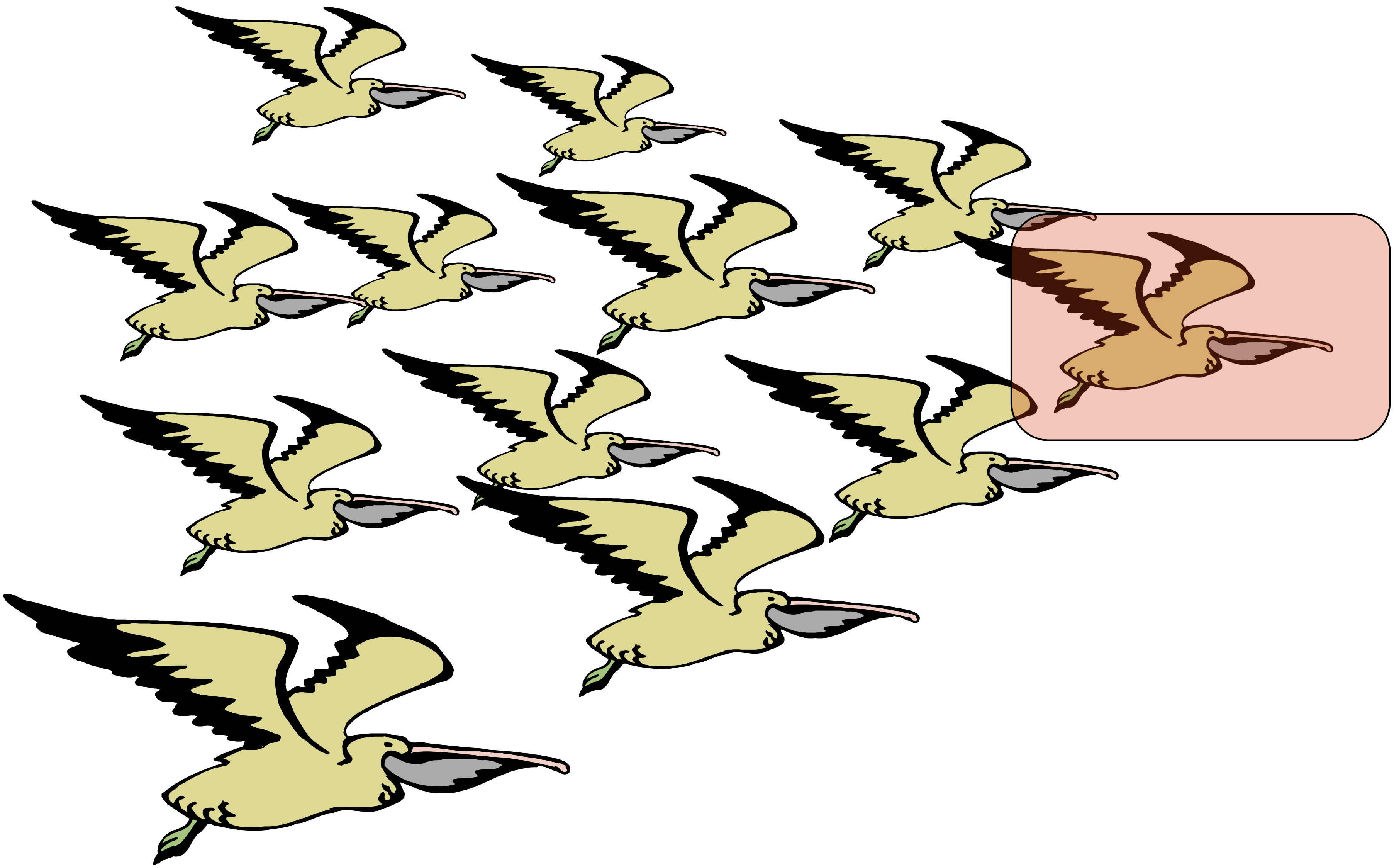
الگوریتم توده ذرات



مقدمه

این الگوریتم اولین بار توسط Kennedy و Eberhart معرفی شد. آنها این الگوریتم را PSO نام نهادند زیرا از روی رفتار گروهی پرندگان در زمان پرواز الهام گرفته شده بود.

همانند سایر الگوریتم‌های جمعیتی، الگوریتم PSO از مجموعه‌ای از پاسخ‌های ممکن استفاده می‌کند که این پاسخ‌ها تا زمانی که یک پاسخ بهینه یافت شده و یا شرایط پایان الگوریتم مهیا شود به حرکت خود ادامه می‌دهند.

مقدمه

در این روش هر پاسخ X به صورت یک **ذره (Particle)** نمایش داده می‌شود و یک **گروه ذرات (Swarm)** در حقیقت یک مجموعه از ذرات است.

در این روش، معادله سرعت ضامن حرکت ذرات به سمت ناحیه بهینه می‌باشد. این معادله معمولا بر اساس سه عنصر اصلی ارائه می‌شود که عبارتند از:

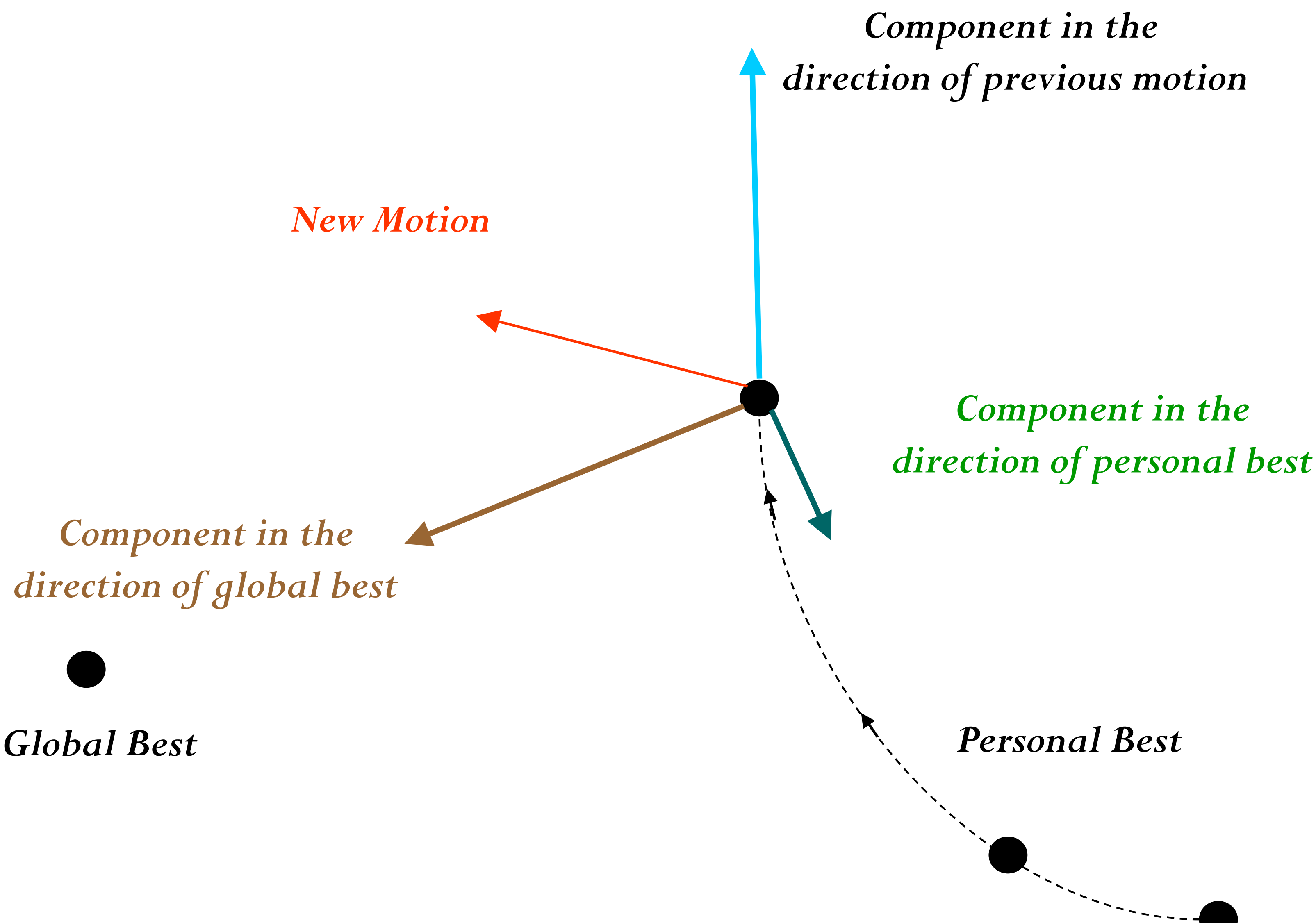
- **سرعت سکون** (Velocity Inertia)
- **مولفه شناختی pbest** (Cognitive Component)
- **مولفه جمعی gbest** (Social Component)

