GSA
Gravitational search algorithm

Esmat Rashedi
Electrical Eng. Dept.,
KGUT, Kerman, Iran

مقدمه:

۱- روش های کلاسیک ⬅ در بعضی مسایل جوابگو نمی باشند.

۲- الگوریتم های جستجوی هیوریستیک (ابتکاری) ⬅ با الهام از فرآیند های فیزیکی یا رفتارهای موجودات زنده.

- الگوریتم بهینه سازی گرانشی با الهام از قانون جاذبه و مفهوم جرم شکل یافته است.
- عامل های جستجوگر (agents) ، مجموعه ای از اجسام (Objects) می باشد.
- با توجه به محدوده وسیع گرانش در علم فیزیک، منبع الهام بزرگی برای بهبود الگوریتم وجود دارد. (گرانش نیوتن، نسبیت، کوانتوم، حرکات ستاره ها و سیارات، ...)

نیروی گرانش در طبیعت

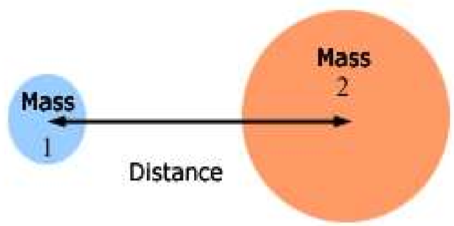
قانون گرانش

$$F = G\frac{M_1 \times M_2}{R^2}$$

G ثابت گرانشی

M جرم ذره

R فاصله بین ذرات

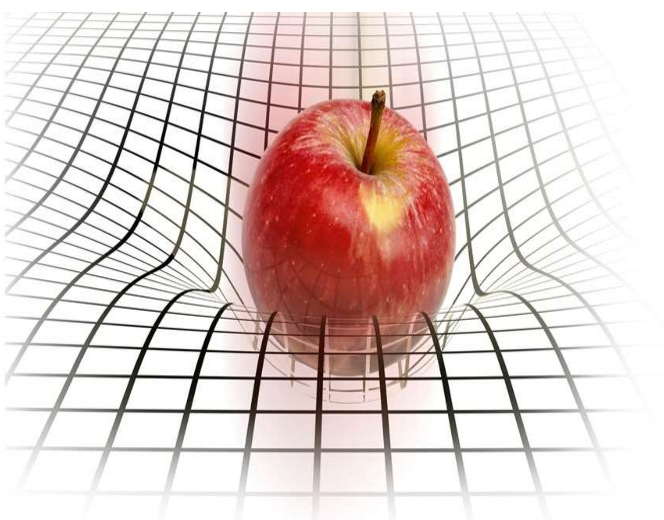


قانون دوم نیوتن

$$a = \frac{F}{M}$$

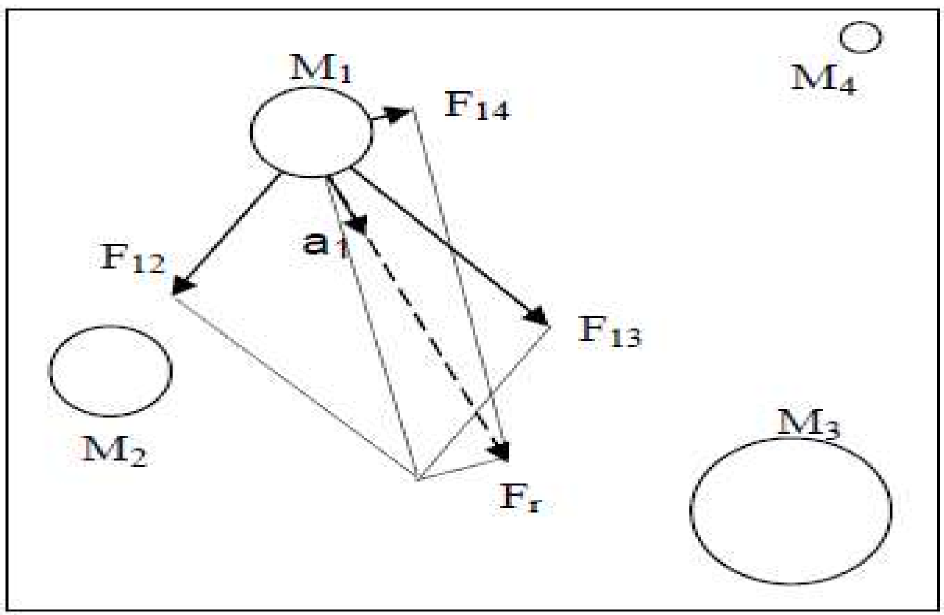
نیروی گرانش در طبیعت

- هر جسم به واسطه نیروی جاذبه، محل و جرم سایر اجسام را درک می کند.
- هر جسم به نسبت میزان جرمش و فاصله ای که با دیگر اجسام دارد، به آنها نیرو وارد می کند.
- نیروی جاذبه گرانشی بین تمام ذرات وجود دارد.
- اثر جرم بزرگتر و نزدیکتر بیشتر است.



نیروی گرانش در طبیعت

در یک سیستم با چند جسم، از جانب سایر اجسام نیروهای گرانشی وارد می شود. در نتیجه جسم به سمت برایند این نیروها که با Fr نشان داده می شود شتاب می گیرد.



مفهوم جرم در طبیعت

- جرم گرانشی فعال(Active gravitational mass): معیاری از شدت نیروی گرانشی حول یک جسم است.
- جرم گرانشی غیر فعال (Passive gravitational mass): نشان دهنده قدرت اثر متقابل در میدان گرانشی میباشد.
- جرم اینرسی (inertia mass) : معیاری از مقاومت شی در برابر تغییر موقعیت مکانی وحرکت است.
- در فیزیک مقدار این سه جرم برای یک جسم با یکدیگر برابرند.

