پروژه‌ی پایانی رایانش تکاملی

عنوان: نسخه‌ی بهبودیافته‌ی الگوریتم تکامل تفاضلی

استاد: آقای دکتر ملک

ارائه دهنده: پدرام یزدی‌پور

شماره دانشجویی: ۹۹۴۴۳۲۴۱

اسفند ۱۴۰۰

فهرست مطالب

توضیح الگوریتم

این الگوریتم دو تفاوت با نسخه‌ی اصلی الگوریتم Differential Evolution دارد.

تفاوت اول: بردار جهش معرفی شده در این الگوریتم در معادله‌ی ۱ آمده است.X۳ یک عضو از اعضای برتر جمعیت است که به صورت تصادفی انتخاب شده است. اعضای برتر جمعیت می‌تواند بین ۱ تا ۳۰ درصد، ۱۱ تا ۴۰ درصد، ۲۱ تا ۵۰ درصد یا ۳۱ تا ۶۰ درصد از شایسته‌ترین اعضا انتخاب شوند. X۱ و X۲ نیز به صورت کاملا تصادفی از کل اعضا انتخاب می‌شوند. X_{best} نیز شایسته‌ترین عضو جمعیت است. تمامی اعضای دخیل در بردار جهش باید متمایز باشند. دو پارامتر F۱ و F۲ نیز باید تنظیم شوند.

معادله‌ی ۱:

$$V_i^g = X_{r_3}^g + F_1 \times (X_{best}^g - X_{r_3}^g) + F_2 \times (X_{r_1}^g - X_{r_2}^g)$$

تفاوت دوم: اجرای Opposition Learning به نحوی که به احتمالی تحت عنوان Jumping Rate برای هر عضو از جمعیت و برای هر ژن، مقدار آن ژن(الل) را از مجموع مقدار کمینه و بیشینه‌ی آن ژن بین کل اعضای جمعیت کم می‌کنیم و یک کروموزوم جدید می‌سازیم. در نهایت بهترین اعضا را میان اجتماع جمعیت اصلی و جمعیت جدید به نسل بعد راه می‌دهیم. استدلال مقاله برای معرفی این عملگر این است که ممکن است قرینه‌ی یک عضو با شایستگی پایین، شایستگی خیلی بیشتری داشته باشد و در واقع برای افزایش تنوع به کار می‌رود.