مقدمه

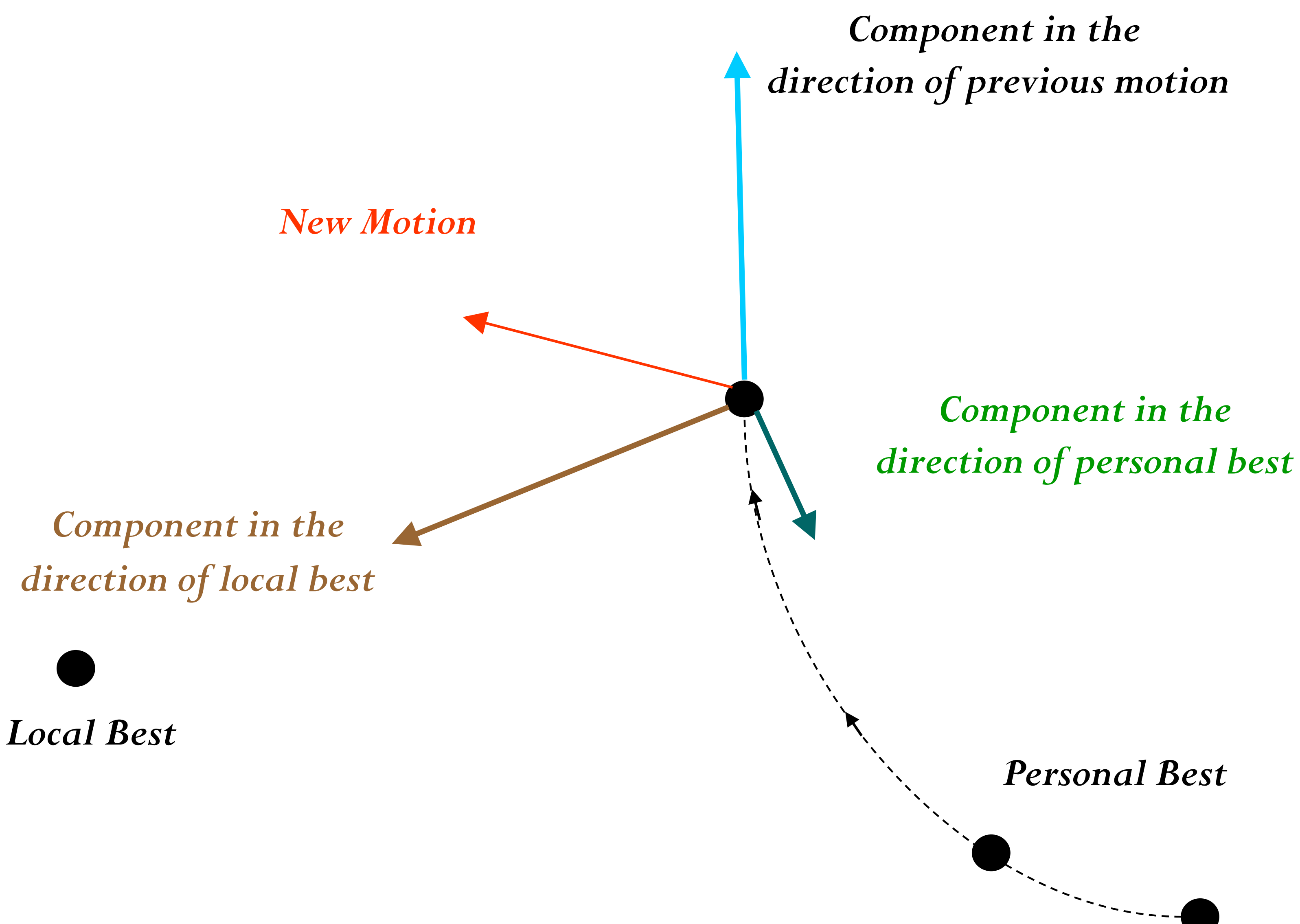
رهیافت نهایی می‌تواند به عنوان الگوریتمی‌شناخته شود که جستجویی را به صورت چند بعدی اعمال می‌کند.

در شبیه سازی این الگوریتم، رفتار هر ذره می‌تواند تحت تاثیر بهترین ذره محلی (در داخل یک همسایگی مشخص) و یا بهترین ذره عمومی باشد.

الگوریتم توده ذرات



الگوریتم توده ذرات



ویژگی

خصوصیت جالب PSO این می‌باشد که این الگوریتم به ذرات اجازه می‌دهد تا از بهترین تجربه گذشته خویش بهره برداری نمایند (یاد آوری می‌شود که در سایر روش‌ها همانند الگوریتم ژنتیک معمولا جمعیت فعلی تنها حافظه‌ای است که توسط پاسخ‌ها مورد استفاده قرار می‌گیرد).

قابل ذکر است که الگوریتم PSO تا کنون برای مسایل غیر خطی پیوسته و نیز مسایل **گسسته دودویی** برای بهینه سازی تک هدفه و چند هدفه مورد استفاده قرار گرفته است.

الگوریتم

الگوریتم PSO دارای پارامترهای زیر می‌باشد:

- **معیار خاتمه:** این معیار ضوابط اتخاذ شده برای به پایان رساندن اجرای الگوریتم را در بر دارد ولی معمولا به تعداد دفعات تکراری گفته می‌شود که الگوریتم اجرا خواهد شد.
- **تعداد ذرات:** این معیار به تعداد کل ذراتی که در فضای جستجو حرکت می‌کنند اشاره دارد.

الگوریتم

- **W**: به مقدار سرعت سکون ذره اشاره دارد.
- **C_1**: مقدار ثابت پارامتر **شناختی (Cognitive)** است. این مقدار ثابت دلالت بر میزان شدت تاثیر پذیری از بهترین موقعیت شخصی دارد.
- **C_2**: ترکیب کننده **جمعی (Social)** می‌باشد. این مقدار دلالت بر میزان تاثیر موقعیت بهترین ذره‌ای که تا کنون یافت شده بر روی ذره فعلی دارد.

$$W > \frac{1}{2}(c_1 + c_2) - 1$$

الگوریتم

به طور کلی اگر $\vec{x}_i(t)$ نشان دهنده موقعیت ذره P_i در فضای جستجو در لحظه t باشد، موقعیت P_i با افزودن سرعت $\vec{v}_i(t)$ به موقعیت فعلی به صورت زیر تغییر می‌نماید:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t)$$

$$\vec{v}_i(t) = \vec{v}_i(t-1) + c_1 r_1 (\vec{x}_{pbest} - \vec{x}_i(t-1)) + c_2 r_2 (\vec{x}_{gbest} - \vec{x}_i(t-1))$$

الگوریتم

که در آن $\vec{v}_i(t)$ بردار سرعت در گام t-ام، c_1 و c_2 مقادیر ثابت مثبت r_1 و r_2 اعدادی تصادفی هستند که به صورت نرمال در بازه [۰ ،۱] تولید می‌شوند.

پارامترهای $\vec{x}_{pbest}$ و $\vec{x}_{gbest}$ به ترتیب نشان دهنده موقعیت بهترین تجربه شخصی و جمعی می‌باشند.

به منظور ایجاد قابلیت بهتر جستجو، پارامتری به نام وزن اینرسی به شکل زیر و به صورت ضریبی در پارامتر سرعت به الگوریتم اضافه می‌شود:

$$\vec{v}_i(t) = w\vec{v}_i(t-1) + c_1 r_1(\vec{x}_{pbest} - \vec{x}_i(t-1)) + c_2 r_2(\vec{x}_{gbest} - \vec{x}_i(t-1))$$

الگوریتم

وزن اینرسی تاثیر سرعت ذرات در گام قبل را بر سرعت فعلی تعیین می‌نماید. به این ترتیب که با مقادیر بزرگی از وزن اینرسی قابلیت جستجوی عمومی‌الگوریتم بهبود یافته و فضای بیشتری مورد بررسی قرار می‌گیرد.

از سوی دیگر با مقادیر کوچک وزن اینرسی، فضای مورد بررسی محدود شده و جستجو در این فضای محدود شده صورت می‌گیرد.

الگوریتم

از همین رو به طور معمول الگوریتم با مقدار بزرگی از وزن اینرسی شروع به حرکت می‌کند که سبب جستجوی گسترده فضا در ابتدای اجرا شده و این وزن به مرور در طول زمان کاهش می‌یابد که سبب تمرکز جستجو در فضای کوچک در گام‌های پایانی می‌شود.

