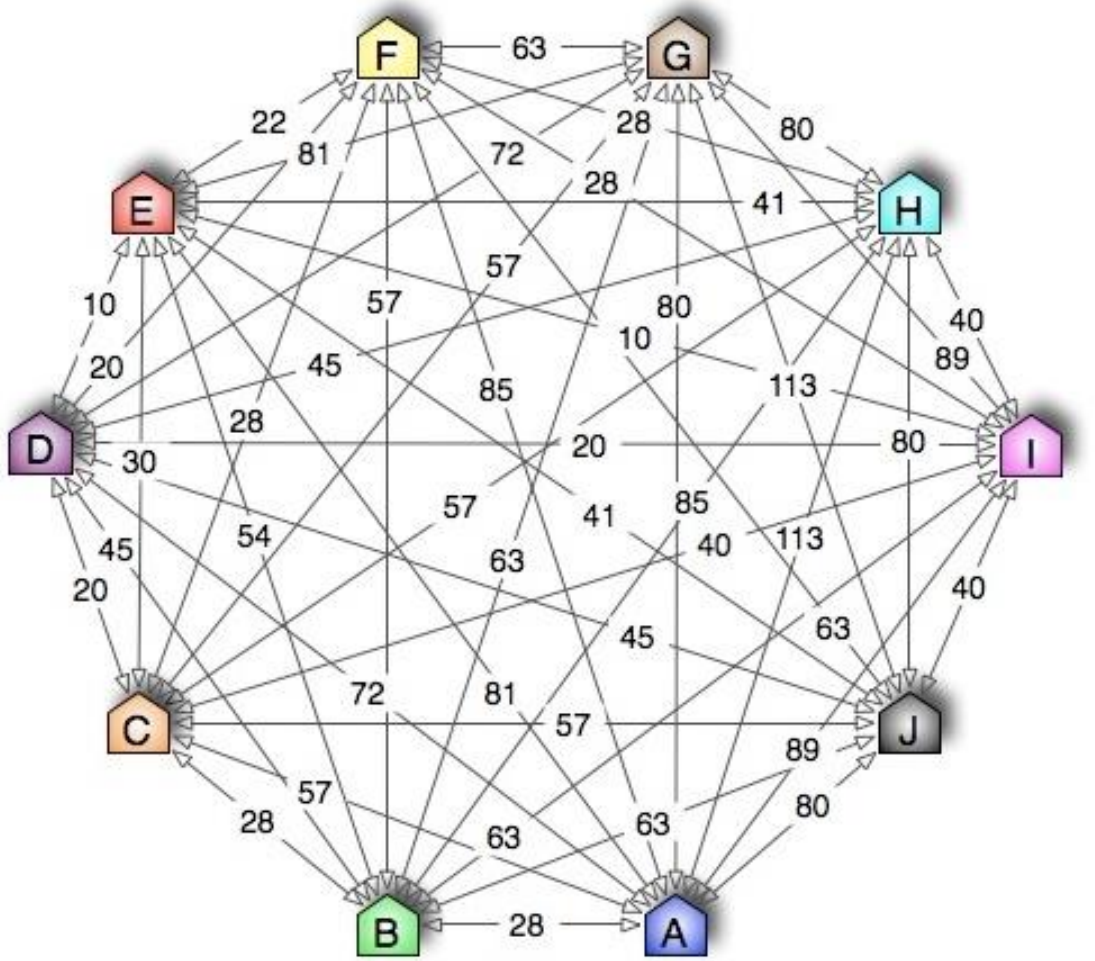
مقدمه :

الگوریتم‌های ژنتیک یکی از الگوریتم‌های جستجوی تصادفی است که ایده آن برگرفته از طبیعت می‌باشد.

الگوریتم‌های ژنتیک برای روش‌های کلاسیک بهینه‌سازی در حل مسائل خطی، محدب و برخی مشکلات مشابه بسیار موفق بوده‌اند ولی الگوریتم‌های ژنتیک برای حل مسائل گسسته و غیر خطی بسیار کاراتر می‌باشند.

به عنوان مثال می‌توان به مسئله فروشنده دوره‌گرد اشاره کرد.

مسئله فروشنده دوره‌گرد :



مقدمه :

در طبیعت از ترکیب کروموزوم‌های بهتر، نسل‌های بهتری پدید می‌آیند. در این بین گاهی اوقات جهش‌هایی نیز در کروموزوم‌ها روی می‌دهد که ممکن است باعث بهتر شدن نسل بعدی شوند.

مقدمه :

روند استفاده از الگوریتم‌های ژنتیک به صورت زیر می‌باشد:

الف) معرفی جواب‌های مسئله به عنوان کروموزوم

ب) معرفی تابع برازندگی (فیت نس)

ج) جمع‌آوری اولین جمعیت

د) معرفی عملگرهای انتخاب

ه) معرفی عملگرهای تولید مثل

در الگوریتم‌های ژنتیک ابتدا به‌طور تصادفی یا الگوریتمیک، چندین جواب برای مسئله تولید می‌کنیم. این مجموعه جواب را جمعیت اولیه می‌نامیم. هر جواب را یک کروموزوم می‌نامیم. سپس با استفاده از عملگرهای الگوریتم ژنتیک پس از انتخاب کروموزوم‌های بهتر، کروموزوم‌ها را باهم ترکیب کرده و جهشی در آن‌ها ایجاد می‌کنیم. در نهایت نیز جمعیت فعلی را با جمعیت جدیدی که از ترکیب و جهش در کروموزوم‌ها حاصل می‌شود، ترکیب می‌کنیم.

مقدمه :

مثلاً فرض کنید گونه خاصی از افراد، هوش بیشتری از بقیه افرادِ یک جامعه یا کولونی دارند. در شرایط کاملاً طبیعی، این افراد پیشرفت بهتری خواهند کرد و رفاه نسبتاً بالاتری خواهند داشت و این رفاه، خود باعث طول عمر بیشتر و باروری بهتر خواهد بود

(توجه کنید شرایط، طبیعی است نه در یک جامعه سطح بالا با ملاحظات امروزی؛ یعنی طول عمر بیشتر در این جامعه نمونه با زاد و ولد بیشتر همراه است).

حال اگر این خصوصیت (هوش) ارثی باشد بالطبع در نسل بعدی همان جامعه تعداد افراد باهوش به دلیل زاد و ولد بیشترِ این‌گونه افراد، بیشتر خواهد بود.

اگر همین روند را ادامه دهید خواهید دید که در طی نسل‌های متوالی دائماً جامعه نمونه ما باهوش و باهوش‌تر می‌شود.