نیروی گرانش در طبیعت

بازنویسی قوانین نیوتن:

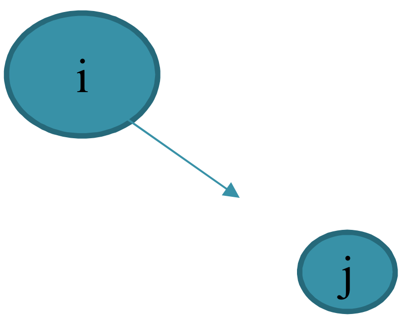
$$F_{ij} = G\frac{M_{aj} \times M_{pi}}{R^2}$$

$$a_i = \frac{F_{ij}}{M_{ii}}$$

Maj جرم گرانشی فعال جسم j

Mpi جرم گرانشی غیر فعال جسم i

Mii جرم اینرسی جسم i



نیروی گرانش در طبیعت

ثابت گرانش G

ثابت گرانشی G در طبیعت در حال کاهش است.

$$G(t) = G(t_0) \times \left(\frac{t_0}{t}\right)^{\beta}, \quad \beta < 1,$$

GSA

الگوریتم جستجوی گرانشی

- عامل ها (agent) را به عنوان اجسام (object) درنظر می گیریم.
- اجسام بهتر سنگینتر هستند و جرم بیشتری دارند.
- همه اجسام یکدیگر را توسط نیروی جاذبه جذب می کنند.
- اجرام سنگین تر متناظر جواب های بهتر هستند.
- سنگین ترین جرم، جواب شبه بهینه در فضای جستجو است.

GSA

خصوصیات هر جسم در GSA

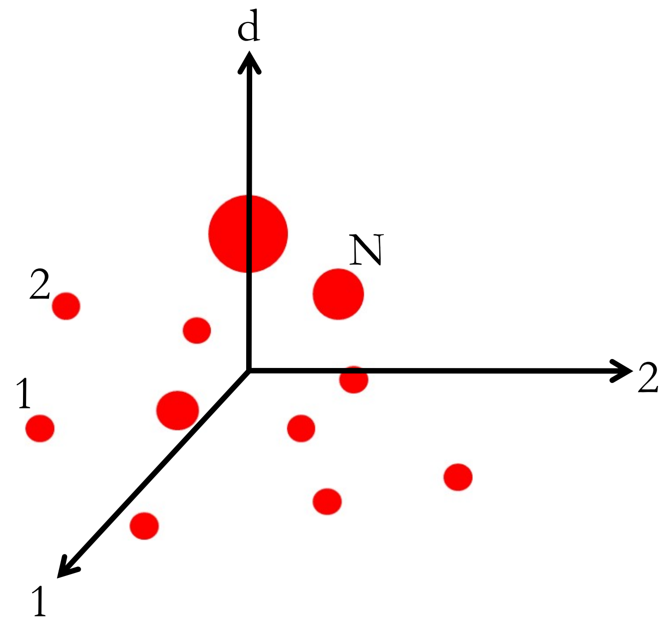
Mp **Ma** **Mi** **Position**

A solution for the problem

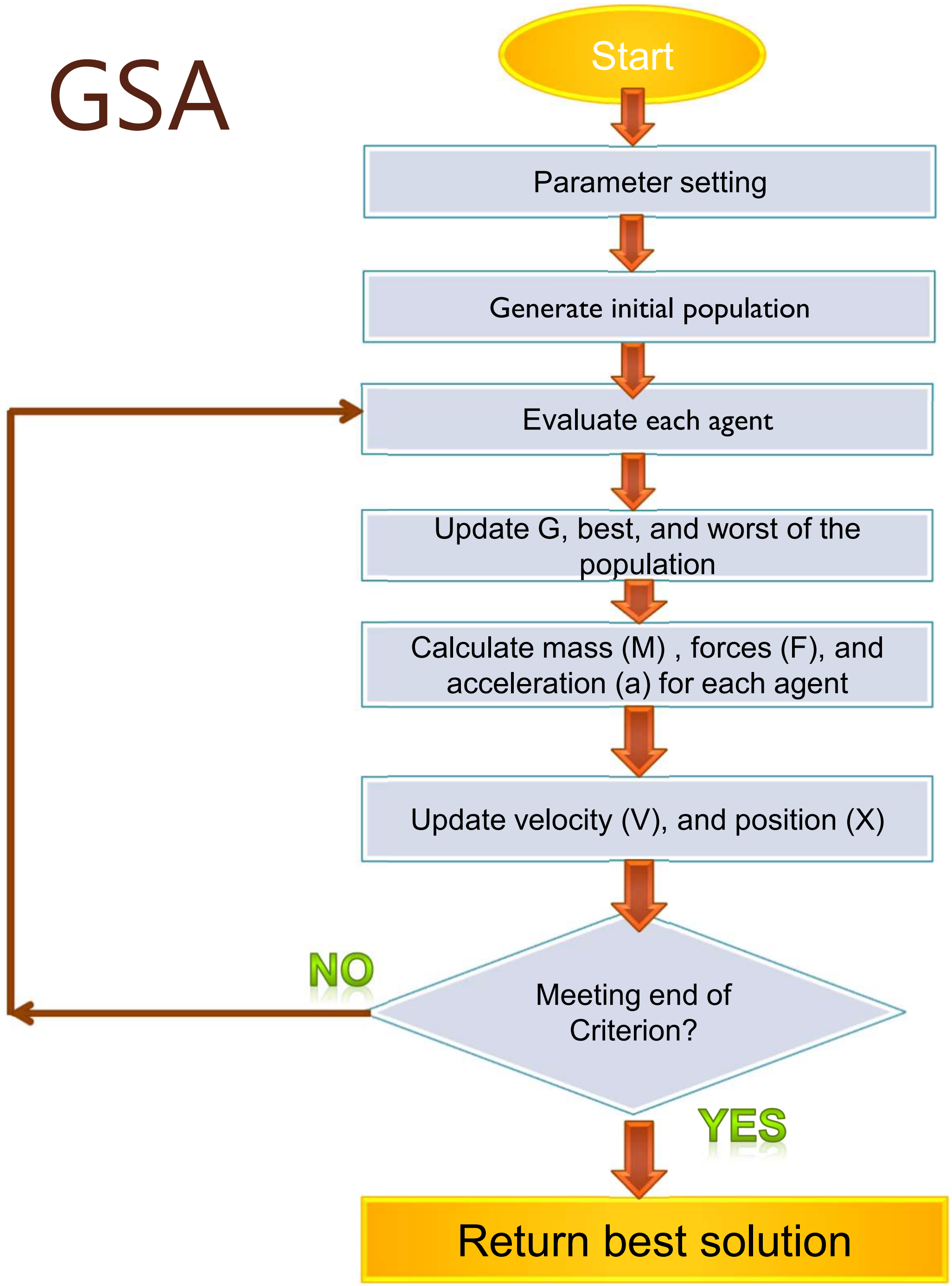
جرم های اینرسی، گرانشی فعال و گرانشی غیر فعال توسط تابع هدف (objective function) تعیین می شوند.