الگوریتم

روش‌های گوناگونی به منظور کنترل پارامتر وزن اینرسی معرفی شده است از جمله:

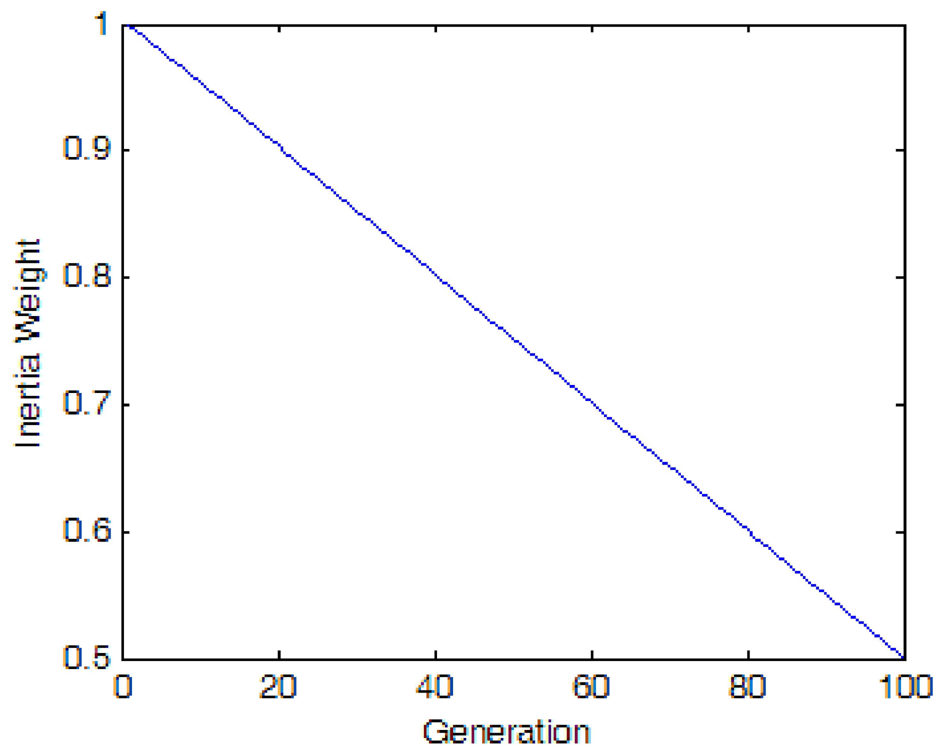
- انتخاب تصادفی
- کاهش خطی
- کنترل فازی
- ....

الگوریتم

در الگوریتم پایه PSO از روش زیر به منظور تولید این وزن استفاده می‌شود که

در آن t شماره گام فعلی و T تعداد کل گام‌های الگوریتم است.

$$w = 1 - 0.5\left(\frac{1-t}{1-T}\right)$$



الگوریتم

به منظور تولید وزن‌ها با کاهش خطی از روش زیر نیز می‌توان استفاده نمود که در آن از یک وزن اولیه بزرگ (حدودا ۰.۹) به یک وزن کوچک (حدودا ۰.۴) خواهیم رسید:

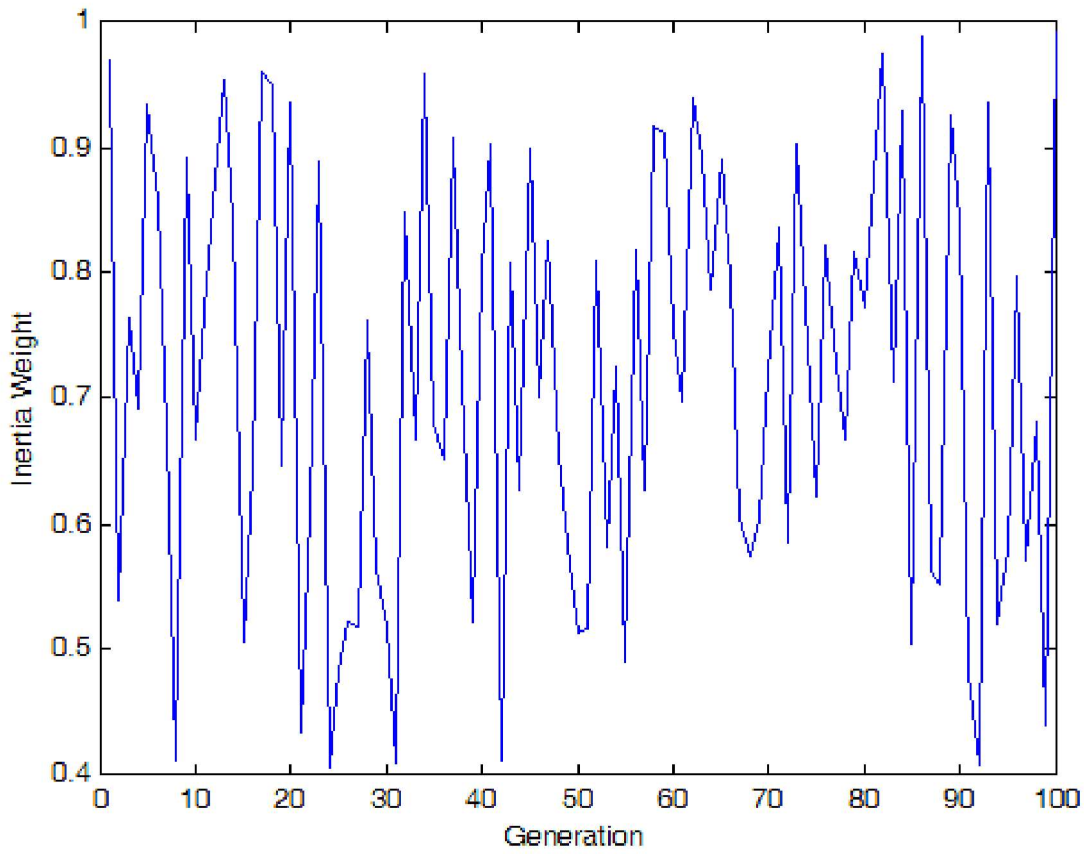
$$w_t = (w_1 - w_T)\frac{(T-t)}{T} + w_T$$

که در آن T حداکثر تعداد دفعات اجرای الگوریتم است، w_1 وزن اینرسی اولیه، w_T وزن اینرسی نهایی و w_t وزن اینرسی در گام t است. قابل ذکر است که $w_1 > w_T$ در نظر گرفته می‌شود.

الگوریتم

روش دیگر استفاده از تولید وزن‌های تصادفی به صورت زیر است:

$$w = w_0 + rand(1 - w_0)$$



الگوریتم

به منظور تولید وزن‌های تصادفی از روش زیر نیز می‌توان استفاده نمود:

$$w \sim N(0.72, \sigma)$$

در این روش در هر تکرار مقدار وزن اینرسی به صورت تصادفی از تابعی با توزیع گوسی انتخاب می‌شود. به این نکته باید توجه کرد که مقدار σ به نحوی انتخاب می‌شود که سبب تولید وزن‌های بزرگتر از ۱ نشود.

یک رویکرد دیگر هم استفاده از رابطه زیر است:

$$w = c_1 r_1 + c_2 r_2$$

الگوریتم

در ابتدا، ذرات به صورت تصادفی در سرتاسر فضای جستجو مقداردهی می‌شوند که این موقعیت‌های اولیه به عنوان بهترین تجربه شخصی ذرات نیز شناخته می‌شوند (pbest).

در گام بعد بهترین ذره از میان ذرات موجود انتخاب شده و به نام بهترین پاسخ شناخته می‌شود (gbest).

سپس گروه ذرات در فضای جستجو حرکت می‌نمایند تا زمانی که شرایط پایان محقق شود. این حرکت شامل اعمال معادله سرعت به گروه ذرات می‌باشد که موقعیت هر ذره بر اساس آن تغییر می‌کند.

الگوریتم

در الگوریتم توده ذرات موارد زیر در هر چرخه می باید به روزرسانی شوند:

- موقعیت بهترین تجربه شخصی : $\vec{X}_{pbest}$
- مقدار بهترین تجربه شخصی
- موقعیت بهترین تجربه عمومی : $\vec{X}_{gbest}$
- مقدار بهترین تجربه عمومی

الگوریتم

مقدار تطابق جدید حاصل از ذره با مقدار pbest همان ذره مقایسه می‌شود. در حالتی که موقعیت جدید دارای تطابق بهتری باشد، این موقعیت جدید جایگزین موقعیت pbest شده و مقدار آن نیز به روز می‌شود.

رویه مشابهی نیز برای به روز رسانی مقدار و موقعیت gbest انجام می‌پذیرد.