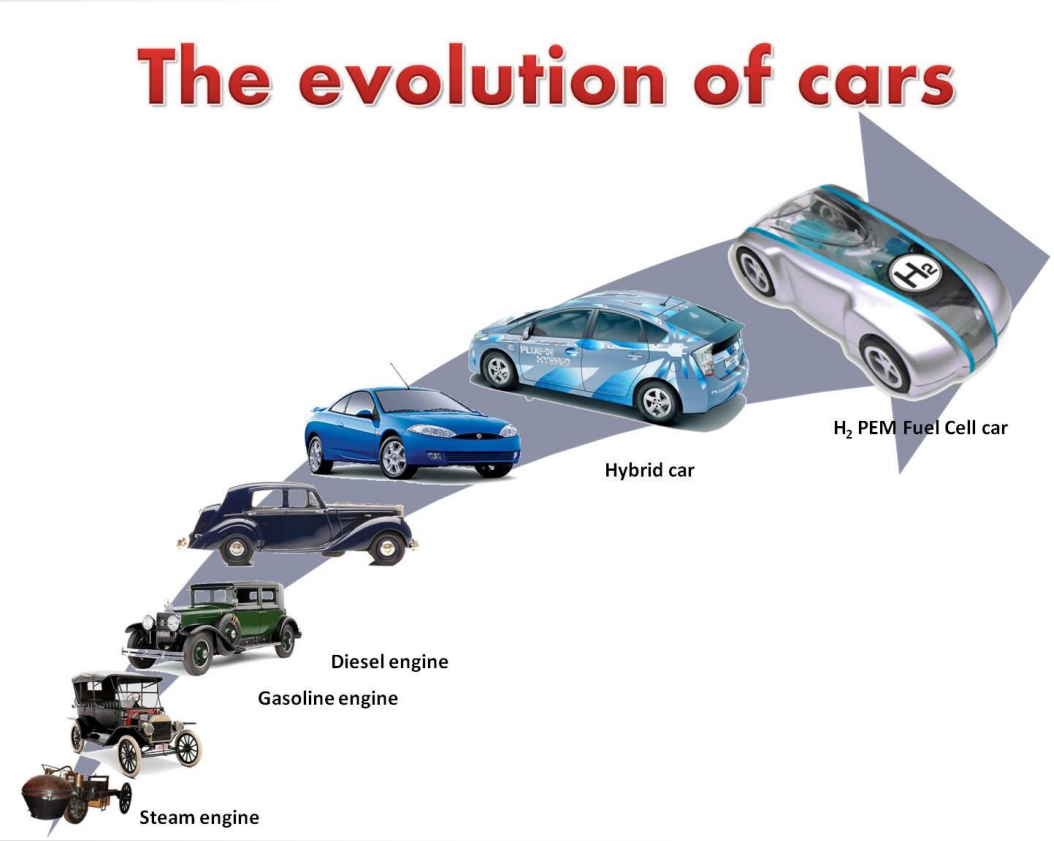
بدین ترتیب یک مکانیزم ساده طبیعی توانسته‌است در طی چند نسل عملاً افراد کم هوش را از جامعه حذف کند علاوه بر اینکه میزان هوش متوسط جامعه نیز دائماً در حال افزایش است.

بدین ترتیب می‌توان دید که طبیعت با بهره‌گیری از یک روش بسیار ساده (حذف تدریجی گونه‌های نامناسب و در عین حال تکثیر بالاتر گونه‌های بهینه)، توانسته‌است دائماً هر نسل را از لحاظ خصوصیات مختلف ارتقاء بخشد.

البته آنچه در قبل ذکر شد به تنهایی توصیف‌کننده آنچه واقعاً در قالب تکامل در طبیعت اتفاق می‌افتد نیست.

بهینه‌سازی و تکامل تدریجی به خودی خود نمی‌تواند طبیعت را در دسترسی به بهترین نمونه‌ها یاری دهد. اجازه دهید تا این مسئله را با یک مثال شرح دهیم:

مقدمه :



پس از اختراع اتومبیل به تدریج و در طی سال‌ها اتومبیل‌های بهتری با سرعت‌های بالاتر و قابلیت های بیشتر نسبت به نمونه‌های اولیه تولید شدند.

طبیعی است که این نمونه های متأخر حاصل تلاش مهندسان طراح جهت بهینه‌سازی طراحی‌های قبلی بوده‌اند.

اما دقت کنید که بهینه‌سازی یک اتومبیل، تنها یک «اتومبیل بهتر» را نتیجه می‌دهد.

مقدمه :

آیا می‌توان گفت اختراع هواپیما نتیجه همین تلاش بوده‌است؟ یا فرضاً می‌توان گفت فضاپیماها حاصل بهینه‌سازی طرح اولیه هواپیماها بوده‌اند؟

مقدمه :

اگرچه اختراع هواپیما قطعاً تحت تأثیر دستاوردهای‌های صنعت اتومبیل بوده‌است؛ اما به هیچ وجه نمی‌توان گفت که هواپیما صرفاً حاصل بهینه‌سازی اتومبیل یا فضاپیما حاصل بهینه‌سازی هواپیماست.

در طبیعت هم عیناً همین روند حکم‌فرماست. گونه‌های متکامل‌تری وجود دارند که نمی‌توان گفت صرفاً حاصل تکامل تدریجی گونه قبلی هستند.

در این میان آنچه شاید بتواند تا حدودی ما را در فهم این مسئله یاری کند مفهومیست به نام تصادف یا جهش.

مقدمه :

طرح هواپیما نسبت به طرح اتومبیل یک جهش بود و نه یک حرکت تدریجی. در طبیعت نیز به همین گونه‌است.

در هر نسل جدید بعضی از خصوصیات به صورتی کاملاً تصادفی تغییر می‌یابند سپس بر اثر تکامل تدریجی که پیشتر توضیح دادیم در صورتی که این خصوصیت تصادفی شرایط طبیعت را ارضا کند حفظ می‌شود در غیر این‌صورت به شکل اتوماتیک از چرخه طبیعت حذف می‌گردد.

در واقع می‌توان **تکامل طبیعی** را به این‌صورت خلاصه کرد:

بقای قوی تر + **جستجوی کورکورانه**

مقدمه :

حال ببینیم که رابطه تکامل طبیعی با روش‌های هوش مصنوعی چیست. هدف اصلی روش‌های هوشمندِ به کار گرفته شده در هوش مصنوعی، یافتن پاسخ بهینه مسائل مهندسی است.

به عنوان مثال اینکه چگونه یک موتور را طراحی کنیم تا بهترین بازدهی را داشته باشد یا چگونه بازوهای یک ربات را متحرک کنیم تا کوتاه‌ترین مسیر را تا مقصد طی کند (دقت کنید که در صورت وجود مانع یافتن کوتاه‌ترین مسیر دیگر به سادگی کشیدن یک خط راست بین مبدأ و مقصد نیست) همگی مسائل بهینه‌سازی هستند.

مقدمه :

روش‌های کلاسیک ریاضیات دارای دو اشکال اساسی هستند.

اغلب این روش‌ها نقطه بهینه محلی را به عنوان نقطه بهینه کلی در نظر می‌گیرند و نیز هر یک از این روش‌ها تنها برای مسئله خاصی کاربرد دارند. این دو نکته را با مثال‌های ساده‌ای روشن می‌کنیم.

در مورد نکته دوم باید بگوییم که روش‌های ریاضی بهینه‌سازی اغلب منجر به یک فرمول یا دستورالعمل خاص برای حل هر مسئله می‌شوند. در حالی که روش‌های هوشمند دستورالعمل‌هایی هستند که به صورت کلی می‌توانند در حل هر مسئله‌ای به کار گرفته شوند.

مقدمه :

نحوه عملکرد الگوریتم ژنتیک روش کار الگوریتم ژنتیک به‌طور فریبنده‌ای ساده، قابل درک و به‌طور قابل ملاحظه‌ای روشی است که ما معتقدیم حیوانات آنگونه تکامل یافته‌اند.

هر فرمولی که از طرح داده شده بالا تبعیت کند فردی از جمعیت فرمول‌های ممکن تلقی می‌شود.

الگوریتم ژنتیک در انسان متغیرهایی که هر فرمول داده‌شده را مشخص می‌کنند به عنوان یکسری از اعداد نشان داده‌شده‌اند که معادل DNAآن فرد را تشکیل می‌دهند.

مقدمه :

موتور الگوریتم ژنتیک یک جمعیت اولیه این‌گونه است که هر فرد در برابر مجموعه‌ای از داده‌ها مورد آزمایش قرار می‌گیرد و مناسبترین آن‌ها باقی می‌مانند؛ بقیه کنار گذاشته می‌شوند.