Gravitational Search Algorithm (GSA)

- Law of gravity
- Law of motion



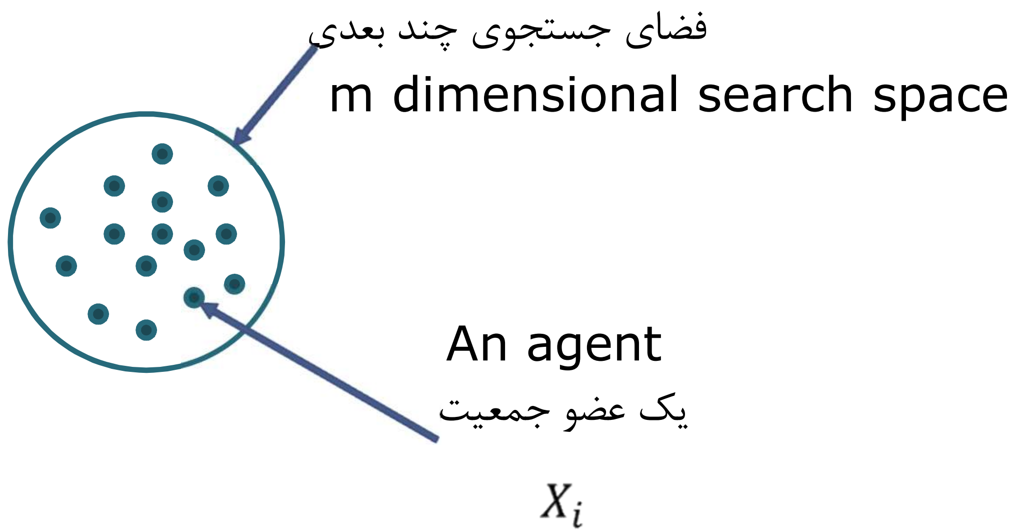
GSA


Start
Parameter setting
Generate initial population
Evaluate each agent
Update G, best, and worst of the population
Calculate mass (M) , forces (F), and acceleration (a) for each agent
Update velocity (V), and position (X)
Meeting end of Criterion?
NO
YES
Return best solution

ویژگی های الگوریتم های جستجوی (بهینه سازی) ابتکاری

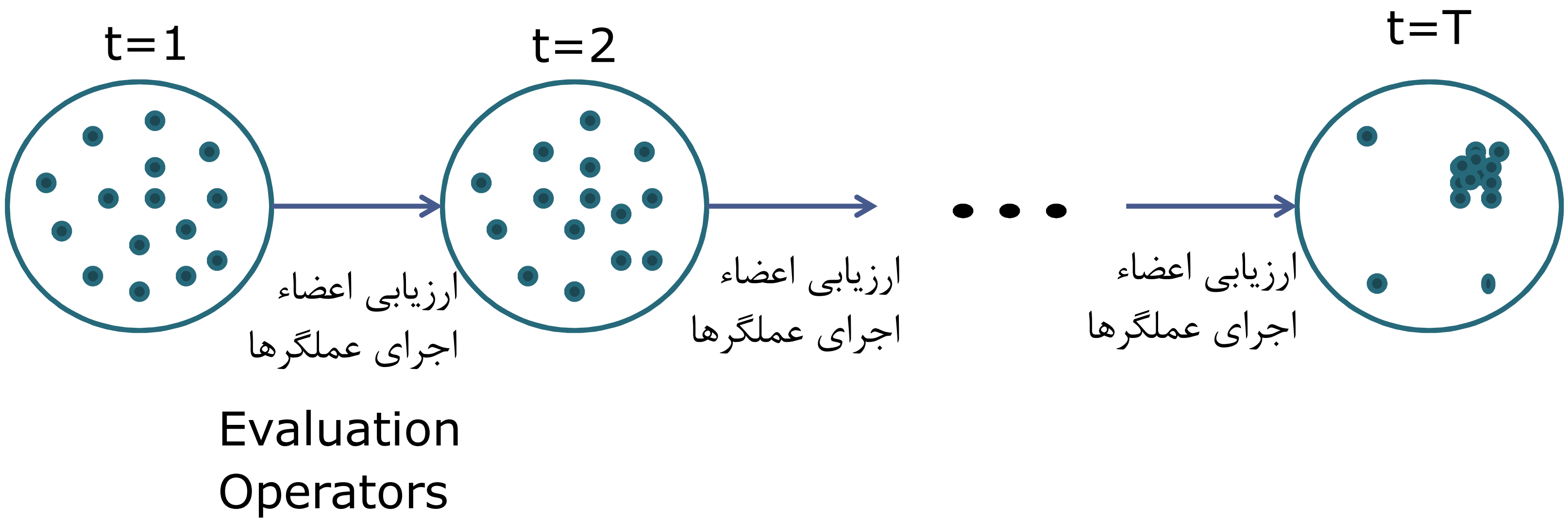
- Optimization algorithm
- Search algorithm
- Heuristic algorithm
- Population based (Agents)
- Iterative search
- Random search
- Parallel search

