

بسم الله الرحمن الرحيم

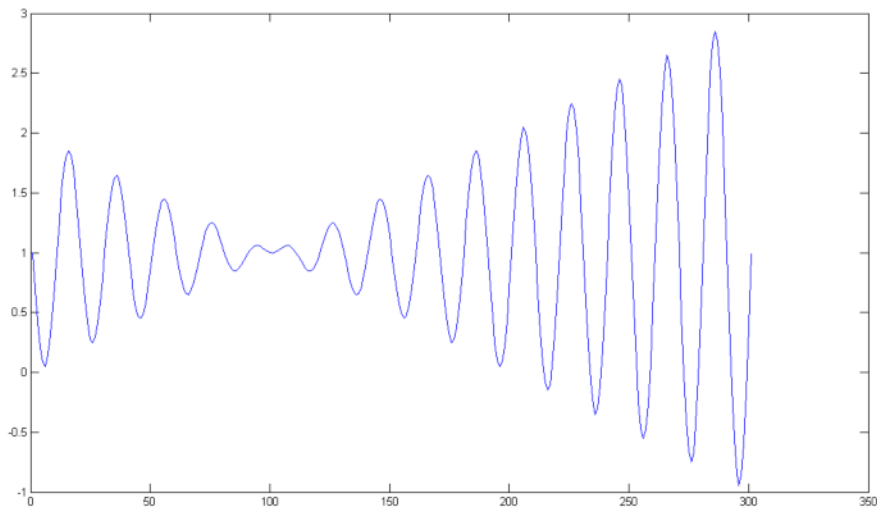
الهم اخرجني من ضلمات وهم و اكرمني به نور الفهم، اللهم افتح علينا ابواب رحمتك والنشورنا خزائن علومك، برحمتك يا الرحمن رحمين

طرح مسئله.

می خواهیم مقدار بیشینه تابع  $F(x) = x \cdot \sin(10\pi \cdot x) + 1.0$  را در فاصله  $-1 \leq x \leq 2$  با روش الگوریتم ژنتیک با دقت شش رقم اعشار محاسبه کنیم.

راه حل:

شکل تابع به صورت زیر است:



کروموزوم ها را به صورت باینری تعریف می کنیم.

چند بیت نیاز داریم؟

برای اینکه بفهمیم کروموزوم ها باید چند بیتی باشند باید ببینیم که  $x$  چند رقم میتواند داشته باشد.

چون با دقت تا ۶ رقم اعشار می خواهیم پس حداقل هفت رقم مورد نیاز است.

چون طول بازه ۳ است پس ۳۰۰۰۰۰۰ حداکثر عددی است که کروموزوم ها می توانند داشته باشند و اما

این عدد بین  $2^{21}(2097152)_{10} < 3000000 < 2^{22}(4194304)_{10}$  است بنابر این به ۲۲ بیت نیازمندیم. پس

هر کروموزوم داری ۲۲ بیت خواهد بود. باید با روشی اعدادی صحیح ایجاد شده را در رنج  $-1 \leq x \leq 2$

نگاشت کنیم. برای این منظور ابتدا رنج را به  $0 \leq x \leq 3$  می بریم. فرض می کنیم که  $x'$  عدد رندم ۲۲

بیتی باشد

$$x' \times \frac{3}{2^{22} - 1}$$

دقت شود که  $x'$  عددی بین  $0 \leq x' \leq 2^{22} - 1$  است (با ماشین حساب ویندوز امتحان کنید). بنابراین

$$\text{رابطه بالا } 0 \leq x' \times \frac{3}{2^{22} - 1} \leq 3 \text{ خواهد بود، حال داریم}$$

$$x = -1 + x' \times \frac{3}{2^{22} - 1}$$

بنابراین داریم

$$-1 \leq x \leq 2$$

برای درک بهتر محاسبات انجام شده به صورت باینری تمامی محاسبات را با عبارات باینری در متلب انجام می دهیم. برای این منظور اعداد اعشاری بین ۰ تا ۳ را در  $10^6$  ضرب میکنیم تا به صورت اعداد صحیح در آیند. سپس با دستور *dec2bin* اعداد را به صورت باینری در آورده و پس از انجام محاسبات با دستور *bin2dec* به حالت عادی بر می گردانیم.