مناسبترین افراد با هم جفتگیری (جابجایی عناصر DNA) و تغییر تصادفی عناصر DNA کرده و مشاهده می‌شود که با گذشت از میان تعداد زیادی از نسلها، الگوریتم ژنتیک به سمت ایجاد فرمول‌هایی که دقیقتر هستند، میل می‌کنند.

ایده اصلی :

در دهه هفتاد میلادی دانشمندی از دانشگاه میشیگان به نام جان هلند ایده استفاده از الگوریتم ژنتیک را در بهینه‌سازی‌های مهندسی مطرح کرد. ایده اساسی این الگوریتم انتقال خصوصیات موروثی توسط ژن‌هاست.

فرض کنید مجموعه خصوصیات انسان توسط کروموزوم‌های او به نسل بعدی منتقل می‌شوند. هر ژن در این کروموزوم‌ها نماینده یک خصوصیت است. به عنوان مثال ژن ۱ می‌تواند رنگ چشم باشد، ژن ۲ طول قد، ژن ۳ رنگ مو و الی آخر.

ایده اصلی :

حال اگر این کروموزوم به تمامی، به نسل بعد انتقال یابد، تمامی خصوصیات نسل بعدی شبیه به خصوصیات نسل قبل خواهد بود. بدیهیست که در عمل چنین اتفاقی رخ نمی‌دهد. در واقع به صورت هم‌زمان دو اتفاق برای کروموزوم‌ها می‌افتد. اتفاق اول جهش (Mutation) است.

«جهش» به این صورت است که بعضی ژن‌ها به صورت کاملاً تصادفی تغییر می‌کنند. البته تعداد این‌گونه ژن‌ها بسیار کم می‌باشد اما در هر حال این تغییر تصادفی همانگونه که پیشتر دیدیم بسیار مهم است. مثلاً ژن رنگ چشم می‌تواند به صورت تصادفی باعث شود تا در نسل بعدی یک نفر دارای چشمان سبز باشد. در حالی که تمامی نسل قبل دارای چشم قهوه‌ای بوده‌اند.

ایده اصلی :

علاوه بر «جهش» اتفاق دیگری که می‌افتد و البته این اتفاق به تعداد بسیار بیشتری نسبت به «جهش» رخ می‌دهد چسبیدن دو کروموزوم از طول به یکدیگر و تبادل برخی قطعات بین دو کروموزوم است.

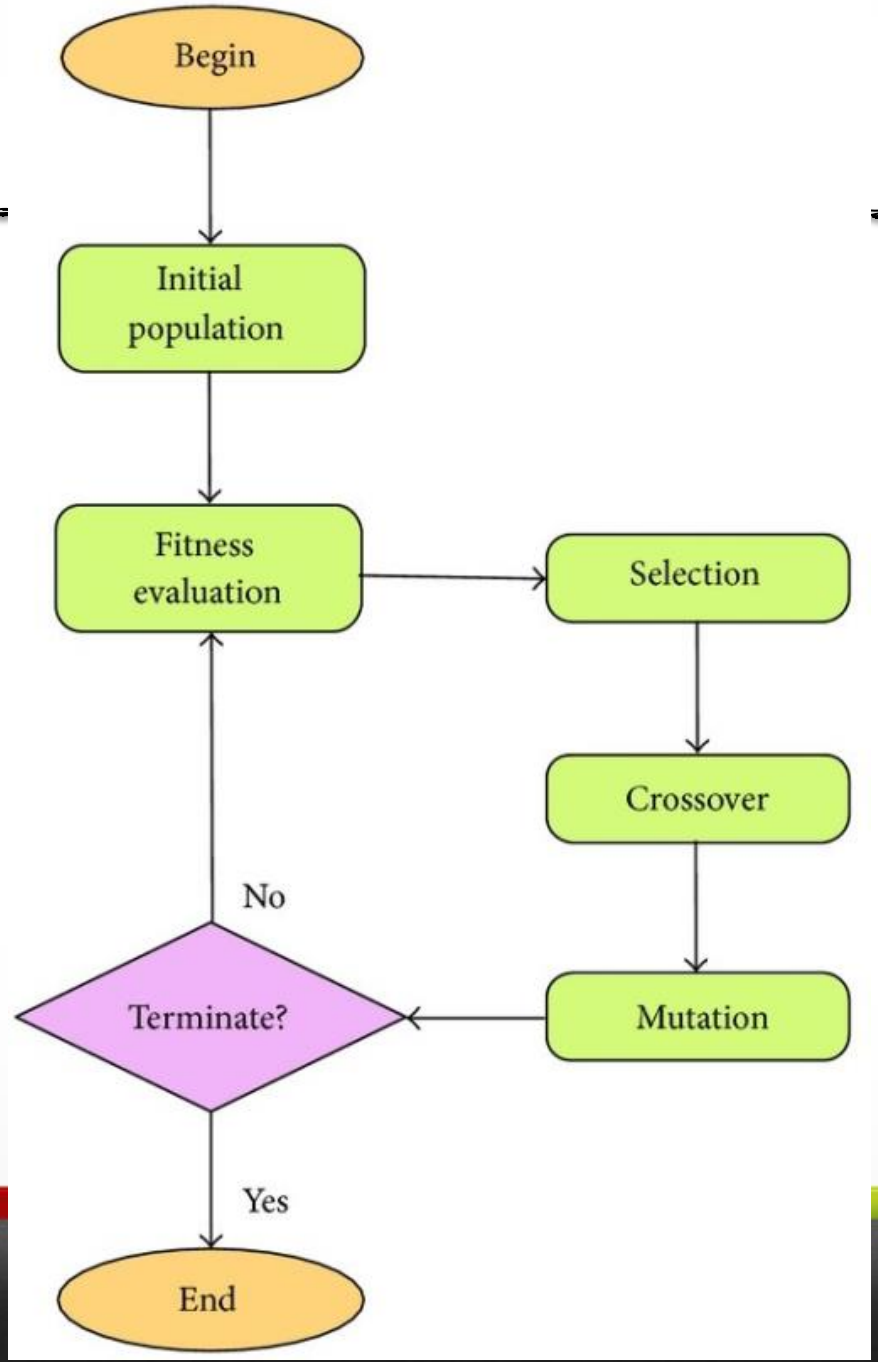
این مسئله با نام Crossover شناخته می‌شود. این همان چیزیست که باعث می‌شود تا فرزندان ترکیب ژنهای متفاوتی را (نسبت به والدین خود) به فرزندان خود انتقال دهند.

روش‌های انتخاب :

روش‌های مختلفی برای الگوریتم‌های ژنتیک وجود دارند که می‌توان برای انتخاب ژنوم‌ها از آن‌ها استفاده کرد. اما روش‌های لیست شده در پایین از معمول‌ترین روش‌ها هستند.