Population



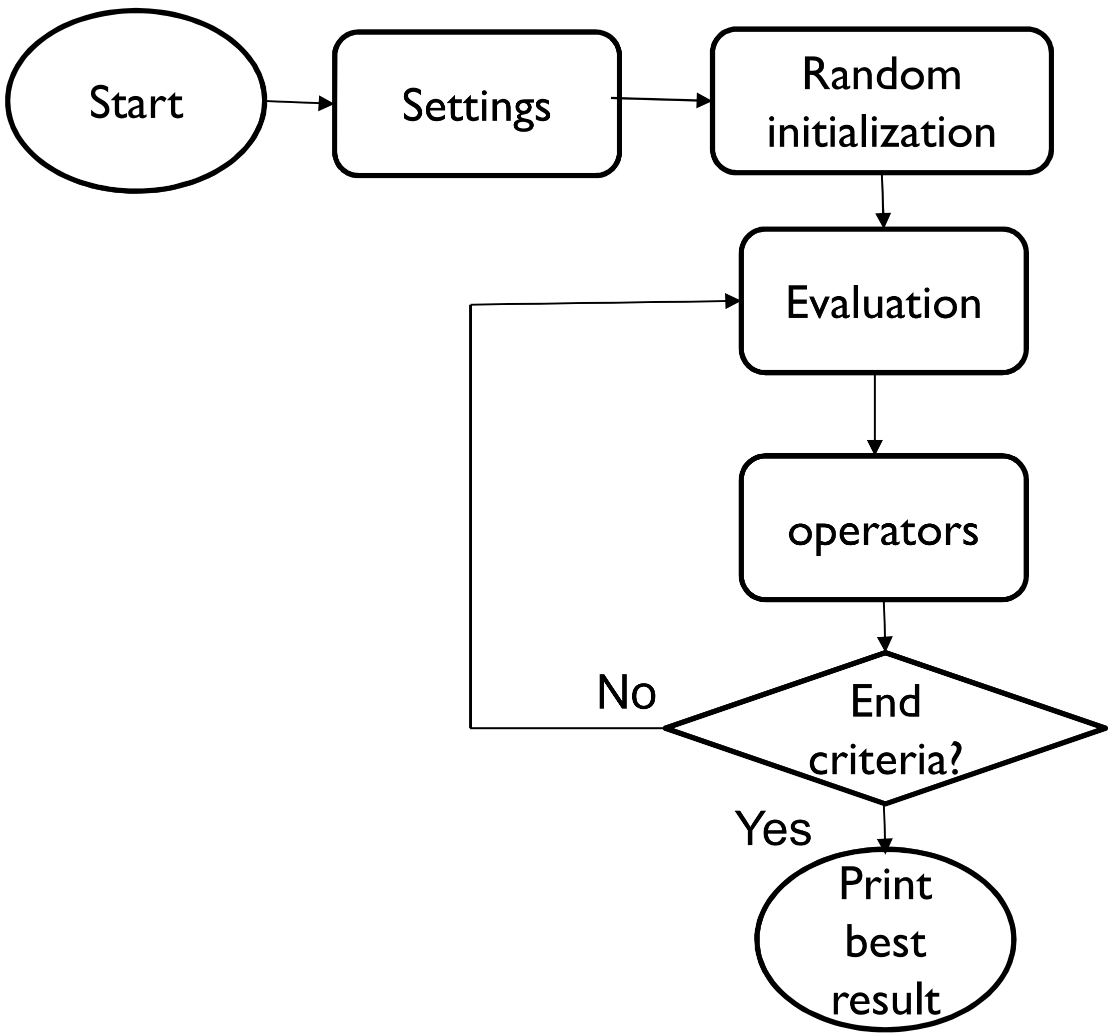
فضای جستجوی چند بعدی
m dimensional search space
An agent
یک عضو جمعیت
X_i

قالب کلی هر الگوریتم بهینه سازی هیوریستیک General form



- جمعیت اولیه تصادفی تولید می شود.
- تولید اعضاء جدید به کمک عملگرهای الگوریتم operators انجام می شود.
- هر الگوریتم تعدادی عملگر ساده دارد.
- الگوریتم تکرارشونده است. هر تکرار یک iteration یا یک time نامیده می شود. (تکرار یا لحظه)