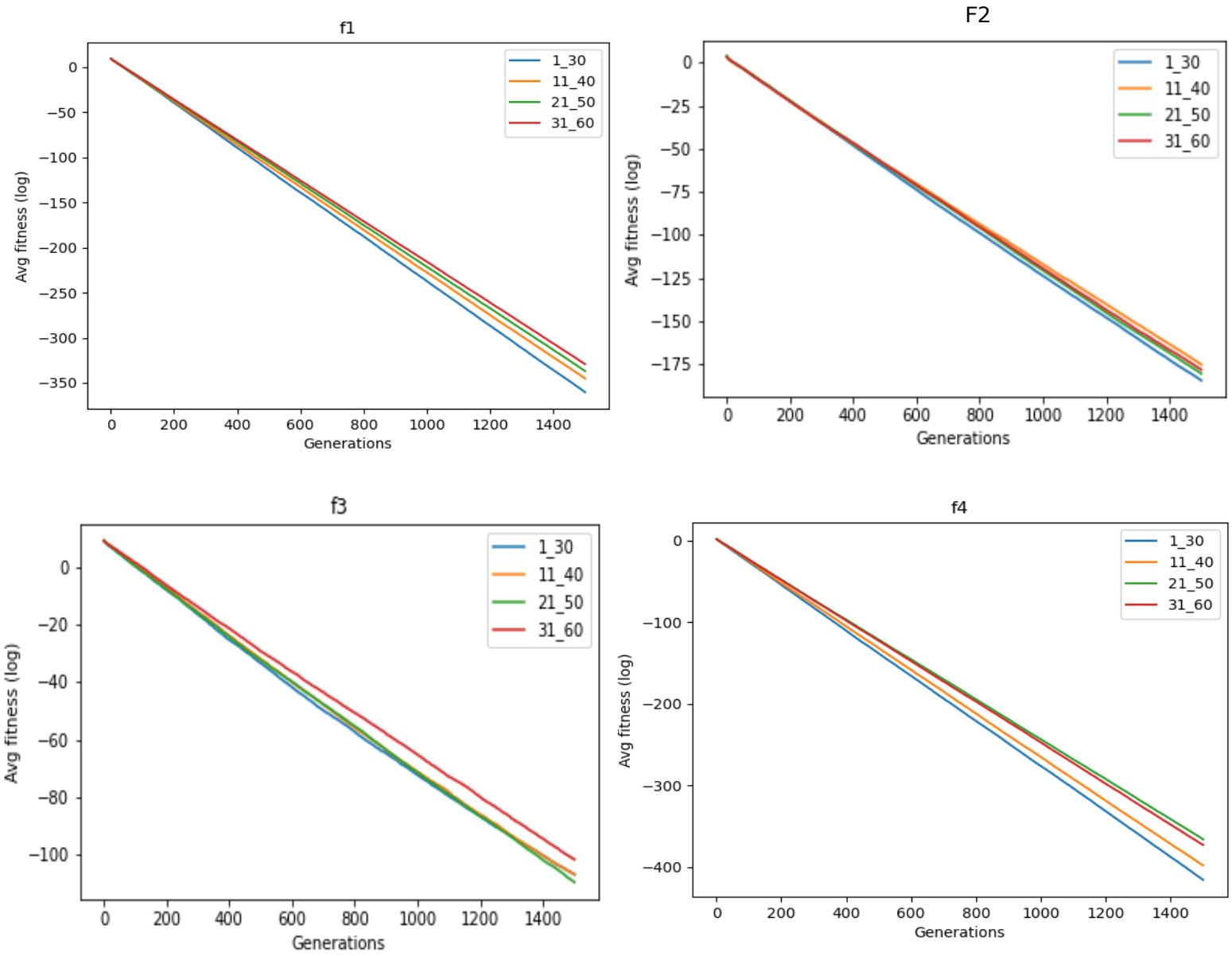
توابع ارزیابی

شکل ۱: توابع ارزیابی

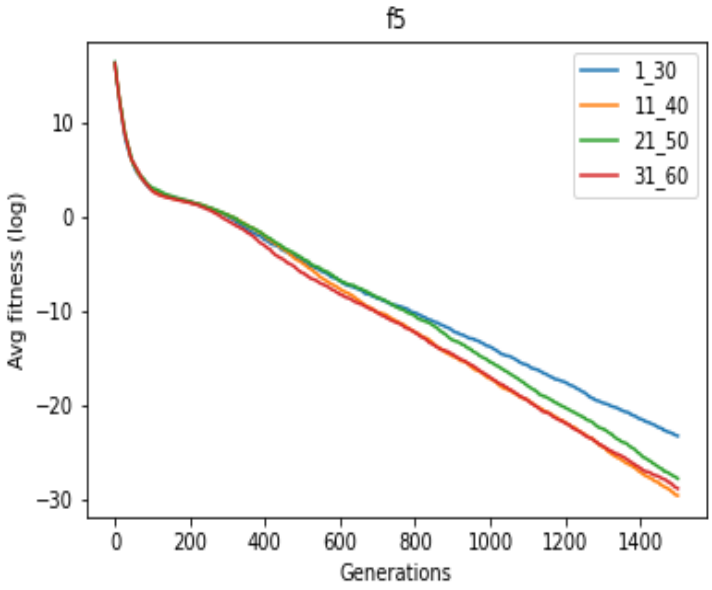
Functions	S	f_{min}
$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$	0
$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100, 100]^D$	0
$f_4(x) = \sum_{i=1}^{D} i x_i^2$	$[-1.28, 1.28]^D$	0
$f_5(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30,30]^D$	0
$f_6(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos\cos(2\pi x_i) + 10)$	$[-5.12,5.12]^D$	0
$f_7(x) = -20\exp\left(-0.2\frac{\sum_{i=1}^{D} x_i^2}{D}\right) - \exp\left(\frac{\sum_{i=1}^{D} \cos(2\pi x_i)}{D}\right) + 20 + \exp(1)$	$[-32,32]^D$	0
$f_8(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^D$	0
$f_9(x) = \sum_{i=1}^{D} \lfloor x_i + 0.5 \rfloor^2$	$[-100,100]^D$	0
$f_{10}(x) = \sum_{i=1}^{D} i x_i^4 + random[0, 1)$	$[-1.28,1.28]^D$	0
$f_{11}(x) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	$[-10,10]^D$	0
$f_{12}(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{(i+1)}$	$[-1,1]$	0