الگوریتم

همچنین رویکرد زیر نیز قابل ارائه می‌باشد:

$$c_1(t) = (c_{1,\min} - c_{1,\max})\frac{t}{T} + c_{1,\max}$$

$$c_2(t) = (c_{2,\max} - c_{2,\min})\frac{t}{T} + c_{2,\min}$$

که در روابط فوق داریم:

$$c_{1,\max} = c_{2,\max} = 2.5$$

$$c_{1,\min} = c_{2,\min} = 0.5$$

الگوریتم

در مورد حل مشکل مقادیر سرعت ایجاد شده در خارج از باند سرعت مجاز، به این ترتیب عمل می‌شود که هر کدام از اجزا بردار وزن که محدودیت حداکثر سرعت مجاز را رعایت نکنند با جایگزینی از مقادیر بازه تولید متغیر x اصلاح می‌شوند.

if $V_{ij}(t) < X_L$ then $V_{ij}(t) = X_L + r$

if $V_{ij}(t) > X_U$ then $V_{ij}(t) = X_U - r$

$r \in U[0,1]$

Algotithm gbest PSO (Initialize)

gbest = A (A is a large positive number for Minimization and a large minus number for Maximization Problems)

for i=1 to $N_{particles}$ **do**

Initialize randomly X_i

$Xpbest_i = X_i$

$V_i = X_i$ Or $V_i = 0$

$fitness_i = f(X_i)$

$pbest_i = fitness_i$

if $fitness_i < gbest$ **then**

$Xgbest = X_i$

$gbest = fitness_i$

end if

end for

الگوریتم

Algotithm gbest PSO (Main loop)

repeat

for i=1 to $N_{particles}$ **do**

$V_i = WV_i + c_1 r_1 (X_{pbest\ i} - X_i) + c_2 r_2 (X_{gbest} - X_i)$

if $V_i \notin V_{admissible}$ **then**

correct V_i

end if

$X_i = X_i + V_i$

$fitness_i = f(X_i)$

if $fitness_i < pbest_i$ **then**

$X_{pbest\ i} = X_i$ and $pbest_i = fitness_i$

end if

if $fitness_i < gbest$ **then**

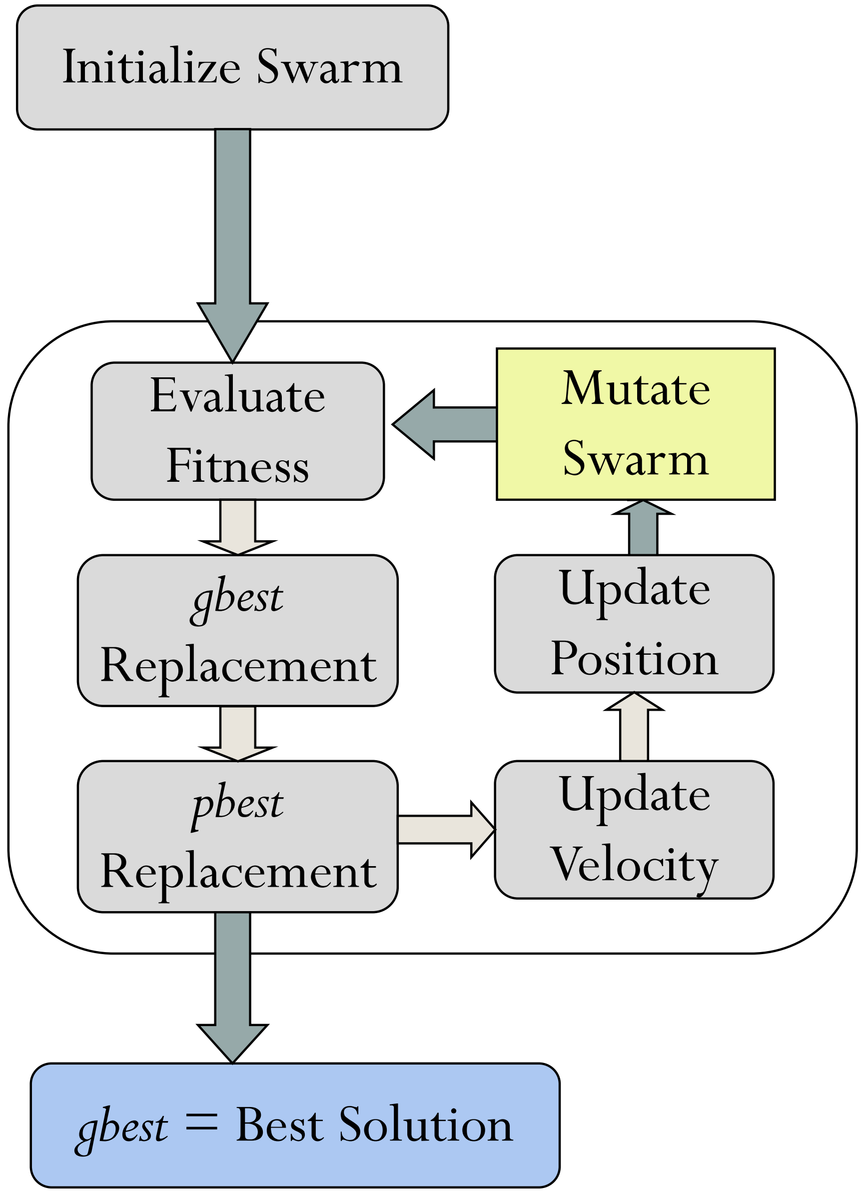
$X_{gbest} = X_i$ and $gbest = fitness_i$

end if

end for

until Termination criteria

Synchronous PSO Algorithm with Mutation

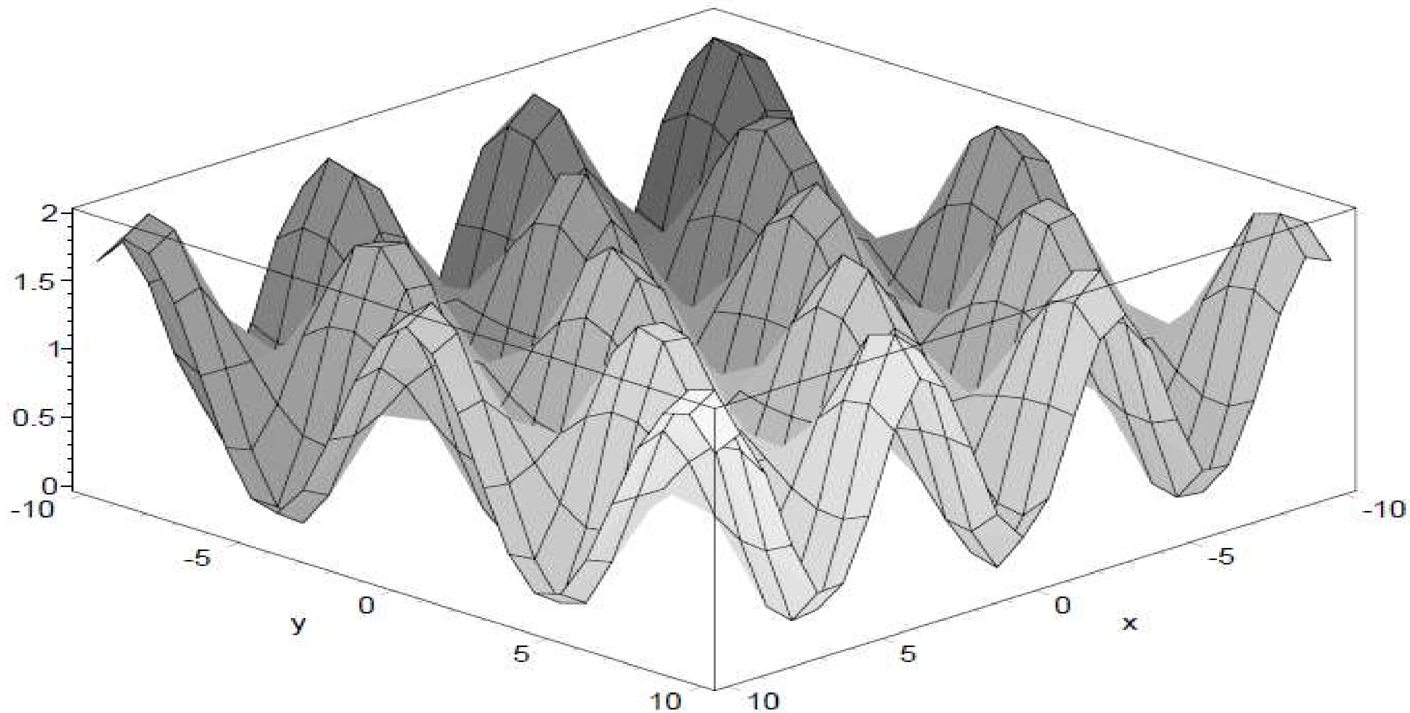


توابع ارزیابی

Griewank (f_1) :

$$f_1 = 1/4000 \sum_{i=1}^{n} x_i^{\,2} - \prod_{i=1}^{n} \cos(x_i / \sqrt{i}) + 1$$

$$\vec{x} \in [-300, 300]^n \qquad , \quad \min\ f_1(\vec{x}^*) = f_1(\vec{0}) = 0$$