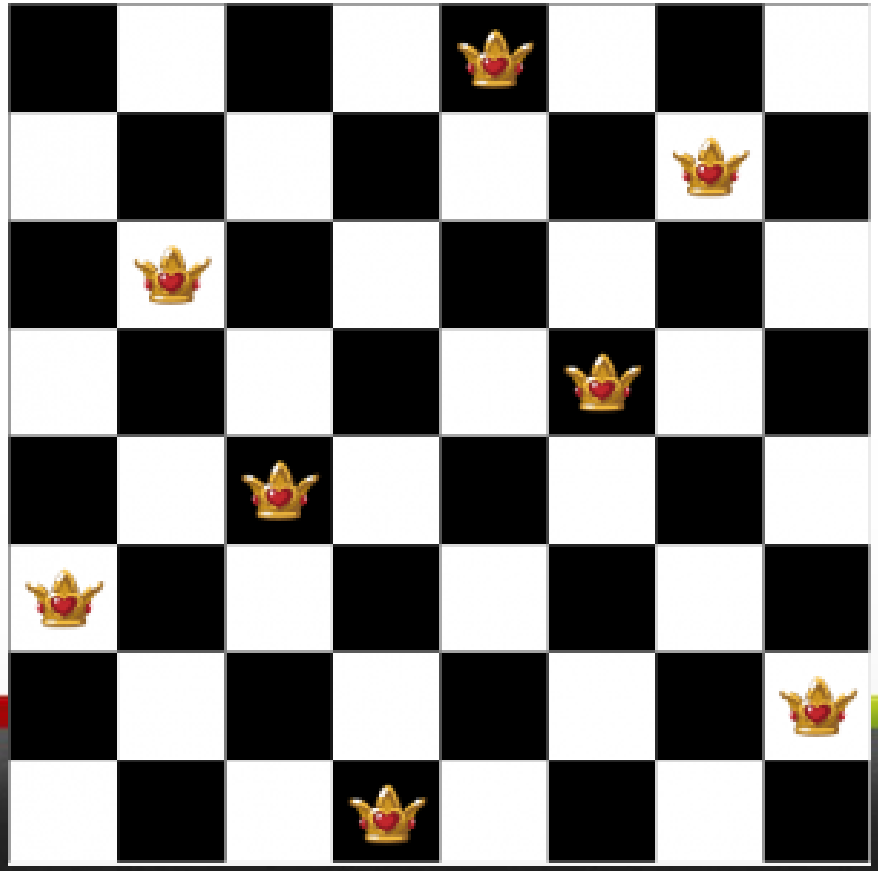
- Selection Elitist
- Selection Roulette
- Selection Scaling
- Selection Tournament
- Hierarchical Selection
- Steady-State Selection
- Rank Selection

الگوریتم :


Begin
Initial population
Fitness evaluation
Selection
Crossover
Mutation
Terminate?
No
Yes
End

مثال عملی :

در این مثال می‌خواهیم مسئلهی ۸ وزیر را بوسیلهی این الگوریتم حل کنیم. هدف مشخص کردن چیدمانی از ۸ وزیر در صفحه شطرنج است به نحوی که هیچ‌یک همدیگر را تهدید نکنند.

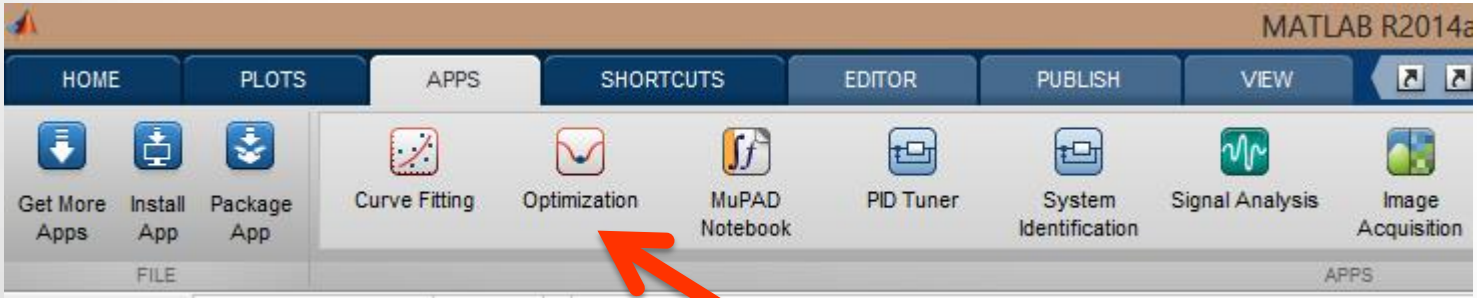


متلب ...