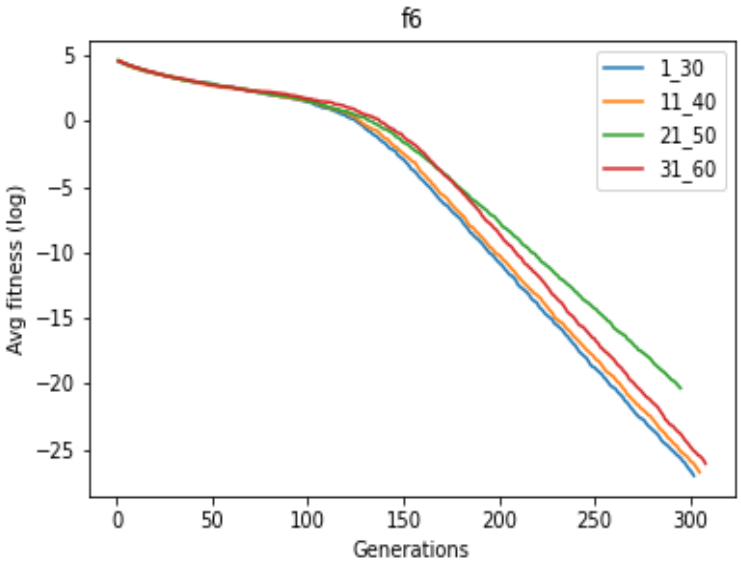
تنظیم پارامترها

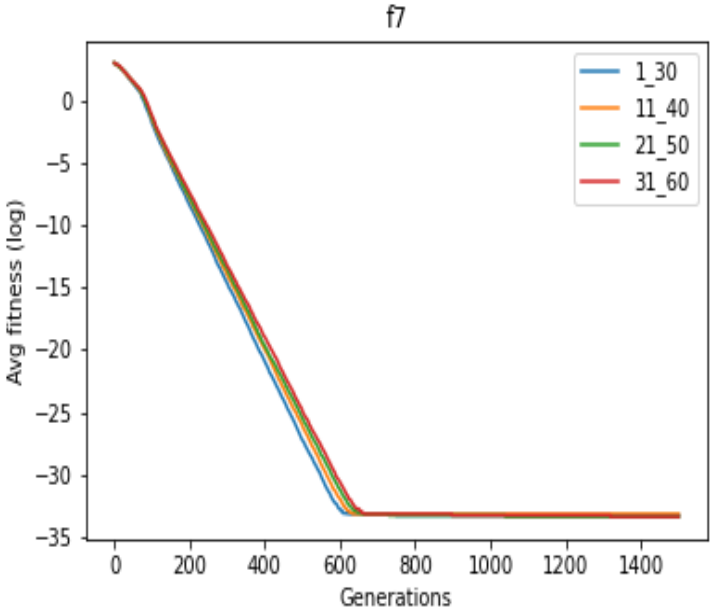
در مقاله یک جدول برای پارامترهای الگوریتم پیشنهاد شده است؛ هر چند در ارزیابی‌های انجام شده دریافتیم این پیشنهادات باید برای هر تابع مجددا تنظیم شوند. یکی از پارامترهایی که متاسفانه چندان به تنظیم آن دقت نشده است و پس از ارائه‌ی نتایج نهایی راجع به انتخاب مقادیر آن تست‌هایی انجام شده مربوط به درصد اعضای برتر جمعیت برای انتخاب عضو X۳ است. به همین منظور، ابتدا روی تمام توابع چهار مقدار مختلف برای انتخاب اعضای برتر جمعیت را که توسط مقاله پیشنهاد شده با هم مقایسه می‌کنیم تا بعد از تعیین آن بتوانیم ارزیابی نهایی را انجام دهیم. شکل ۲ شامل ۱۲ نمودار است. محور قائم این نمودارها معرف میانگین لگاریتم شایستگی به ازای ۱۰ بار اجراست و محور افقی نیز شماره‌ی هر نسل است.

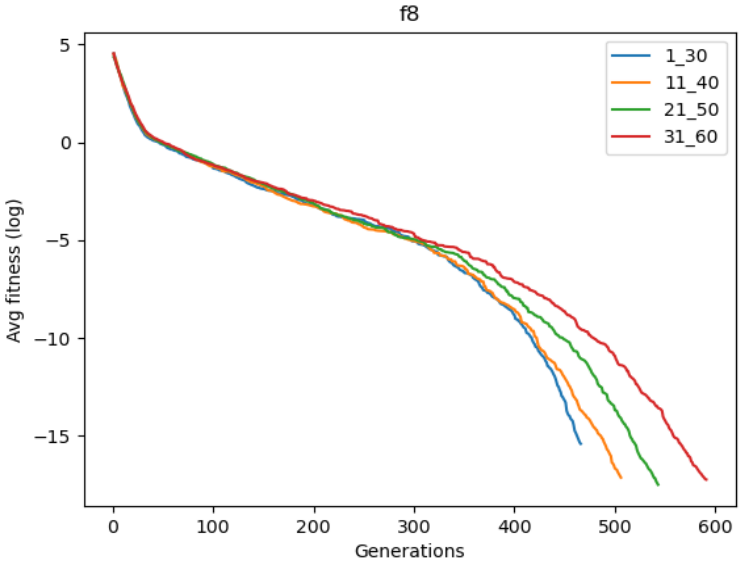


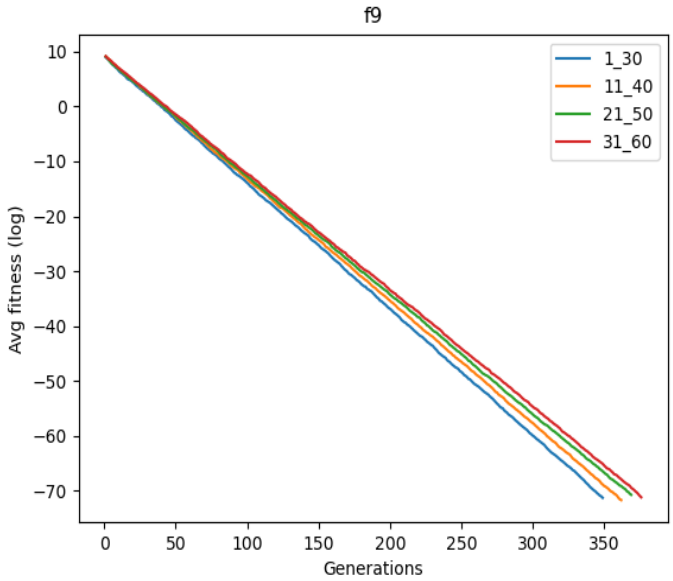
شکل ۲: نمودارهای میانگین فیتنس-نسل برای انتخاب همسایگی اعضای برتر جمعیت

