

به نام خدا

توضیح توابع استفاده شده

تابع اولیه ای که توسط *GUI* اجرا می شود، *GeneticFirst.m* است. تصویر *GUI* طراحی شده:

در قسمت *Input Function* تابع ورودی بر حسب متغیر های $X1$ و $X2$ و ... (حروف بزرگ) نوشته می شود. مثال:

$$21.5 + X1 \cdot \sin(4 \cdot \pi \cdot X1) + X2 \cdot \sin(20 \cdot \pi \cdot X2)$$

Number of Variables: تعداد متغیر هایی استفاده شده.

Population Size: تعداد جمعیت برای هر نسل می باشد.

Generation Size: تعداد حداقل نسل های تولید شده.

Crossover Size: میزان *Crossover* بر حسب درصد که اعداد بین ۰ تا ۱ را شامل می شود.

Mutation Size: میزان جهش بر حسب درصد که اعداد بین ۰ تا ۱ را شامل می شود.

با زدن کلید *Register Function* تابع وارد شده با قالبی مشخص در *FV.m* ذخیره می شود. و

لیست متغیر ها در *Variables* نمایش داده می شود.

با کلیک بر روی هر متغیر و وارد نمودن محدوده متغیر داخل گروه و دقت (تعداد ارقام اعشاری) با

کلیک بر روی *Register Variable* اطلاعات هر متغیر ذخیره می شود.

Variables

Variable range [Down Up]

6

Variable Precision

سپس با زدن کلید *Start*، برنامه شروع به کار نموده و مقدار بیشینه (*MAX*) تابع در قسمت *Answer* نمایش داده شده و مقدار هر متغیر نیز در لیست باکس زیر آن نمایش داده می شود.

Genetic Algorithm

University of Semnan & Amirkabir University of Technology
Artificial Intelligence lecture: Dr. Kourosh Kiani
By Kyvan Jaaferzadeh & Mohamad Nahvi (Students of Semnan University)

Function

Input Function

$21.5 + X1 \cdot \sin(4 \cdot \pi \cdot X1) + X2 \cdot \sin(20 \cdot \pi \cdot X2)$

Number of Variables

2

☒ Max
 ☐ MIN

parameter

Population Size (Range>1)

40

Generation Size (Range>1)

200

Crossover Size (0<=Range<=100)

70

Mutation Size (0<=Range<=100)

0.1

Population Size= 40
 Generation Size= 200
 Crossover Size= %70.000000
 Mutation Size= %0.100000

 Calculate MAX of Function

Variables

-3 <= x1 <= 12.1, Precision=6
 4.1 <= x2 <= 5.8, Precision=6

[4.1 5.8]

Variable range Value [Down Up]

6

8

10

Variable Precision

Command

Example

Clear

Answer

Answer=38.327974
 X1= 11.629810
 X2= 5.618962

توضیح توابع استفاده شده:

FV: در این ام فایل، تابعی که قرار است اجرا شود قرار میگیرد.

Cumulative: احتمال تکرار هر جمعیت را در نسل بعدی به صورت بازه مشخص می کند.

NewPopulation: با توجه به احتمالی که در *Cumulative* تعیین شده است، جمعیت نسل بعد تعیین می شود.

Float2BinCh: متغیرهای هر جمعیت را در یک ماتریس به تعداد بیت های تعیین شده آن به صورت صفر و یکی قرار می دهد. به عنوان مثلا اگر متغیرها ۲۰ بیت داشته باشند، یک ماتریس ۱×۲۰ که هر درایه آن صفر یا یک است می باشد. سپس ژن های هر کروموزوم ها را بهم میچسبانند (مثلا اگر تابع دو متغیر داشته باشیم و یکی از متغیرها ۵ بیت و دیگری ۱۵ بیت داشته باشد هر کروموزوم شامل ۲۰ بیت خواهد شد)

MCrossOver: این تابع به میزان درصد تعیین شده *CrossOver* را اعمال می کند. به این صورت که محل بیت به صورت رندم انتخاب شده و قسمت دوم، دو کروموزوم تعیین شده جا به جا می شوند.

MutationFun: این تابع با توجه به احتمال تعیین شده از بین تمامی بیت های یک نسل به صورت رندم جهش را اعمال می کند.