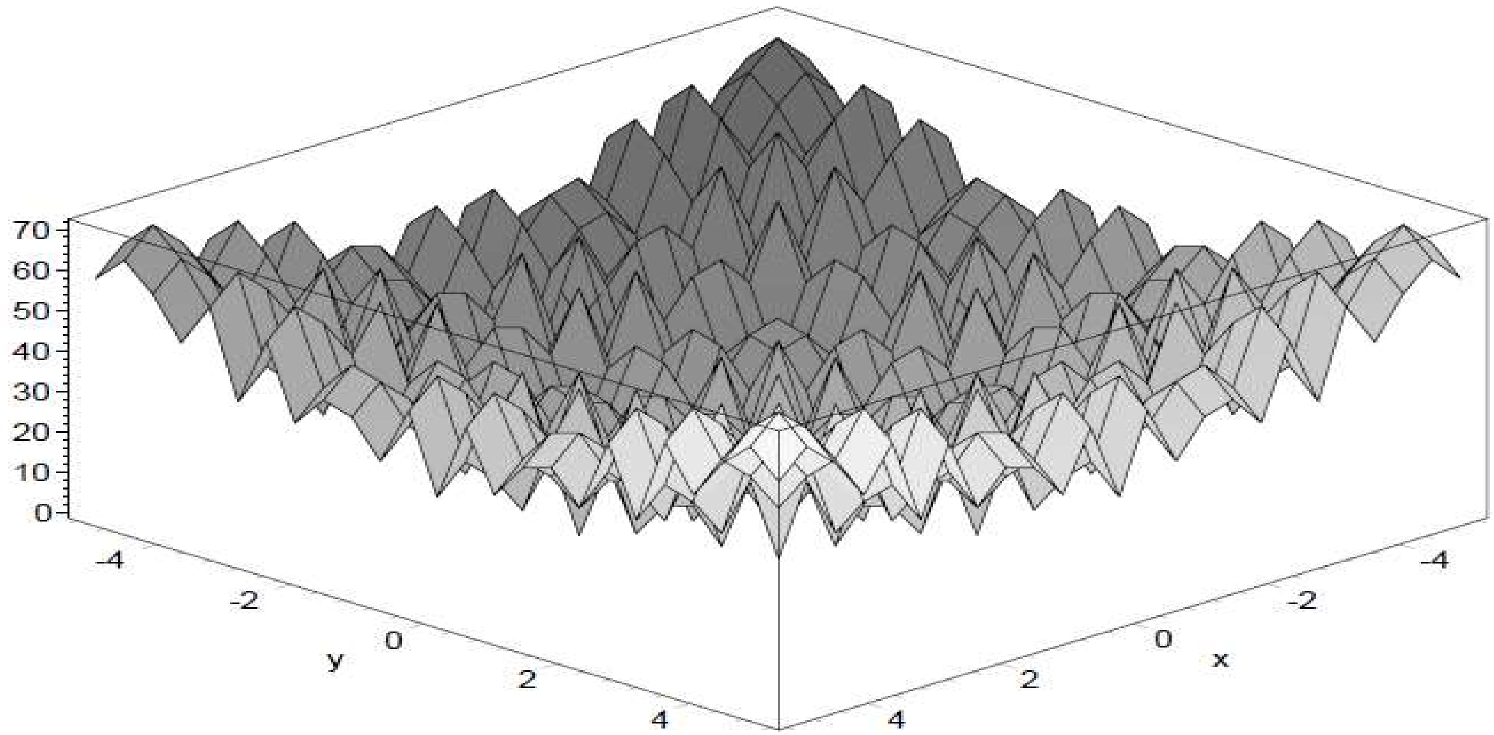


توابع ارزیابی

Rastrigin (f_2):

$$f_2 = \sum_{i=1}^{n} ({x_i}^2 - 10\cos(2\pi x_i) + 10)$$

$$\vec{x} \in [-5.12, 5.12]^n \qquad , \quad \min \ f_2(\vec{x}^*) = f_2(\vec{0}) = 0$$

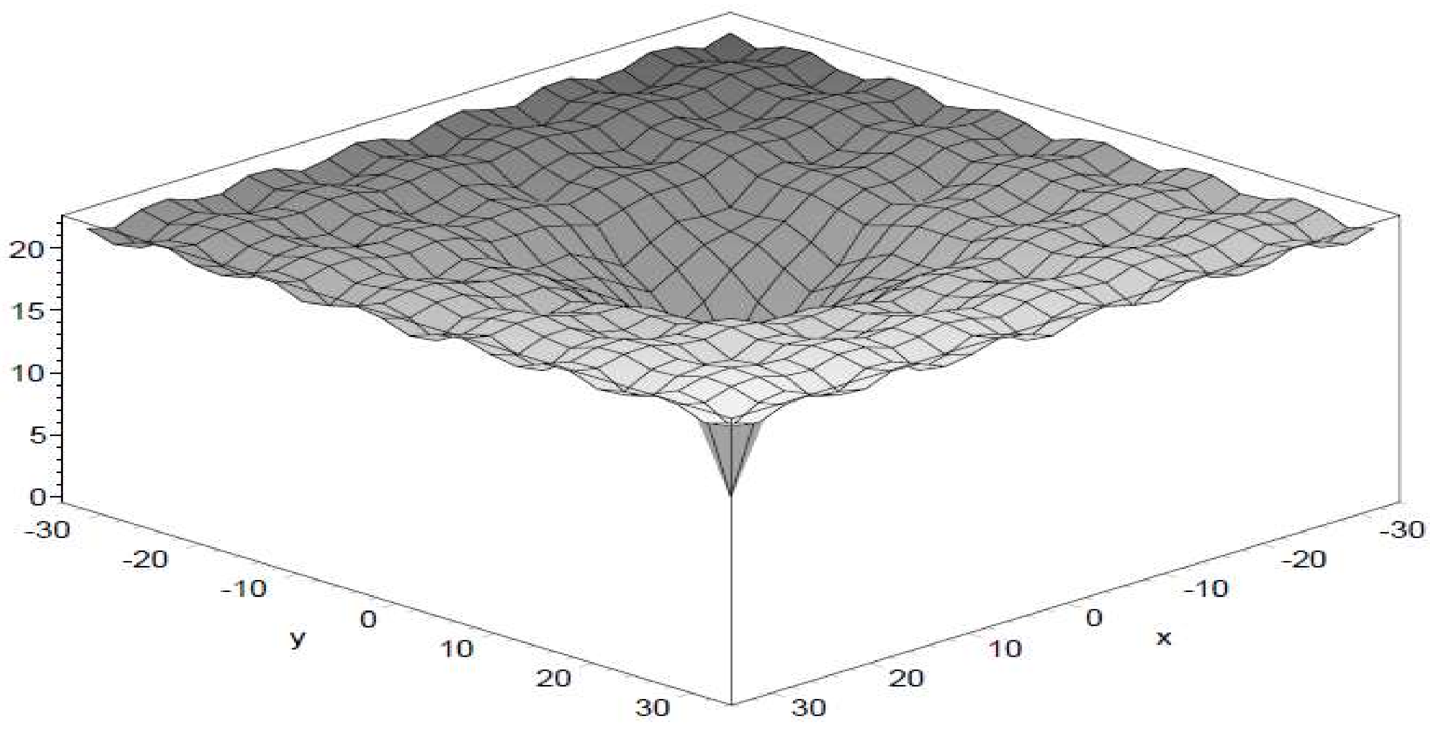


توابع ارزیابی

Ackley (f_3):

$$f_3 = -20\exp\left(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^{2}}\right) - \exp\left(1/n\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$$

$$\vec{x} \in [-32, 32]^n \quad , \quad \min\ f_3(\vec{x}^*) = f_3(\vec{0}) = 0$$