Command Window

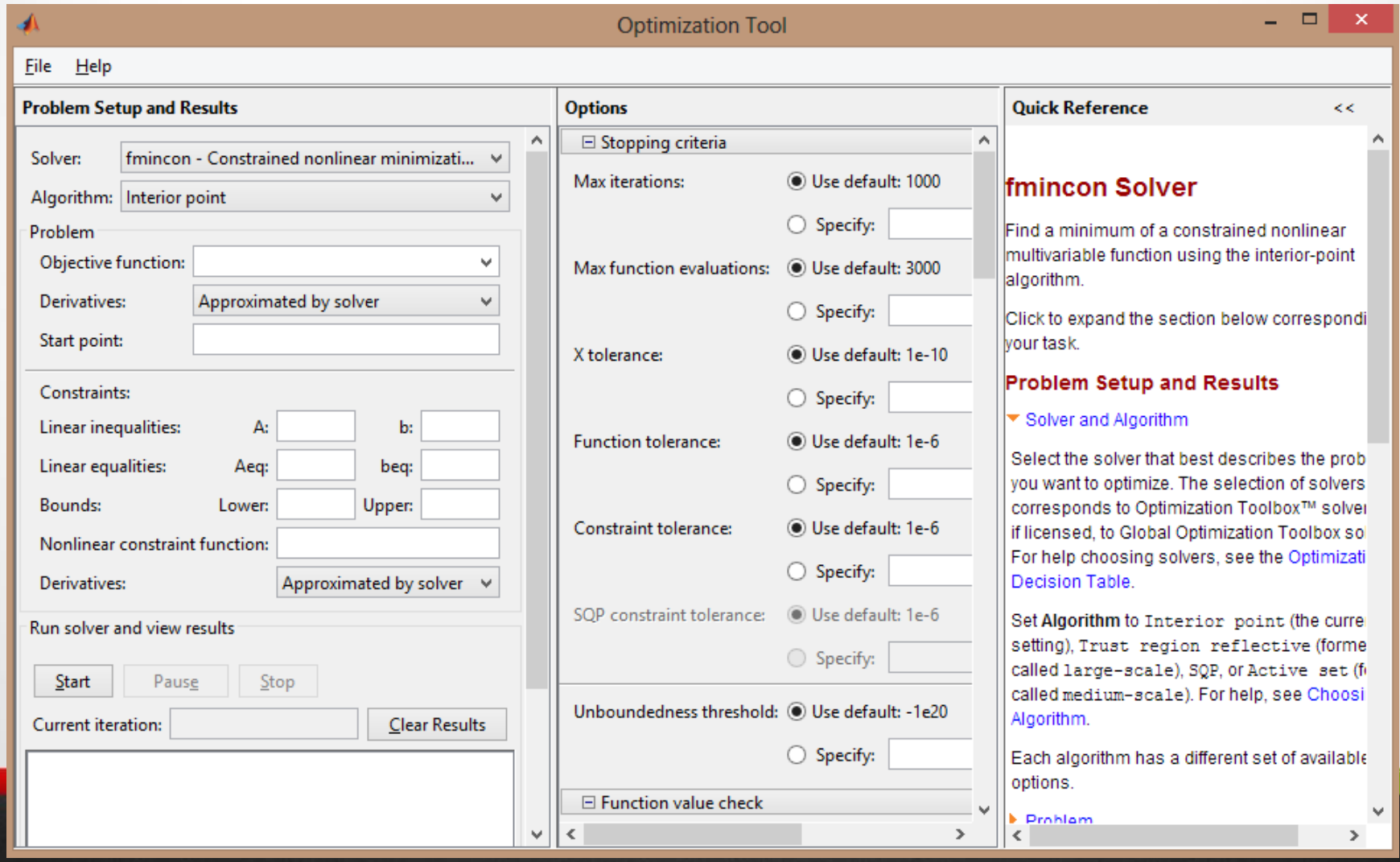
```
>> optimtool
```

1

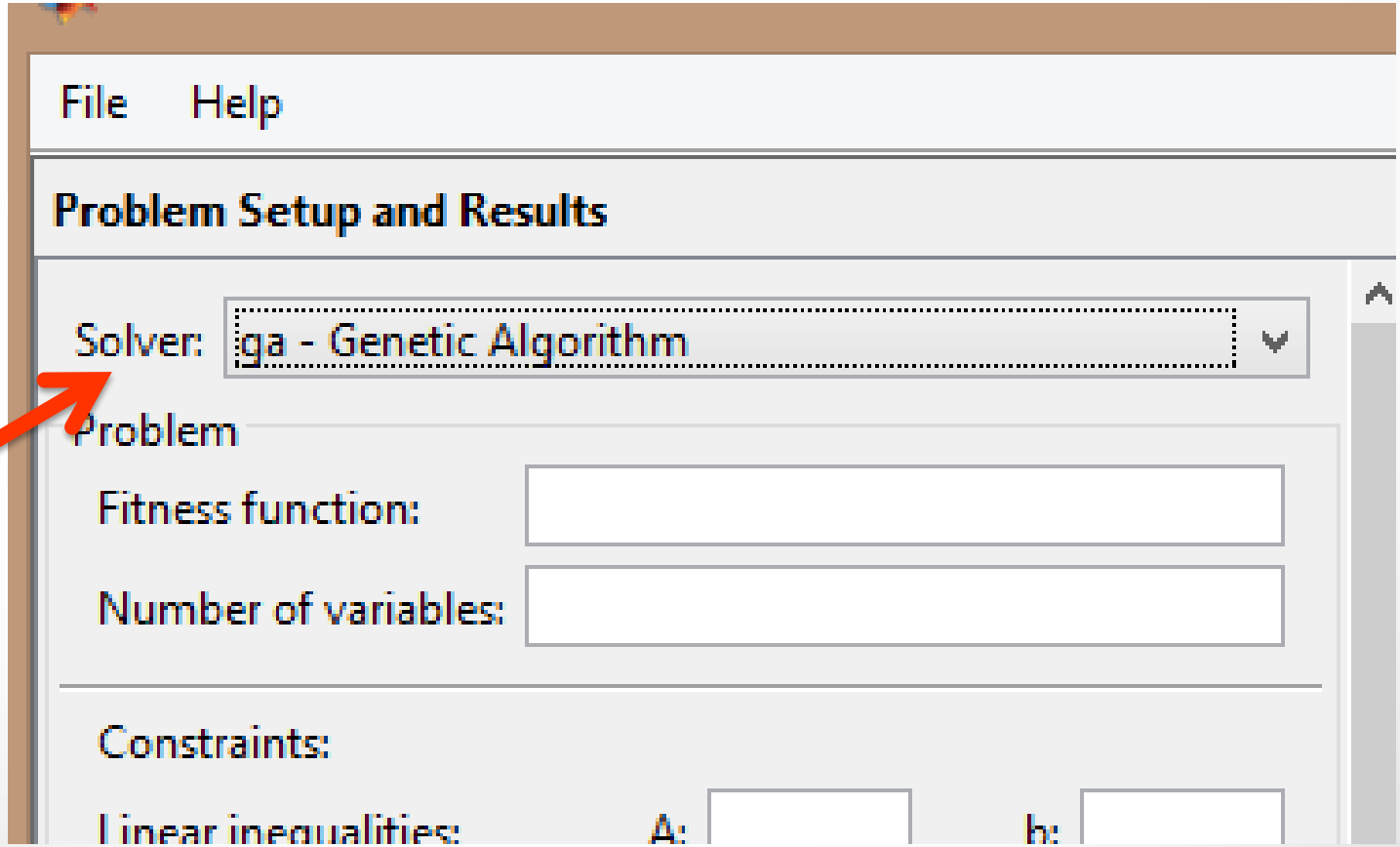


2

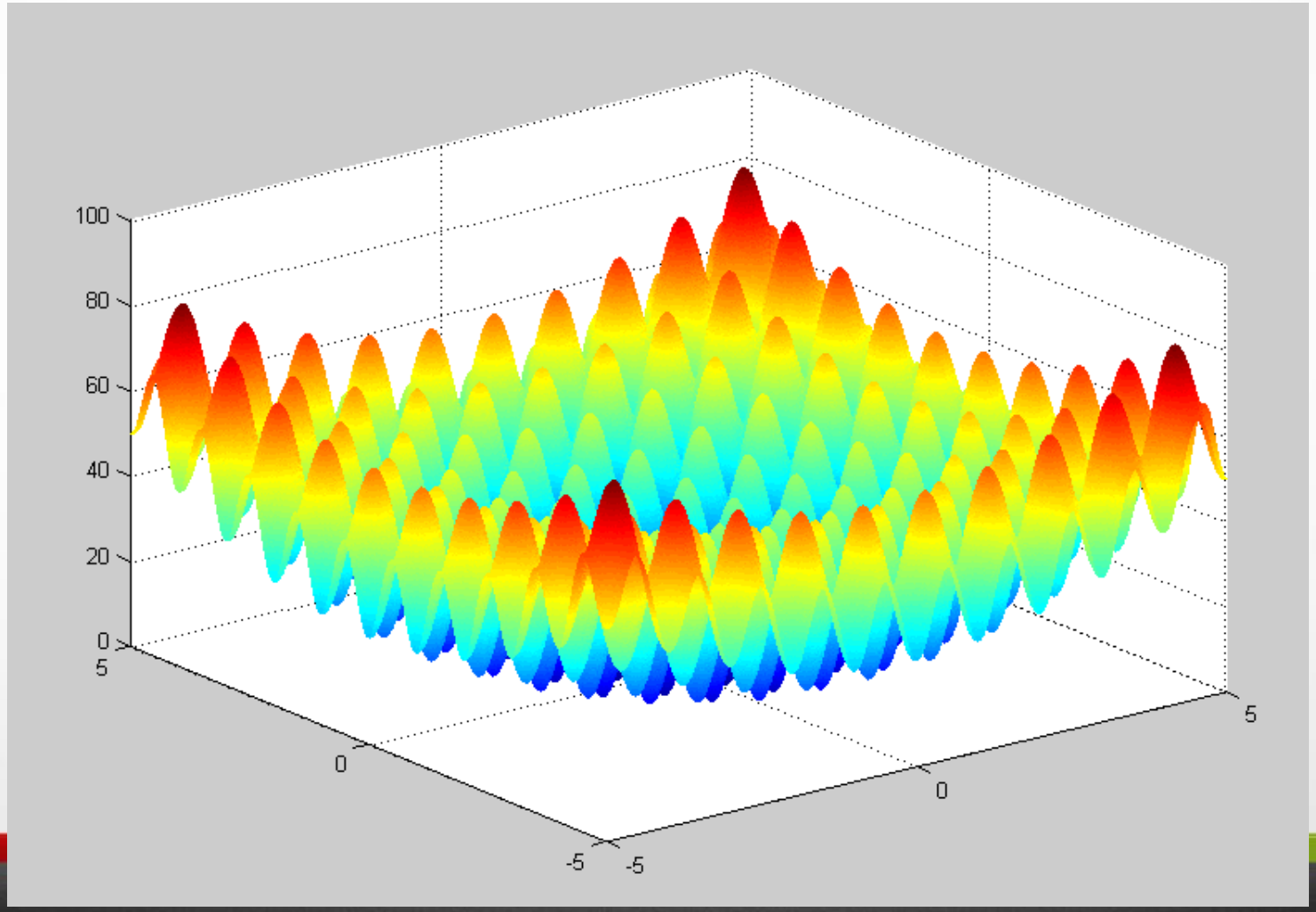
متلب ...