General form of population based heuristic search algorithms

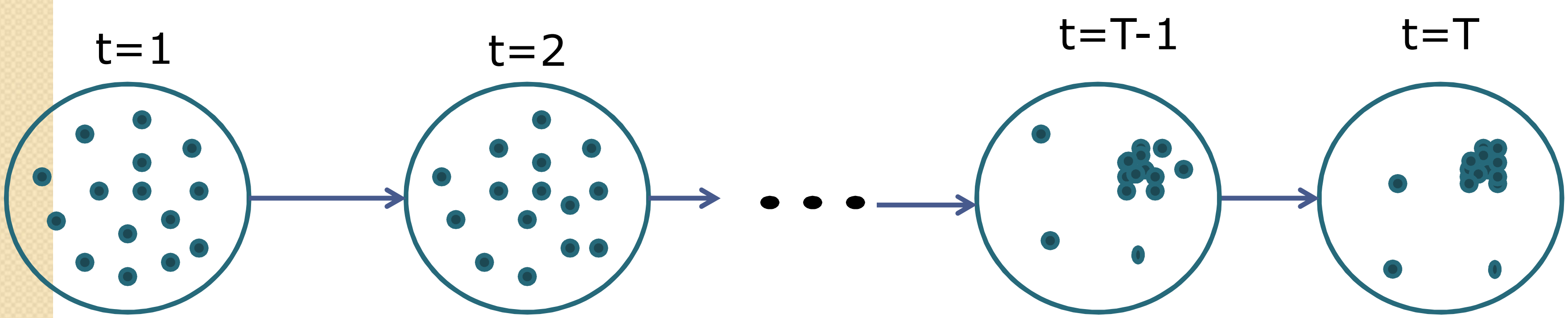


- Convergence
- Exploration / diversification

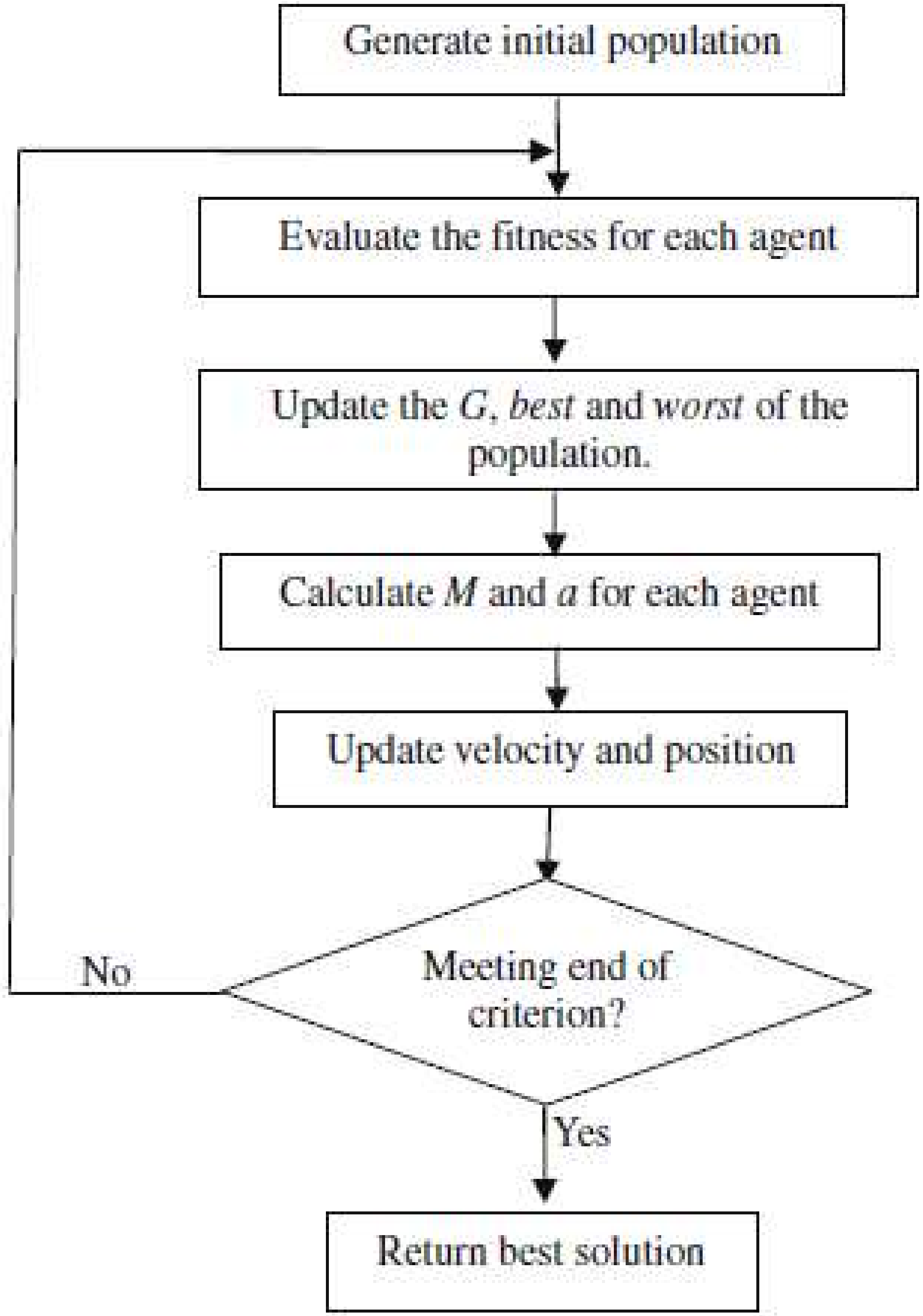
توانایی کاوش نقاط مختلف فضای جستجو

- Exploitation / intensification

توانایی یافتن جواب دقیق در یک منطقه شبه بهینه