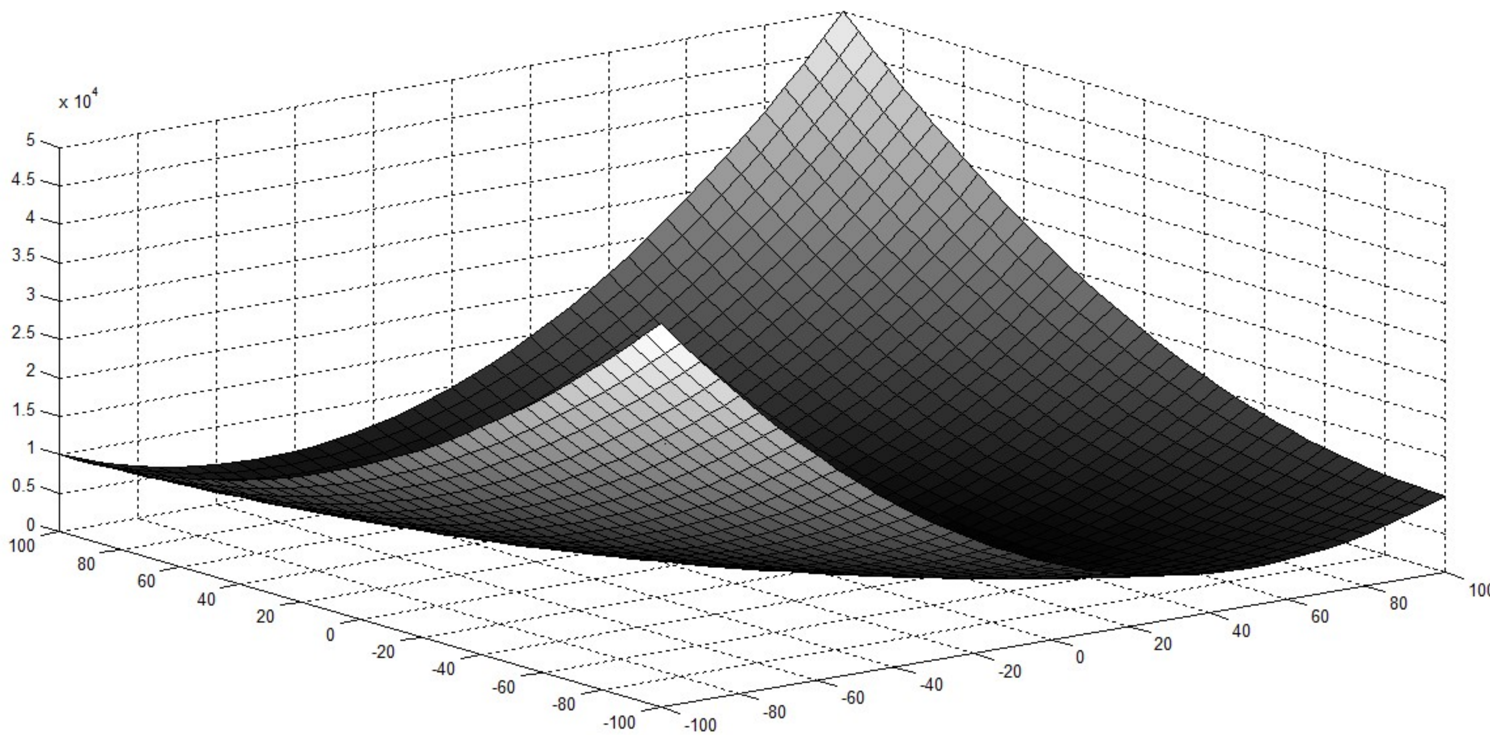


توابع ارزیابی

Quadric (f_4) :

$$f_4 = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$$

$$\vec{x} \in [-100, 100]^n \quad , \quad \min\ f_4(\vec{x}^*) = f_4(\vec{0}) = 0$$

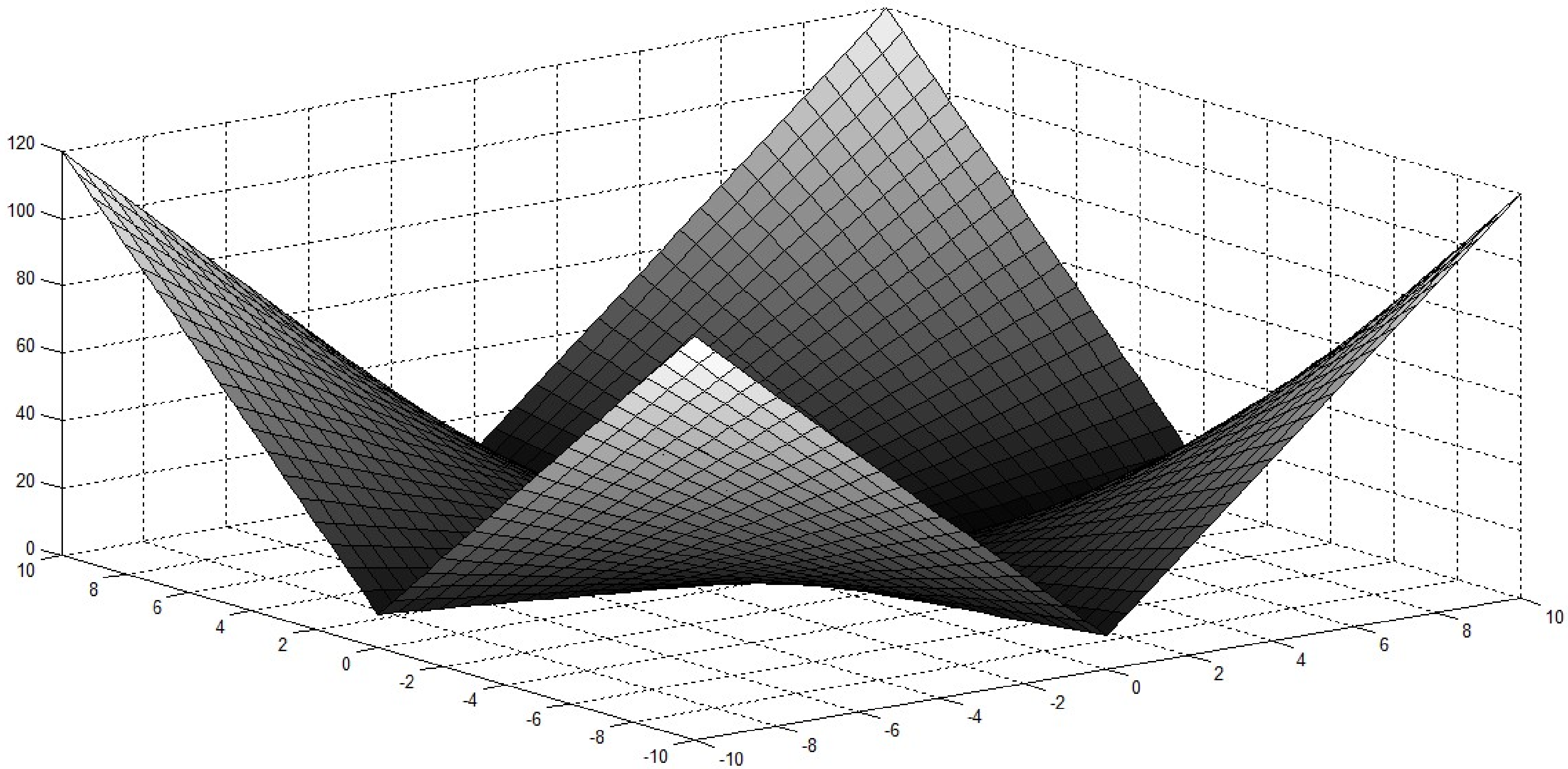


توابع ارزیابی

Schwefel 2.22 (f_5):

$$f_5 = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

$$\vec{x} \in [-10, 10]^n \qquad , \qquad \min \ f_5(\vec{x}^*) = f_5(\vec{0}) = 0$$

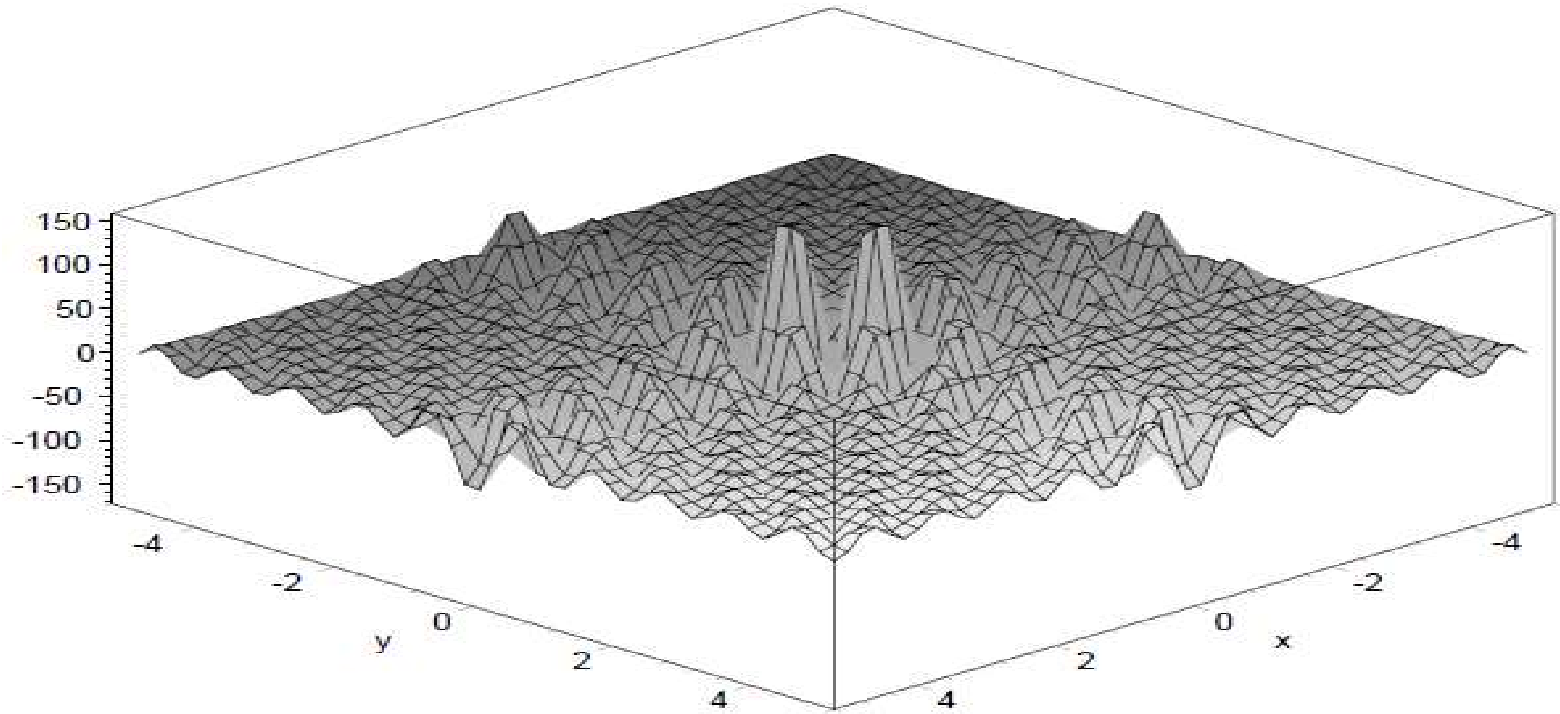


توابع ارزیابی

Shubert (f_6) :