Optimization Tool
File Help
Problem Setup and Results
Solver: fmincon - Constrained nonlinear minimizati...
Algorithm: Interior point
Problem
Objective function:
Derivatives: Approximated by solver
Start point:
Constraints:
Linear inequalities: A: b:
Linear equalities: Aeq: beq:
Bounds: Lower: Upper:
Nonlinear constraint function:
Derivatives: Approximated by solver
Run solver and view results
Start Pause Stop
Current iteration: Clear Results
Options
Stopping criteria
Max iterations: Use default: 1000
Specify:
Max function evaluations: Use default: 3000
Specify:
X tolerance: Use default: 1e-10
Specify:
Function tolerance: Use default: 1e-6
Specify:
Constraint tolerance: Use default: 1e-6
Specify:
SQP constraint tolerance: Use default: 1e-6
Specify:
Unboundedness threshold: Use default: -1e20
Specify:
Function value check
Quick Reference
fmincon Solver
Find a minimum of a constrained nonlinear multivariable function using the interior-point algorithm.
Click to expand the section below correspondi your task.
Problem Setup and Results
Solver and Algorithm
Select the solver that best describes the prob you want to optimize. The selection of solvers corresponds to Optimization Toolbox™ solver if licensed, to Global Optimization Toolbox so For help choosing solvers, see the Optimizati Decision Table.
Set Algorithm to Interior point (the curre setting), Trust region reflective (forme called large-scale), SQP, or Active set (f called medium-scale). For help, see Choosi Algorithm.
Each algorithm has a different set of available options.
Problem

متلب ...


File Help
Problem Setup and Results
Solver: ga - Genetic Algorithm
Problem
Fitness function:
Number of variables:
Constraints:
Linear inequalities: A: b:

متلب ...



متلب ...

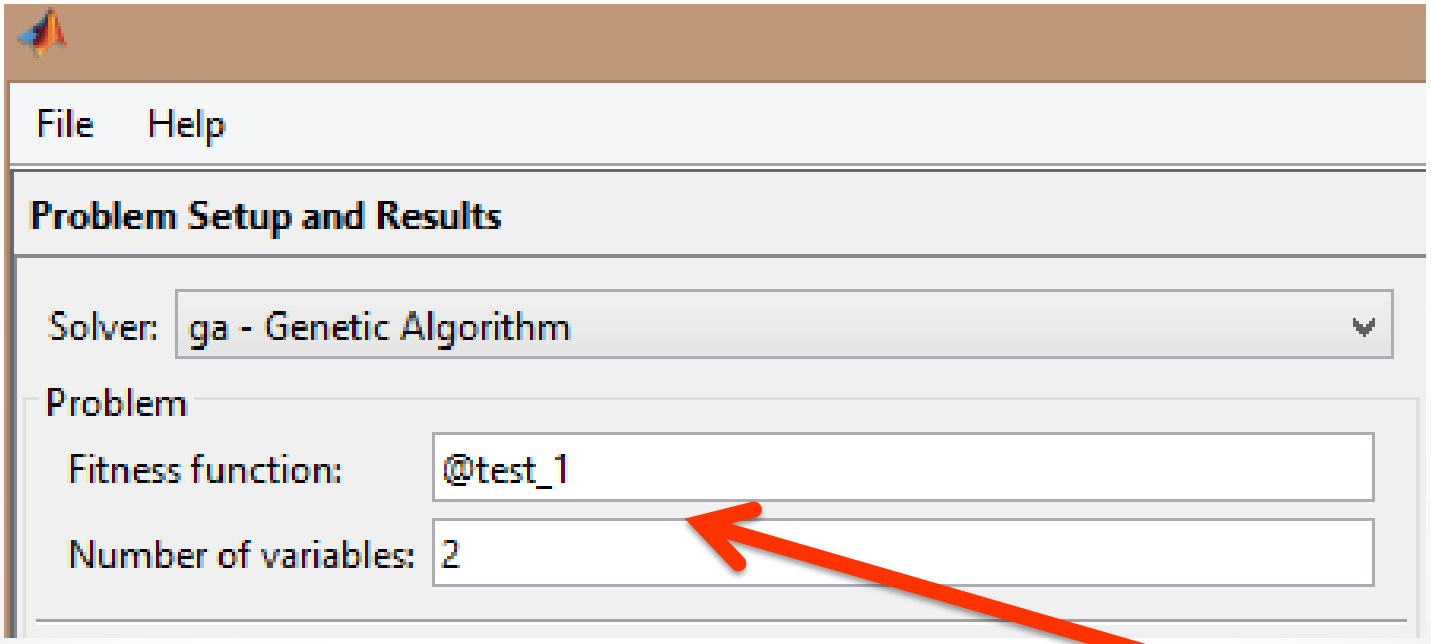
```
1   function f=test_1(x,y)
2
3   f=20+x.^2+y.^2-10*(cos(pi*2*x)+cos(2*pi*y));
4
5   end
```

```
1 -    clc;clear all;close all
2 -    x=-5:0.01:5;
3 -    y=-5:0.01:5;
4 -    [X,Y] = meshgrid(x,y);
5 -    Z=test_1(x,y);
6 -    figure
7 -    mesh(X,Y,Z);
```

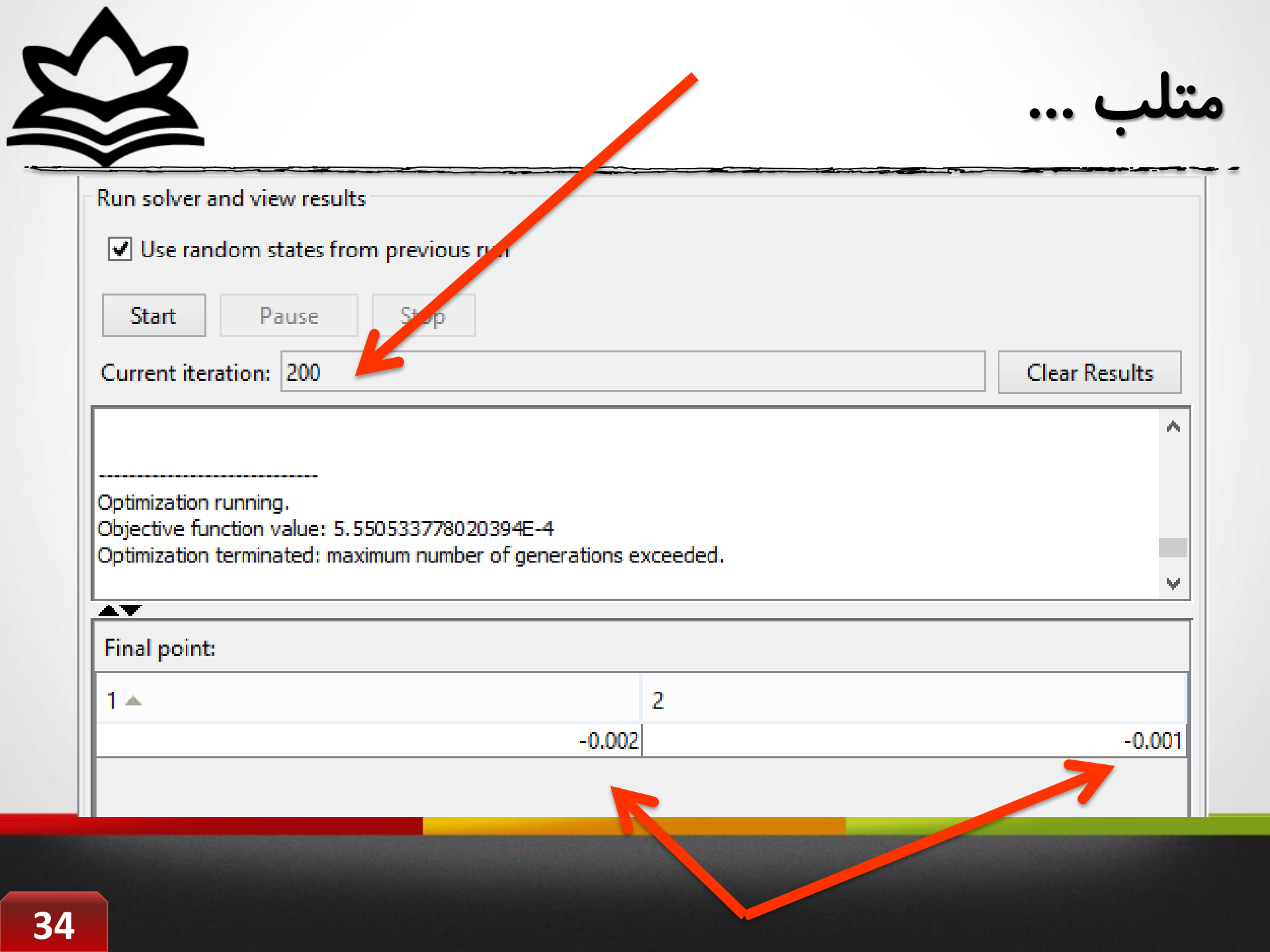
متلب ...

```
1   function f=test_1(x)
2
3   % f=20+x.^2+y.^2-10*(cos(pi*2*x)+cos(2*pi*y));
4   f=20+(x(1)).^2+(x(2)).^2-10*(cos(pi*2*(x(1)))+cos(2*pi*(x(2))));
5
6   end
```