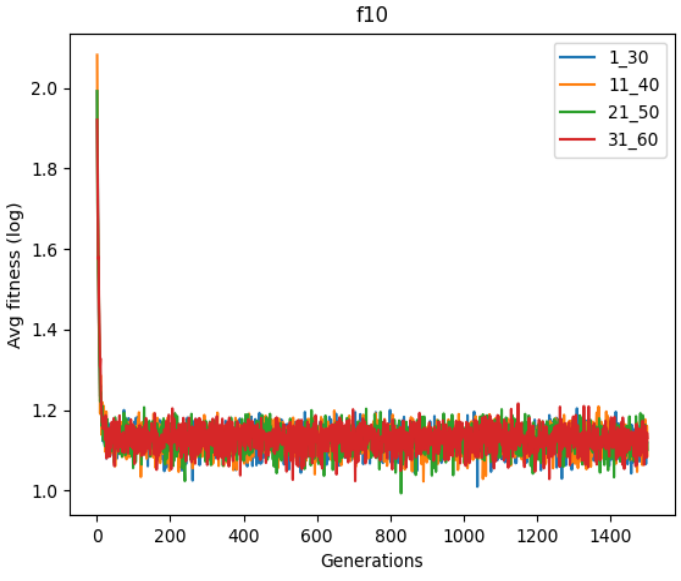




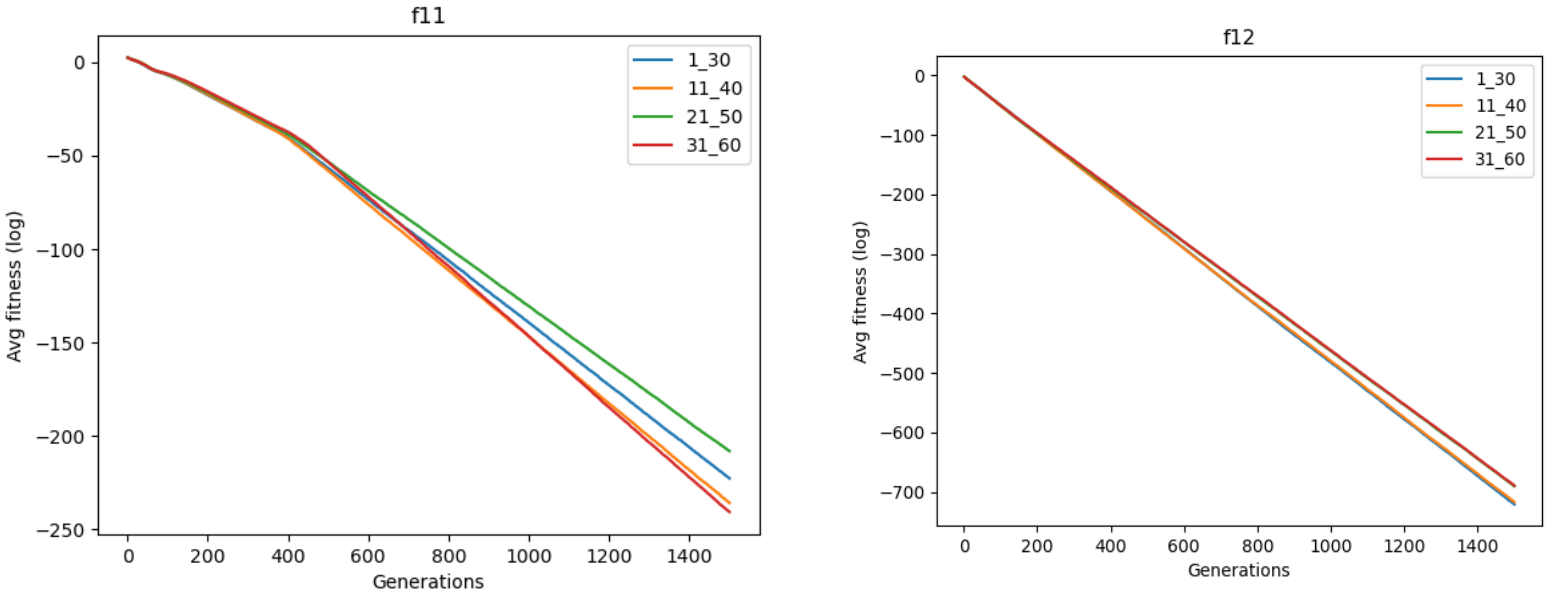








ادامه‌ی شکل ۲: نمودارهای میانگین فیتنس-نسل برای انتخاب همسایگی اعضای برتر جمعیت



ادامه‌ی شکل ۲: نمودارهای میانگین فیتنس-نسل برای انتخاب همسایگی اعضای برتر جمعیت

به جز نمودار تابع F۱۰ که یک تابع نویزی است، برای بقیه‌ی توابع می‌توان درصد مناسب همسایگی اعضای برتر برای انتخاب عضو X۳ را تشخیص داد. مثلا برای تابع F۱۱ بهتر است عضو X۳ از بین ۳۱ تا ۶۰ درصد اعضای برتر جمعیت انتخاب شود. بعد از تعیین این پارامتر، برای بقیه‌ی پارامترها باید با سعی و خطا روی پارامترهای پیشنهادی توسط مقاله تنظیم شوند. جدول ۱ پارامترهای پیشنهادی مقاله را نشان می‌دهد. جدول ۲ نیز پارامترهای پیشنهادی این پروژه است که با توجه تعداد بسیار زیاد حالات ممکن، حالاتی که در آزمون و خطاها عملکرد خوبی داشته‌اند را (بدون تست تمام حالات ممکن) آورده‌ایم.

جدول ۱: پارامترهای پیشنهادی توسط مقاله

F1, F2	0.4
Jumping Rate	0.3
Crossover Factor	0.9
Generations	1500
Population	100

جدول ۲: پارامترهای پیشنهادی توسط پروژه‌ی حاضر

Parameters / Functions	Scaling Factor 1	Scaling Factor 2	Jumping Rate	Crossover Rate	X3 Neighborhood
F1	۰.۵	۰.۲	۱	۰.۱	۱–۳۰
F2	۱	۰.۱	۱	۰.۱	۱–۳۰
F3	۰.۷	۰.۶	۱	۰.۵	۱–۳۰
F4	۰.۴	۰.۳	۱	۰.۲	۱–۳۰
F5	۰.۵	۰.۶	۱	۰.۱	۱۱–۴۰
F6	۰.۵	۰.۶	۱	۰.۱	۱–۳۰
F7	۰.۳	۰.۷	۰.۵	۰.۱	۱–۳۰
F8	۰.۳	۰.۵	۰.۵	۰.۱	۱–۳۰
F9	۰.۵	۰.۳	۱	۰.۱	۱–۳۰
F10	۰.۴	۰.۴	۱	۰.۲	No diff
F11	۰.۵	۰.۳	۱	۰.۱	۳۱–۶۰
F12	۱	۰.۲	۱	۰.۱	۱–۳۰