$$f_6 = \prod_{i=1}^{n} \sum_{j=1}^{5} (j \cos((j+1)x_i + j))$$

$\vec{x} \in [-10, 10]^n$, min $f_6(\vec{x}^*)$ is unknown

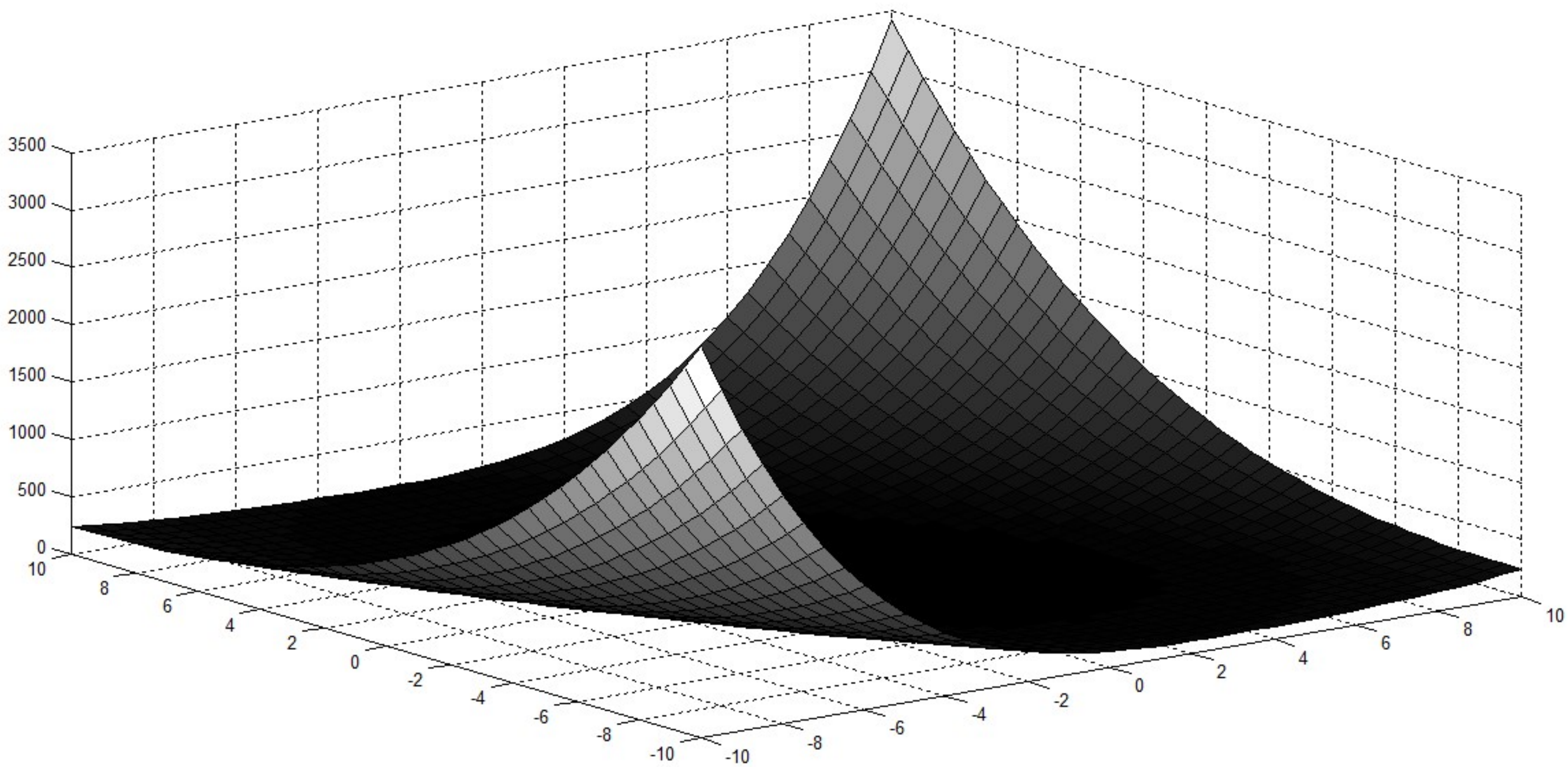


توابع ارزیابی

Zakharov (f_7) :

$$f_7 = \sum_i^n x_i^2 + (\sum_i^n 1/2ix_i)^2 + (\sum_i^n 1/2ix_i)^4$$

$$\vec{x} \in [-10, 10]^n \quad , \quad \min\ f_7(\vec{x}^*) = f_7(\vec{0}) = 0$$



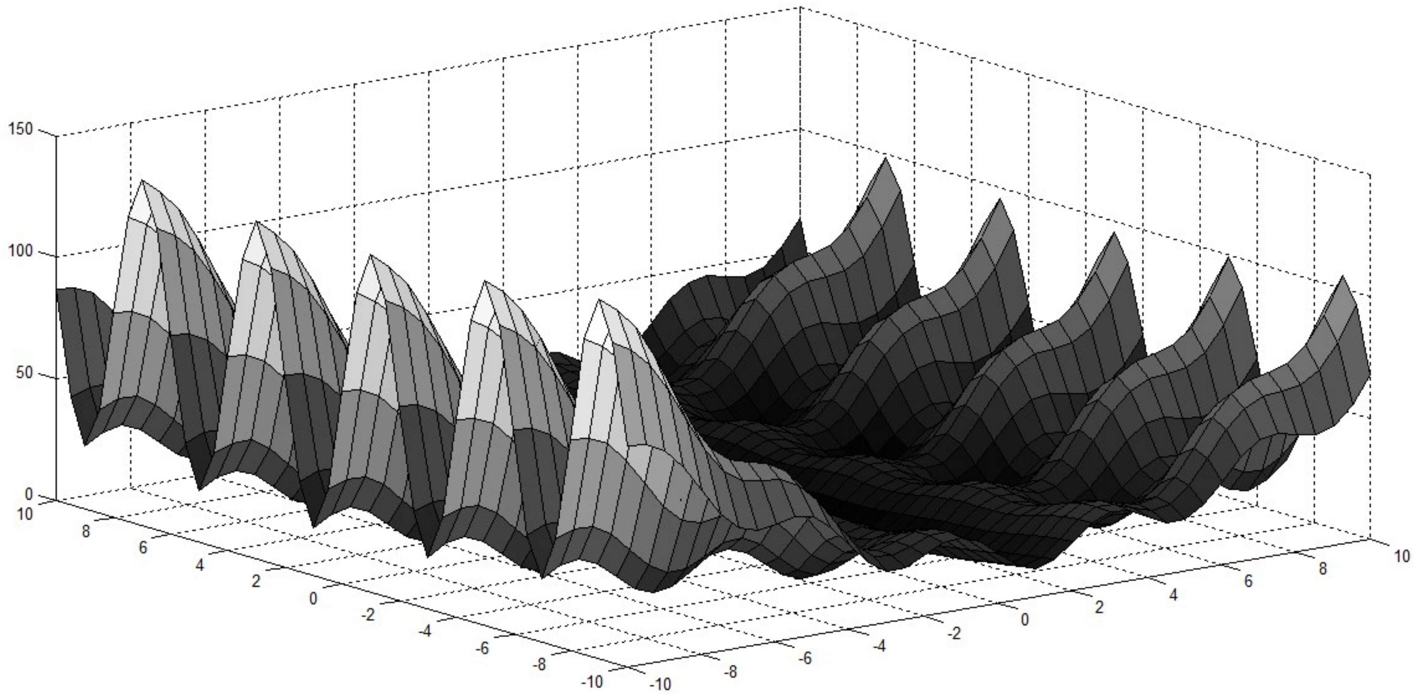
توابع ارزیابی

Levy (f_8) :

$$f_8 = \frac{\pi}{n}\left(k\sin^2(\pi y_1) + \sum_{i=1}^{n-1}((y_i - a)^2 \times (1 + k\sin^2(\pi y_{i+1}))) + (y_n - a)^2\right)$$

$$y_i = 1 + 1/4(x_i - 1), \qquad k = 10, \quad a = 1$$

$$\vec{x} \in [-10, 10]^n \quad , \quad \min\ f_8(\vec{x}^*) = f_8(\vec{1}) = 0$$