## آغاز برنامه

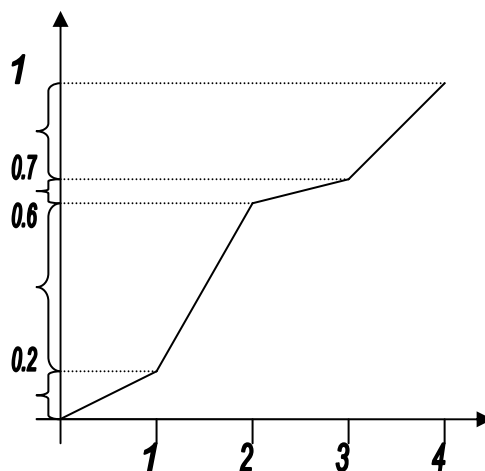
در آغاز چند مقدار پیشفرض را در نظر می گیریم:

*Mutation*=0.1%;

*PN*=50%□

*Mutation* نشان دهنده درصد جهش اعمال شده نسبت به تعداد بیت های موجود در هر نسل است. مثلاً اگر تعداد هر نسل ۵۰ بوده و هر کدوم از کروموزوم ها ۲۰ بیت باشند در مجموع ۱۱۰۰ بیت داریم که ۰.۱٪ آن برابر ۱ بیت در کل نسل می باشد/  
*PN* نشان دهنده جمعیت هر نسل است.

روش کار به این صورت است که جمعیت هر نسل به تابع اعمال شده و مقادیر خروجی حساب می شوند، حال با توجه به روش آماری *cumulative* میزان احتمال تکرار هر کدام از کروموزوم ها در نسل بعدی را محاسبه می کنیم. *Cumulative* به این صورت عمل می کند که مجموع خروجی ها را محاسبه می نماید، اگر جمعیت هر نسل ۵۰ باشد با اعمال این نسل ها به تابع هدف ۵۰ خروجی به دست می آید، مجموع این ۵۰ خروجی را محاسبه می نماییم و هر کدوم از خروجی ها را بر مجموع آنها تقسیم می نماییم، حال درایه اول ماتریس *cumulative* برابر با اولین تقسیم، درایه دوم برابر با اولین تقسیم + دومین تقسیم، درایه سوم برابر اولین + دومین + سومین تقسیم و ... حال ۵۰ عدد رندم انتخاب می کنیم و به ازای هر عدد، اولین عددی که در ماتریس *cumulative* از عدد رندم بیشتر بود را به نسل جدید انتقال می دهیم، با این کار احتمال اینکه اعداد بزرگتر (هدف) به نسل بعد انتقال داده شوند بیشتر است. به این صورت که عددی که مقدار بزرگتری دارد (به مقصود ما نزدیک تر است) در این روش گام بیشتر را طی می کند



در شکل بالا میله افقی نشان دهنده شماره کروموزوم و نمودار عمودی *cumulative* داده هاست که مقدار نهایی آن ۱ است. روش کار به این صورت است که به تعداد جمعیت عدد رندم تولید می کنیم حال اولین کروموزومی که خروجی آن از مقدار رندم بیشتر است را انتخاب می نماییم و این روند را برای تمام اعداد رندم دنبال میکنیم با توجه به شکل بالا کاملاً مشخص است که احتمالاً اینکده اعداد رندم ایجاد شده در رنج بین ۱ تا ۲ باشند، بیشتر است و به این ترتیب احتمال اینکده تعداد تکرار کروموزوم ۲ بیشتر می شوند و به این ترتیب انتخاب اصلح برای نسل بعد صورت میگیرد.

حال از بین این نسل جدید انتخاب شده *Crossover* و *Mutation* را اعمال می کنیم. *Crossover* ۲۵٪ به این صورت اعمال می کنیم که به تعداد نسل عدد رندم ایجاد می کنیم و با توجه به جایگاه این اعداد رندم که کوچکتر از ۰.۲۵ هستند، *Crossover* را بر آن کروموزوم ها اعمال می کنیم. دقت داشته باشیم که در این مرحله هر تعداد متغییر داریم کنار یکدیگر قرار می گیرند. محلی که از آنجا *Crossover* اتفاق می افتد نیز به صورت رندم مشخص می شود.

پس از اعمال *Crossover* نوبت به اعمال *Mutation* است. برای *Mutation* نیز به این صورت عمل میکنیم که تعداد جمعیت را در تعداد کل بیت ها ضرب می کنیم. به عنوان مثال اگر ۴۰ جمعیت داشته باشیم و هر جمعیت هم در مجموع ۲۰ بیت داشته باشد در کل ۸۰۰ بیت خواهیم داشت حال اگر جهش ۰.۰۱ درصدی مد نظر باشد در هر نسل ۸ بیت باید جهش داده شود. ولی در عمل اینچنین عمل نمی کنیم. در عمل به تعداد بیت های جمعیت عدد رندم ایجاد می کنیم و با توجه به محل اعداد رندمی که کوچکتر از ۰.۰۱ ( جهش به تعداد یک درصد) هستند، شماره کروموزوم ها و محل بیت درون آنها را تعیین می کنیم.

مراحل بالا با توجه به تعداد تکرار تعیین شده، تکرار می شود.

توجه داشته باشید که توقف این روند الگوریتم های متفاوتی دارد.