راهنمای اجرای کد

هم لینک گوگل کولب و هم فایل کد آورده شده است. برای اجرا، ابتدا باید پارامترها را تنظیم کنید. در اولین قسمت کد بعد از کتابخانه‌های لازم، می‌توانید هر پارامتری را مقداردهی کنید که در شکل ۴ آمده است.

```
####################================================ PARAMETERS ======
cr = 0.1                    #crossover rate
F1, F2 = 0.3 , 0.7          #scaling factors
jr = 0.5                    #jumping rate
#dimensions = [10,30,50,100,200,500,1000]
dimension = 30
pop_size = 100
generations = 1500
function_number = 1     #index == function number (12 functions, 1-12)
```

شکل ۳: کد مربوط به مقداردهی پارامترهای الگوریتم

تمام پارامترهای موجود در شکل ۴ توضیح داده شده‌اند و برای مقداردهی آنها می‌توانید به جدول ۲ مراجعه کنید. برای انتخاب تابع هدف، متغیر function_number را بین ۱ تا ۱۲ مقدار دهید که هر کدام یک تابع بنچمارک است. توابع بنچمارک در شکل ۱ معرفی شده‌اند. لطفا مقدار صفر به این متغیر ندهید چون کد ارور خواهد داد.

تابع اصلی برای اجرای کد در شکل ۵ آمده است. این تابع دو ورودی دریافت می‌کند؛ اولی مربوط به درصد اعضای برتر جمعیت است که در شکل ۳ نمودارهای آن به طور کامل آمده است و باید اولین عدد هر مقدار وارد شود که در جدول ۳ آمده است.

```
#rank = [1,11,21,31]
execute(rank = 1, n_executions = 10)
```

شکل ۴: کد مربوط به تابع اصلی اجرای الگوریتم و ورودی‌های آن

پارامتر n_executions تعیین‌کننده‌ی تعداد دفعات اجرای الگوریتم برای ارزیابی نتایج است.

جدول ۳: نحوه‌ی دریافت ورودی پارامتر rank تابع اصلی کد

مقدار پارامتر rank برای تابع execute()	درصد اعضای برتر جمعیت برای انتخاب X^3
۱	۱-۳۰
۱۱	۱۱-۴۰
۲۱	۲۱-۵۰
۳۱	۳۱-۶۰

نتایج اجرای الگوریتم

نتایج اجرای الگوریتم روی تمام توابع ارزیابی در جدول ۴ قابل مشاهده است. فایل متنی این نتایج نیز با دقت اعشاری بالاتر در فایل پروژه موجود است. ستونهای جدول ۴ شامل بهترین شایستگی، بدترین شایستگی، میانگین شایستگی‌ها، میانه‌ی شایستگی‌ها، متوسط زمان اجرا و انحراف معیار زمان اجرا بر حسب ثانیه است. علت در نظر گرفتن این فیلدها امکان مقایسه با نتایج مقاله است.

جدول ۴: نتایج اجرای الگوریتم روی ۱۲ تابع ارزیابی مختلف

F1	Best	Worst	Mean	Median	Avg(s)	Std(s)
۱۰	۴.۱۲e-۱۵۹	۱.۹۴e-۱۵۵	۴.۶۰e-۱۵۶	۱.۵۸e-۱۵۷	۴.۸۴	۰.۳۳
۳۰	۳.۴۲e-۵۱	۵.۰۹e-۴۹	۱.۹۲e-۴۹	۴.۴۵e-۵۰	۱۱.۲۷	۰.۵۵
۵۰	۲.۲۲e-۲۷	۵.۹۰e-۲۶	۱.۴۱e-۲۶	۳.۶۰e-۲۷	۱۶.۶۴	۰.۴۵
۱۰۰	۴.۴۲e-۰۹	۳.۲۸e-۰۸	۱.۶۳e-۰۸	۷.۹۳e-۰۹	۳۲.۴۸	۰.۸۱
۲۰۰	۳.۳۵	۶۱.۲۶	۱۶.۵۸	۵.۹۱	۶۷.۴۰	۴.۸۱
۵۰۰	۱۱۸۰۷.۳۶	۲۴۴۸۳.۵۶	۱۶۴۲۷.۲۳	۱۵۱۸۲.۹۳	۱۸۰.۳۵	۳.۳۸
۱۰۰۰	۱۳۴۵۵۳.۷۴	۱۶۰۹۰۲.۵۲	۱۴۷۷۵۵.۶	۱۵۰۹۴۵.۹	۳۷۹.۹۳	۲.۴۱
F2	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۷.۳۹e-۸۶	۱.۶۰e-۸۳	۱.۰۴e-۸۳	۱.۵۳e-۸۳	۱۳.۴۲	۰.۶۳
۳۰	۱.۰e-۲۷	۳.۰۶e-۲۶	۱.۱۰e-۲۶	۱.۴۲e-۲۷	۳۴.۲۲	۱.۴۴
۵۰	۳.۲۰e-۱۶	۲.۰۲e-۱۵	۱.۲۸e-۱۵	۱.۵۰e-۱۵	۵۳.۱۴	۰.۱۹
۱۰۰	۰.۰۰۰۱۴	۰.۰۰۰۶۰	۰.۰۰۰۳۵	۰.۰۰۰۳۱	۱۰۴.۳۰	۰.۲۵
۲۰۰	۴.۵۶	۶.۸۸	۵.۵۳	۵.۱۴	۲۲۶.۲۴	۳.۴۹
۵۰۰	۳۰۹.۳۵	۳۷۸.۵۹	۳۴۵.۰۱	۳۴۷.۰۹	۵۵۸.۷۹	۱.۹۹
۱۰۰۰	inf	inf	inf	inf	۱۱۰۹.۱۱	۴۲.۶۶
F3	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۷.۴۳e-۴۲	۲.۵۲e-۱۲	۸.۴۲e-۱۳	۵.۵۰e-۳۸	۶۳.۸۸	۱.۳۱
۳۰	۰.۰۰۱۰	۰.۰۴۹	۰.۰۱۸	۰.۰۰۳۹	۲۰۲.۶۱	۱.۳۰
۵۰	۳۵۱.۹۵	۶۲۷.۱۸	۵۲۶.۰۲	۵۹۸.۹۳	۳۵۶.۳۱	۰.۴۱۸
۱۰۰	۸۷۷۹۶.۰۱	۹۵۵۷۸.۴۳	۹۲۱۱۱.۲۶	۹۲۹۵۹.۳۳	۸۵۴.۷۷	۴.۵۲
۲۰۰	۳.۳۵	۶۱.۲۶	۱۶.۵۸	۵.۹۱	۶۷.۴۰	۴.۸۱
۵۰۰	۷۴۲۷۳۳.۸۷	۹۷۲۶۸۸.۹۲	۸۷۳۳۰۷.۷۲	۹۰۴۵۰۰.۳۸	۲۷۱۶.۲۸	۳۱۲.۸۷
۱۰۰۰						
F4	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۷.۴۰e-۱۸۱	۹.۲۳e-۶۸	۴.۶۰e-۱۵۶	۴.۶۵e-۱۸۰	۹.۰۴	۱.۱۷
۳۰	۱.۳۶e-۶۵	۴.۳۶e-۶۰	۱.۴۷e-۶۰	۷.۲۱e-۶۲	۱۰.۱۳	۲.۲۷
۵۰	۱.۳۴e-۳۲	۱.۸۴e-۲۴	۶.۱۴e-۲۵	۱.۰۶e-۳۰	۱۳.۵۷	۰.۰۵
۱۰۰	۱.۳۶e-۱۱	۰.۰۲۶	۰.۰۰۸	۸.۶۱e-۰۹	۲۷.۱۹	۰.۴۱
۲۰۰	۰.۰۱۹	۰.۲۰۸	۰.۱۳۴	۰.۱۷۴	۵۸.۳۱	۰.۴۷۹
۵۰۰	۶۳۸.۹۹	۱۰۲۷.۵۳	۸۳۲.۳۳	۸۳۰.۴۶	۱۵۸.۱۹	۰.۸۶۸