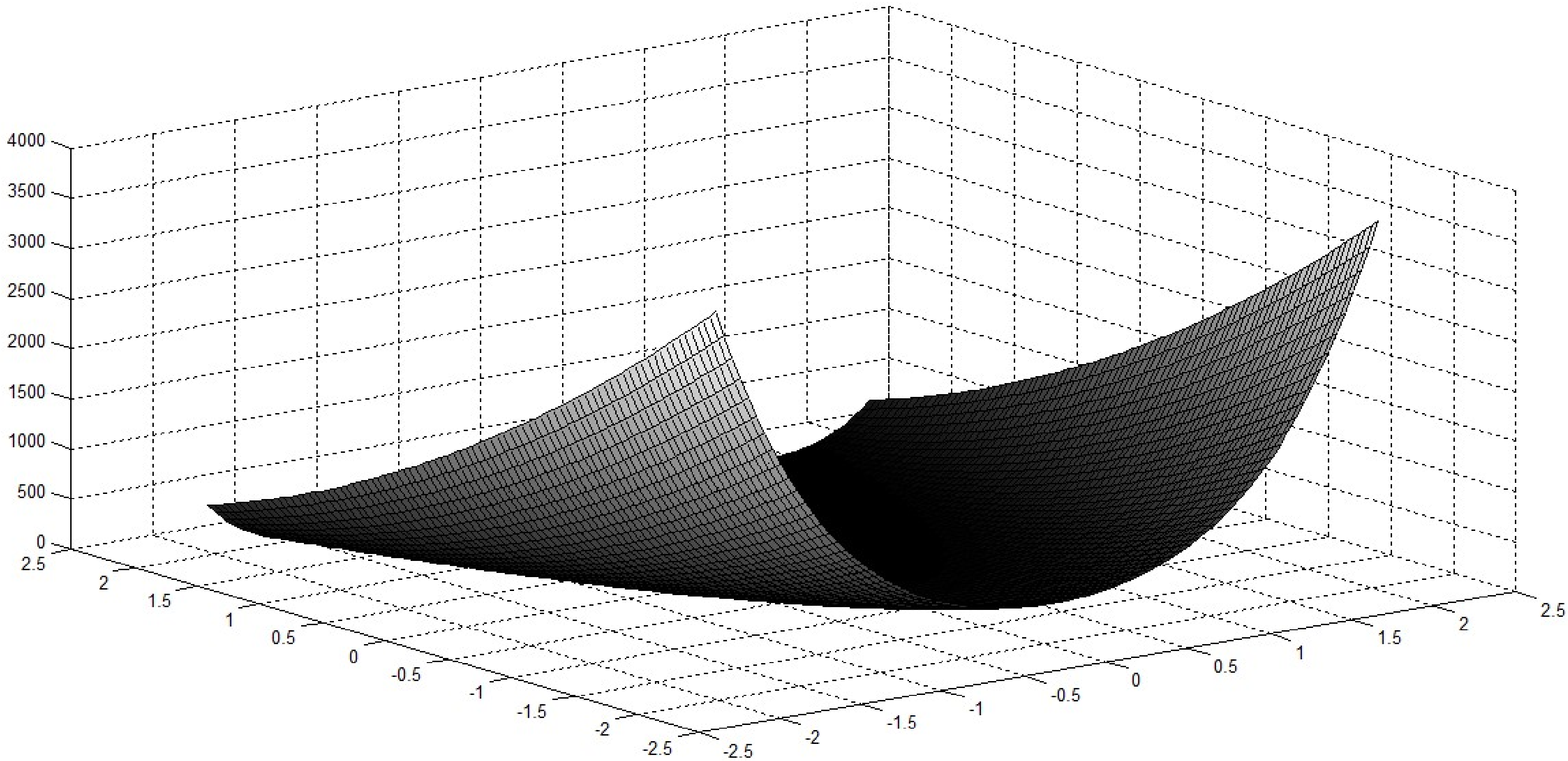


توابع ارزیابی

Rosenbrock (f_9) :

$$f_9 = \sum_i^n (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$$

$$\vec{x} \in [-2.048, 2.048]^n \quad , \quad \min \ f_9(\vec{x}^*) = f_9(\vec{1}) = 0$$

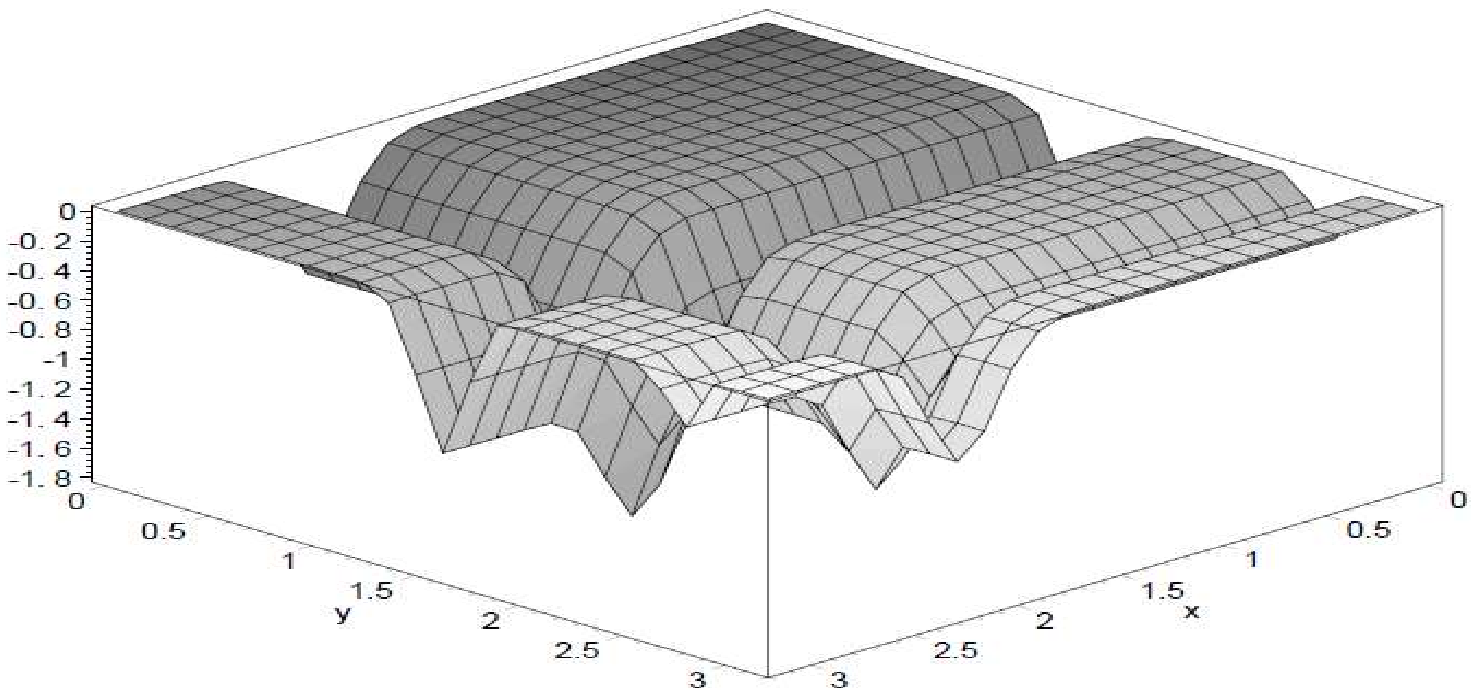


توابع ارزیابی

Michalewiz (f_{10}) :

$$f_{10} = -\sum_{i=1}^{n} \sin(xi) \cdot \left(\sin\left(\frac{i \cdot x_i^2}{\pi} \right) \right)^{2m} \quad , \quad m = 10$$

$$\vec{x} \in [0, \pi]^n \qquad , \qquad \min\ f_{10}(\vec{x}^*)\ \text{is unknown}$$



توابع ارزیابی

Schwefel 2.26 (f_{11}):

$$f_5 = 418.9829n - \sum_{i=1}^{n}\left(x_i \sin(\sqrt{|x_i|})\right)$$

$$\vec{x} \in [-500, 500]^n \quad , \quad \min\ f_{11}(\vec{x}^*) = f_{11}(420.\vec{9}687) = 0$$