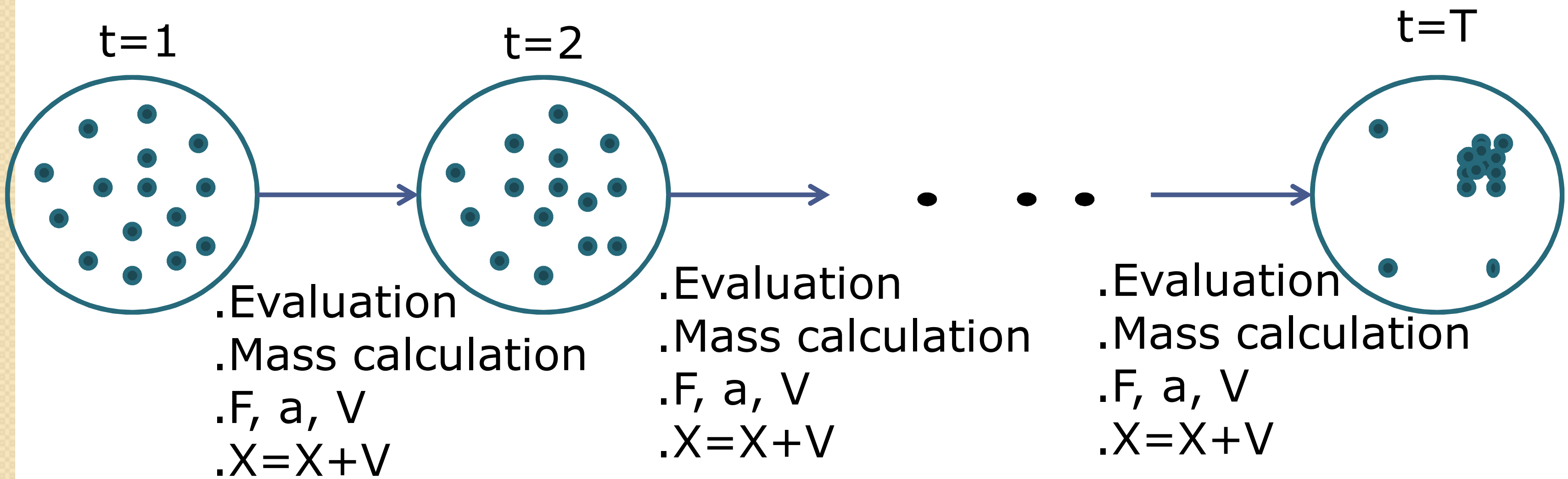


GSA



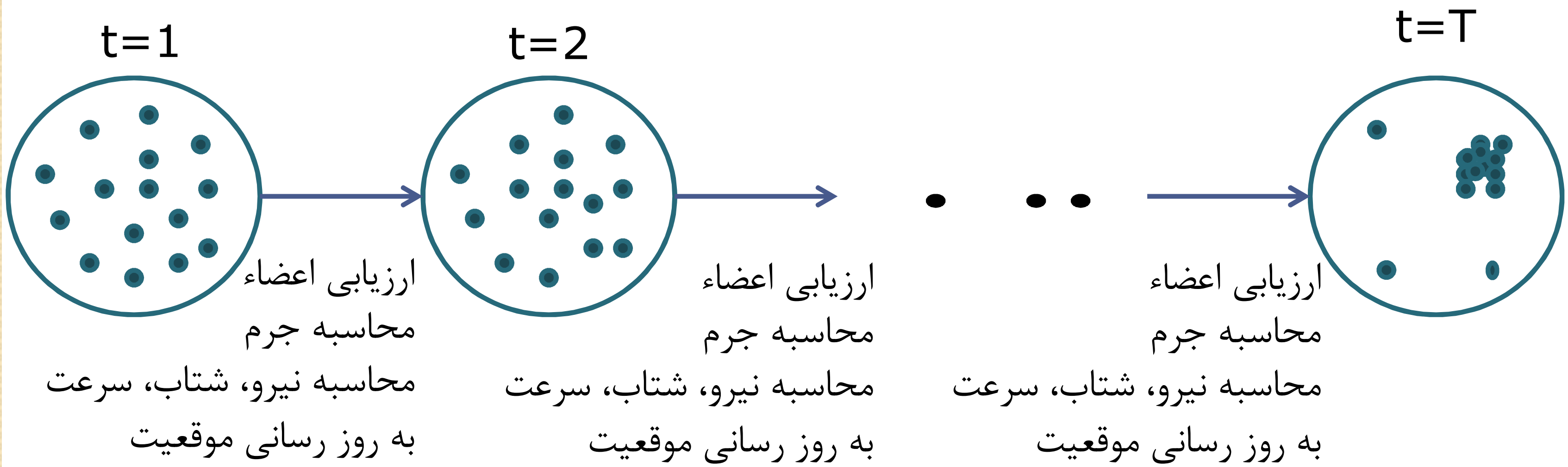
Generate initial population
Evaluate the fitness for each agent
Update the G, best and worst of the population.
Calculate M and a for each agent
Update velocity and position
Meeting end of criterion?
No
Yes
Return best solution