۱۰۰۰	۱۰۹۸۹.۳۱	۱۶۸۳۶.۷۱	۱۳۳۳۳.۸۹	۱۲۱۷۵.۶۴	۳۲۴.۱۰	۳.۷۵۳
F5	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۲.۹۵e-۱۵	۳.۹۸	۱.۳۲	۱.۶۱e-۱۴	۸.۹۸	۰.۲۱
۳۰	۱۰.۶۲	۲۸.۵۹	۱۷.۴۸	۱۳.۲۱	۲۲.۸۸	۰.۶۵
۵۰	۳۸.۳۴	۳۹.۲۵	۳۸.۸۴	۳۸.۹۴	۳۶.۰۱	۰.۳۹
۱۰۰	۹۲.۹۴	۱۴۷.۲۹	۱۱۱.۳۲	۹۳.۷۱	۷۲.۴۱	۰.۳۲
۲۰۰	۲۸۴.۰۴	۳۱۸.۴۲	۲۹۹.۱۷	۲۹۵.۰۷	۱۴۹.۷۳	۱.۵۳
۵۰۰	۵۲۴۳.۸۳	۱۲۵۶۱.۸۸	۸۲۷۶.۲۲	۷۰۲۲.۹۴۳	۳۹۴.۶۹	۱.۵۵
۱۰۰۰	۱۰۱۲۹۹۳.۰۸	۱۴۱۲۸۰۰.۵۰	۱۱۹۸۰۵۷.۳۱	۱۱۶۸۳۷۸.۳۶	۸۰۸.۰۹۱	۲.۰۵
F6	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۰	۰	۰	۰	۸.۹۴	۰.۶۷
۳۰	۰	۰	۰	۰	۱۹.۷۸	۰.۱۷
۵۰	۵۴.۲۶	۵۷.۷۰	۵۵.۸۲	۵۵.۴۹	۳۲.۰۷	۰.۳۸
۱۰۰	۳۷۰.۹۹	۳۸۸.۱۹	۳۷۷.۴۱	۳۷۳.۰۳	۶۵.۷۷	۰.۸۸
۲۰۰	۱۱۹۹.۲۶	۱۲۲۸.۱۴	۱۲۱۰.۷۳	۱۲۰۴.۷۹	۱۳۵.۶۸	۱.۰۹
۵۰۰	۴۳۵۶.۵۱	۴۵۰۲.۴۸	۴۴۰۹.۵۱	۴۳۶۹.۵۳۳	۳۴۳.۳۱	۰.۷۵۷
۱۰۰۰	۱۰۴۲۶.۳۰۹	۱۰۶۲۷.۷۶	۱۰۵۵۵.۴۹	۱۰۶۱۲.۴۲	۶۹۱.۴۹	۲.۷۳
F7	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۳.۹۹e-۱۵	۳.۹۹e-۱۵	۳.۹۹e-۱۵	۳.۹۹e-۱۵	۷.۹۰۱	۰.۹۵
۳۰	۲.۲۴e-۱۲	۴.۷۵e-۱۲	۳.۴۲e-۱۲	۳.۴۲e-۱۲	۱۶.۷۶	۰.۰۵
۵۰	۱.۱۱e-۰۷	۴.۲۴e-۰۷	۲.۷۸e-۰۷	۲.۵۷e-۰۷	۲۶.۶۵۴	۰.۶۲
۱۰۰	۰.۰۰۲۶	۰.۰۰۳۷	۰.۰۰۳۱	۰.۰۰۳۱	۵۲.۷۵	۰.۶۰۶
۲۰۰	۱.۴۹	۱.۶۸	۱.۵۸	۱.۵۸	۱۰۶.۶۴	۱.۱۶
۵۰۰	۸.۷۱	۹.۳۷	۹.۰۹	۹.۰۷	۲۶۹.۰۵	۱.۸۵۸
۱۰۰۰	۱۴.۳۲	۱۵.۲۲	۱۴.۶۸	۱۴.۶۵	۵۳۰.۵۴	۴.۳۸
F8	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۰	۰.۰۱۲۳	۰.۰۰۴	۰	۷.۲۶	۰.۲۱
۳۰	۰	۰	۰	۰	۱۷.۸۹	۰.۷۰
۵۰	۰	۰	۰	۰	۲۸.۶۴	۰.۴۵
۱۰۰	۱.۰۰۳e-۱۲	۳.۱۶e-۱۲	۲.۰۶۷e-۱۲	۲.۰۳e-۱۲	۵۷.۸۰	۱.۱۳
۲۰۰	۰.۱۵۱	۰.۳۹۶	۰.۲۳۹	۰.۱۷۲	۱۱۶.۲۱	۱.۲۹