متلب ...

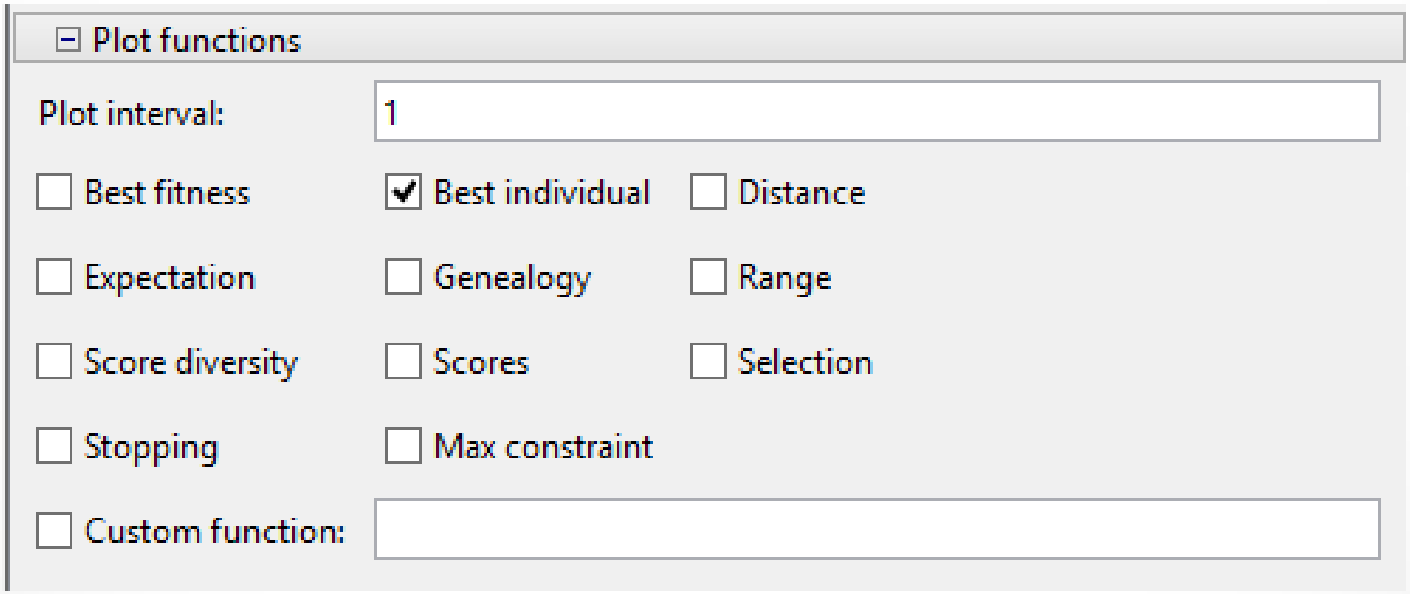

File
Help
Problem Setup and Results
Solver:
ga - Genetic Algorithm
Problem
Fitness function:
@test_1
Number of variables:
2

متلب ...

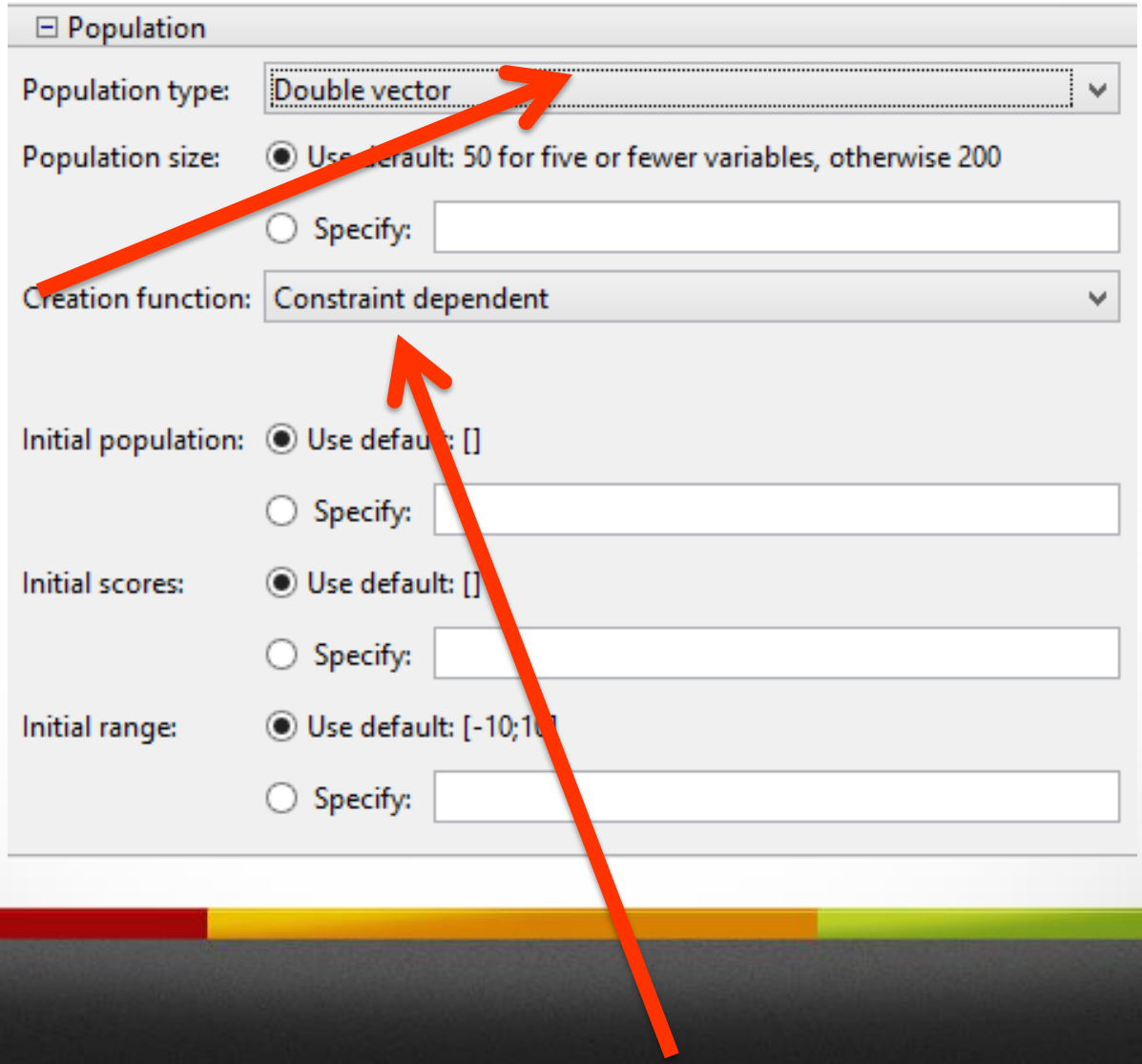

Run solver and view results
Use random states from previous run
Start
Pause
Stop
Current iteration: 200
Clear Results