GSA

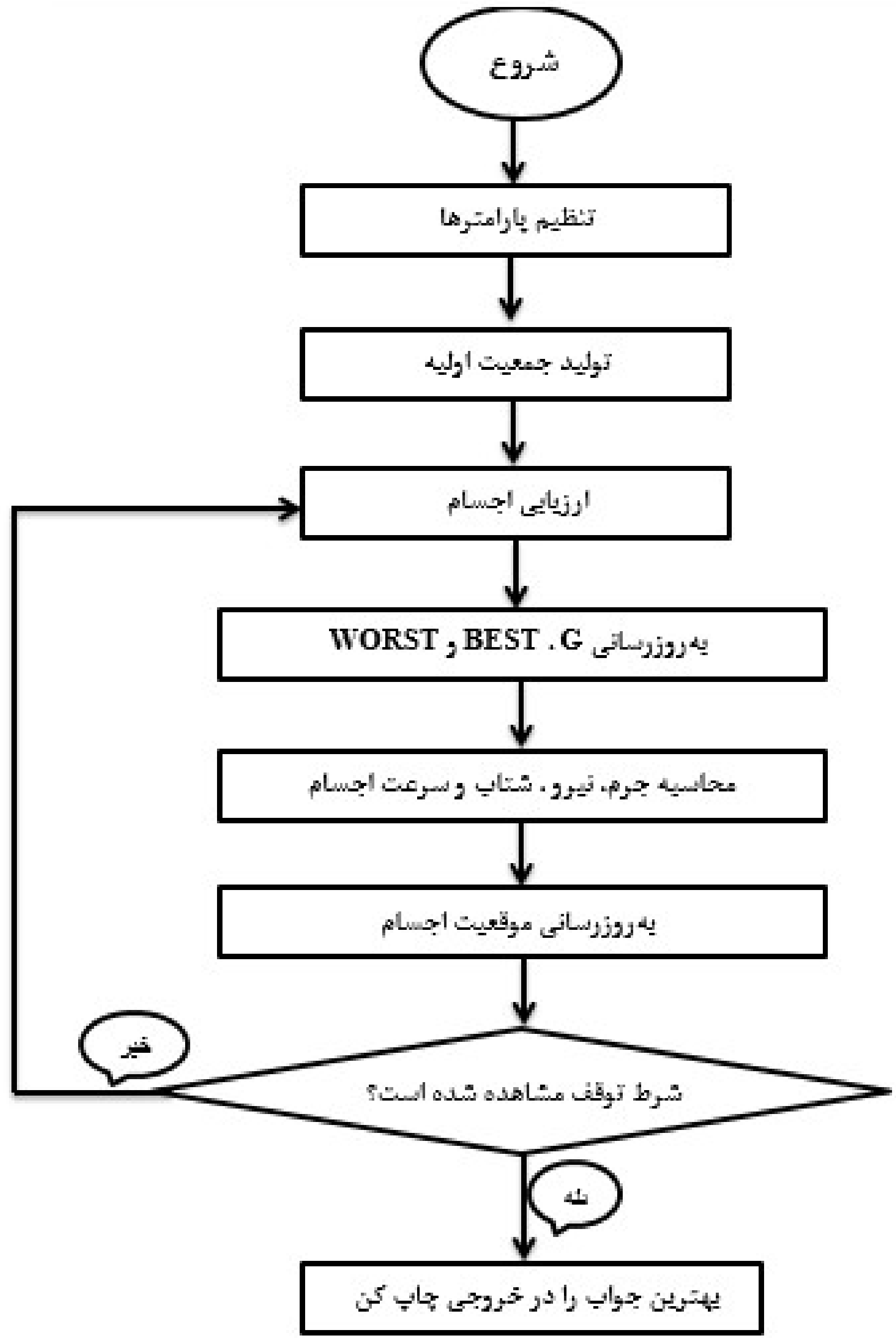


t=1
t=2
t=T
.Evaluation
.Mass calculation
.F, a, V
.X=X+V
.Evaluation
.Mass calculation
.F, a, V
.X=X+V
.Evaluation
.Mass calculation
.F, a, V
.X=X+V

الگوریتم جستجوی گرانشی



GSA


شروع
تنظیم پارامترها
تولید جمعیت اولیه
ارزیابی اجسام
به‌روزرسانی G، BEST و WORST
محاسبه جرم، نیرو، شتاب و سرعت اجسام
به‌روزرسانی موقعیت اجسام
خیر
شرط توقف مشاهده شده است؟
بله
بهترین جواب را در خروجی چاپ کن

Population

$$X = \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \dots \dots \dots x_1^m \\ x_2^1 & x_2^2 \dots \dots \dots x_2^m \\ & . \\ & . \\ & . \\ x_N^1 & x_N^2 \dots \dots \dots x_N^m \end{bmatrix}$$

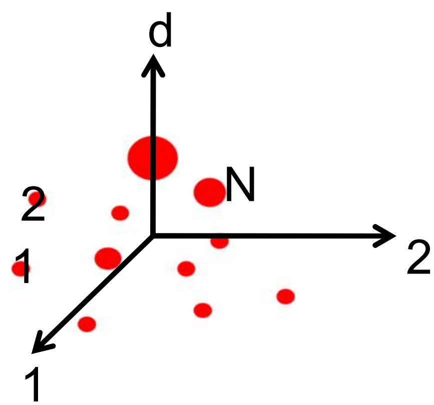
$$X = \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_N \end{bmatrix} = \begin{bmatrix} 0.8 & 3 & 0.1 \dots\ ..10 \\ 1.2 & 2.9 & 0.3 \dots\ ..8 \\ & & . \\ & & . \\ & & . \\ 0.7 & 3.2 & 0.2 \dots\ ..11 \end{bmatrix}$$

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^m) \quad \text{for } i = 1, 2, \dots, N,$$

x_i^d presents the position of ith agent in the dth dimension.