۵۰۰	۷.۳۶	۹.۶۰	۸.۲۸	۷.۸۶	۳۰۸.۴۰	۲.۰۱
۱۰۰۰	۱۷۸.۰۸	۲۰۸.۲۹۹	۱۹۱.۵۹۴	۱۸۸.۳۹	۶۳۶.۵۰	۵.۷۴
F9	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۰	۰	۰	۰	۳.۱۸	۰.۰۲
۳۰	۰	۳.۰۸e-۳۳	۱.۰۲e-۳۳	۰	۷.۶۰	۰.۰۱۷
۵۰	۳.۸۱e-۲۸	۶.۷۴e-۲۷	۲.۸۲e-۲۷	۱.۳۴e-۲۷	۱۲.۲۴	۰.۰۶۲
۱۰۰	۶.۲۱e-۱۰	۲.۱۰e-۰۹	۱.۳۱e-۰۹	۱.۲۲e-۰۹	۲۴.۸۱	۰.۴۸۷
۲۰۰	۰.۵۰۰	۲.۸۲	۱.۷۲۱	۱.۸۴	۵۲.۸۴	۰.۴۱
۵۰۰	۱۱۷۴۷.۱۹۰	۱۶۴۶۴.۶۴	۱۴۲۸۹.۰۵	۱۴۶۵۵.۳۴	۱۴۹.۴۷	۰.۳۷
۱۰۰۰	۱۲۴۷۷۴.۹	۱۳۳۳۳۴.۹	۱۲۸۴۱۰.۸۰۲	۱۲۷۱۲۲.۵	۳۱۵.۱۰۰	۰.۵۶۸
F10	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۲.۲۴	۳.۲۴	۲.۵۸	۲.۲۷	۵.۳۱	۰.۰۳
۳۰	۱۱.۵۴	۱۲.۹۰	۱۲.۱۹	۱۲.۱۲	۱۳.۳۵	۰.۰۶
۵۰	۱۸.۸۷	۲۰.۶۳	۱۹.۸۴	۲۰.۰۳	۲۲.۰۸	۰.۳۶۷
۱۰۰	۴۳.۹۱	۴۵.۸۹	۴۴.۷۹	۴۴.۵۸	۴۴.۷۳۸	۰.۴۹
۲۰۰	۹۴.۳۲	۹۷.۴۲	۹۶.۲۵	۹۷.۰۰	۹۳.۳۱	۰.۵۶
۵۰۰	۳۹۱.۰۵	۴۵۲.۵۷	۴۱۵.۵۵	۴۰۳.۰۳	۲۴۷.۱۶	۰.۳۷
۱۰۰۰	۲۸۱۲.۴۳	۳۴۲۳.۴۷	۳۱۳۸.۶۹	۳۱۸۰.۱۶	۵۱۶.۸۲۰	۱.۱۹
F11	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۱.۷۶e-۱۲۲	۳.۷۳e-۹۳	۱.۲۴e-۹۳	۱.۱۷e-۱۰۶	۳۱.۰۶	۵.۲۱
۳۰	۸.۴۶e-۳۲	۶.۱۰e-۱۶	۲.۰۳e-۱۶	۵.۳۷e-۳۰	۶۷.۸۱	۰.۶۷
۵۰	۶.۸۳e-۱۶	۱.۷۲e-۱۵	۱.۰۳e-۱۵	۷.۰۴e-۱۶	۱۱۱.۷۱	۰.۵۵
۱۰۰	۳.۹۹e-۰۷	۲.۷۵e-۰۶	۱.۴۶e-۰۶	۱.۲۲e-۰۶	۲۲۳.۵۸	۰.۸۰۷
۲۰۰	۰.۰۱۵	۰.۱۱۰	۰.۰۵۷	۰.۰۴۵۸	۴۴۹.۷۹	۳.۸۵
۵۰۰	۲۳.۴۷	۳۰.۳۷	۲۶.۲۷	۲۴.۹۶	۱۱۲۷.۲۸۵	۱.۷۴۲
۱۰۰۰	۲۴۷.۷۹	۲۴۹.۰۰	۲۴۸.۲۱	۲۴۷.۸۴	۲۲۸۰.۴۷	۰.۶۴
F12	**Best**	**Worst**	**Mean**	**Median**	**Avg(s)**	Std(s)
۱۰	۳.۰۷e-۳۲۰	۲.۰۸e-۳۰۱	۲.۰۸e-۳۰۲	۳.۹۱e-۳۱۴	۱۲.۲۱	۰.۳۹
۳۰	۲.۰۷e-۱۱۳	۴.۴۰e-۹۰	۴.۴۰e-۹۱	۳.۴۴e-۹۸	۳۳.۱۵	۰.۹۰
۵۰	۲.۰۹e-۵۰	۸.۰۰e-۲۰	۸.۱۳e-۲۱	۱.۳۷e-۲۶	۵۴.۹۳	۱.۲۵
۱۰۰	۳.۵۵e-۱۸	۲.۳۹e-۱۰	۴.۷۶e-۱۱	۷.۶۱e-۱۳	۱۱۷.۷۶	۹.۵۲