Optimization running.
Objective function value: 5.550533778020394E-4
Optimization terminated: maximum number of generations exceeded.
Final point:
1
2
-0.002
-0.001

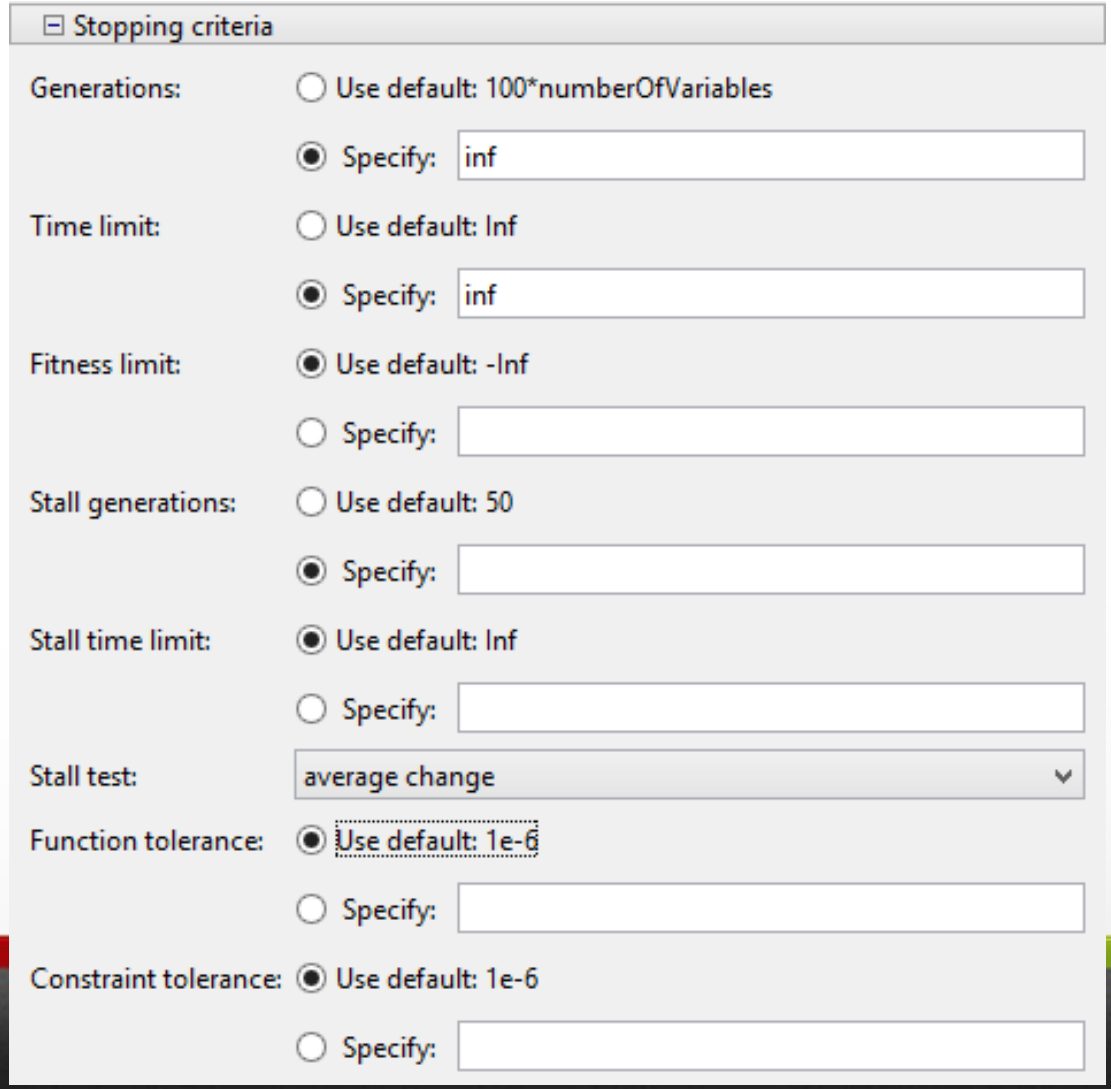
متلب ...


Plot functions
Plot interval:
1
Best fitness
Best individual
Distance
Expectation
Genealogy
Range
Score diversity
Scores
Selection
Stopping
Max constraint
Custom function:

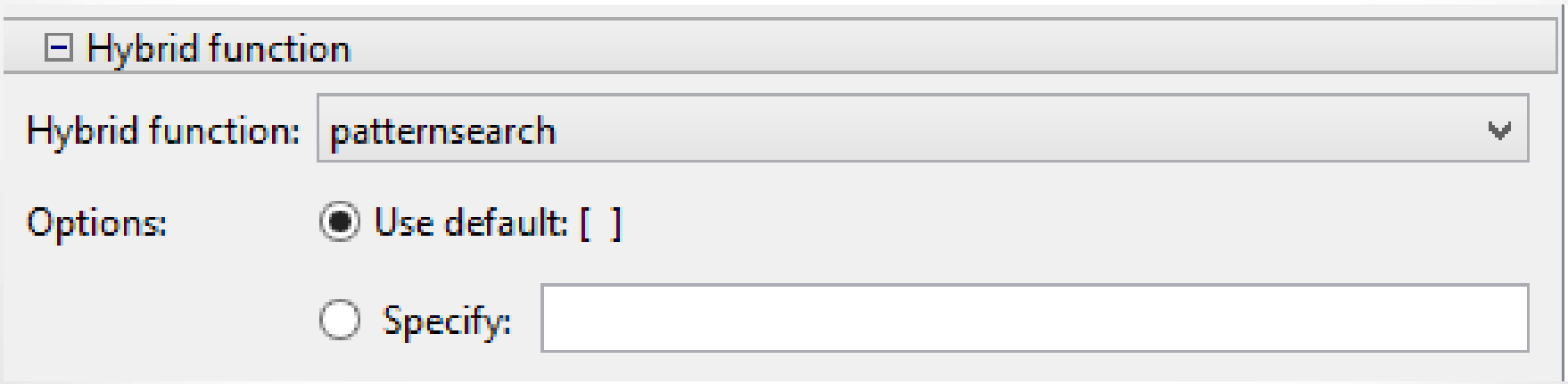
متلب ...


Population
Population type:
Double vector
Population size:
Use default: 50 for five or fewer variables, otherwise 200
Specify:
Creation function:
Constraint dependent
Initial population:
Use default: []
Specify:
Initial scores:
Use default: []
Specify:
Initial range:
Use default: [-10;10]
Specify:

متلب ...


Stopping criteria
Generations:
Use default: 100*numberOfVariables
Specify: inf
Time limit:
Use default: Inf
Specify: inf
Fitness limit:
Use default: -Inf
Specify:
Stall generations:
Use default: 50
Specify:
Stall time limit:
Use default: Inf
Specify:
Stall test:
average change
Function tolerance:
Use default: 1e-6
Specify:
Constraint tolerance:
Use default: 1e-6
Specify:

متلب ...


Hybrid function
Hybrid function:
patternsearch
Options:
Use default: [ ]
Specify:

متلب ...