۲۰۰	۲.۲۲e-۱۳	۱.۰۶e-۰۷	۱.۰۷e-۰۸	۲.۷۳e-۱۱	۲۲۳.۵۸	۴.۲۷
۵۰۰	۸.۹۳e-۱۴	۴.۱۴e-۰۹	۴.۵۴e-۱۰	۱.۰۹e-۱۱	۵۹۶.۵۷۶	۲۲.۶۵

تحلیل نتایج

به طور کلی با ارزیابی این الگوریتم دریافتیم در ابعاد بالا (۲۰۰، ۵۰۰ و ۱۰۰۰) تعداد جمعیت ۱۰۰ و تعداد نسل ۱۵۰۰ فرصت کافی برای بهینه شدن الگوریتم به دست نمی‌دهد و نتایج مقاله کمی عجیب است چون با ادامه دادن بهینه‌سازی برای بیشتر از ۱۵۰۰ نسل می‌توان پاسخهای بهتری گرفت.

نکته‌ی مهم دیگر آن که متاسفانه در هیچکدام از توابع بنچمارک موفق نشدیم به ازای پارامترهای پیشنهاد شده‌ی مقاله که در جدول ۱ نیز آمده است، نتایج خوبی کسب کنیم. اگر نویسندگان مقاله به ازای هر تابع از مجموعه پارامترهای متفاوتی بهره برده‌اند بهتر بود در متن مقاله ذکر می‌کردند.

توابع Multi Modal

الگوریتم مذکور در بهینه‌سازی تابع چندمداله‌ی Rosenbrock تا بعد ۱۰۰ نسبت به مقاله بهتر عمل کرده است. در تابع Rastrigin تا بعد ۵۰ کاملا بهتر عمل کرده است اما در ابعاد سه رقمی به همان علت کافی نبودن تعداد نسل نتوانسته مانند مقاله جواب بگیرد؛ ما تعداد نسل و جمعیت را دقیقا مثل مقاله در نظر گرفتیم تا امکان مقایسه‌ی دقت‌ها فراهم باشد. در تابع Ackley در هیچ بعدی جوابی بهتر از مقاله نگرفتیم اما جوابها در ابعاد پایین به جوابهای مقاله نزدیک است و حتی در بعد ۱۰ انحراف معیار بهترین جوابها صفر است. در تابع Griewank فقط در ابعاد دو رقمی موفق شدیم پاسخ بهینه‌ی سراسری(صفر) را بیابیم اما در ابعاد بالاتر گویا باید جستجو ادامه می‌یافته است.

توابع Uni Modal

در تابع F1 مقاله در تمام ابعاد جواب صفر را یافته است اما در این پروژه در بعد ۱۰ جوابی بسیار کوچک یافتیم و هر چه ابعاد بالاتر می‌رود جوابها از صفر بیشتر فاصله می‌گیرند. در باقی توابع تک حالته نیز جوابها تقریبا به همین صورت هستند؛ مجددا بازگو می‌کنیم که علت فاصله داشتن جوابها خصوصا در توابع تک‌حالته به خاطر تعداد نسل کم در مقابل ابعاد بزرگ مثل ۱۰۰۰ یا ۵۰۰ است و جوابهایی که مقاله گزارش کرده است، عجیب به نظر می‌رسند.

جمع‌بندی

در این الگوریتم یک روش یادگیری Opposition برای افزایش Diversity و بردار جهشی نوین برای افزایش فشار انتخاب معرفی شده‌اند تا از هر دو جهت الگوریتم بتواند خوب عمل کند. این الگوریتم بر خلاف ادعاهای مقاله موفق شد در بهینه‌سازی توابع چندحالته در ابعاد پایین خوب عمل کند اما در ابعاد بالا نیاز به نسل بیشتری برای رسیدن به جوابهایی نزدیک آنچه مقاله گزارش داده دارد. در توابع تک‌حالته نتوانستیم دقتهایی به خوبی مقاله بیابیم. انتقادی که به روش کار مقاله‌ی مذکور وارد است این است که ابتدا توابع ارزیابی را اجرا نموده سپس به دنبال یافتن پارامتر مربوط به همسایگی برای X^3 رفته در حالی که بهتر بود ابتدا تمام پارامترها تنظیم می‌شدند و سپس خروجی نهایی گرفته می‌شد. نقد دیگری که می‌توان به این مقاله وارد آورد این است که به ازای پارامترهای پیشنهادی آن هرگز نمی‌توان به جوابهایی مناسب رسید و باید دید هدف نویسندگان محترم از ارائه‌ی چنین مقادیری برای پارامترهای اصلی این الگوریتم چه بوده است.

مقاله

Deng, Wu, Shifan Shang, Xing Cai, Huimin Zhao, Yingjie Song, and Junjie Xu. "An improved differential evolution algorithm and its application in optimization problem." *Soft Computing* 25, no. 7 (2021): 5277-5298.