Population

سیستم را به صورت مجموعه ای از N جسم دارای جرم تصور کنید



m=3

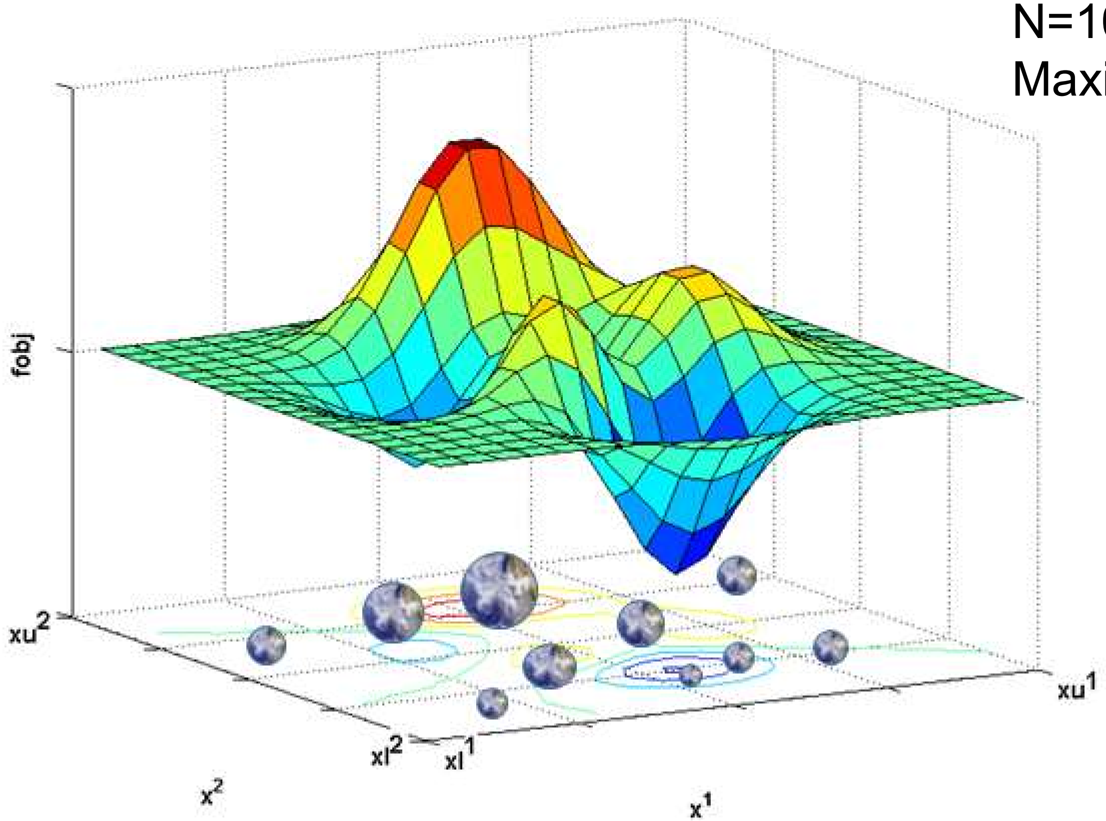
موقعیت جسم iام

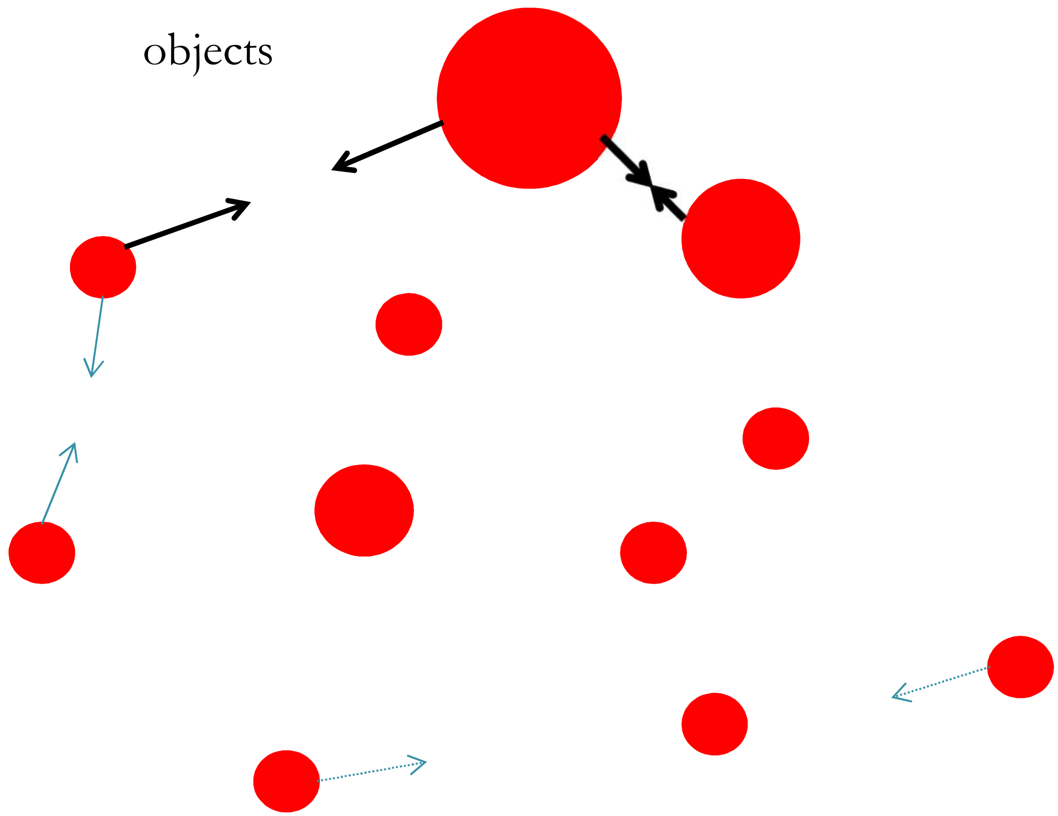
موقعیت بعد d از جسم i با x_i^d نشان داده می شود

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^m), \quad i = 1, 2, \ldots, N$$

Population

m=2
N=10
Maximization example




objects

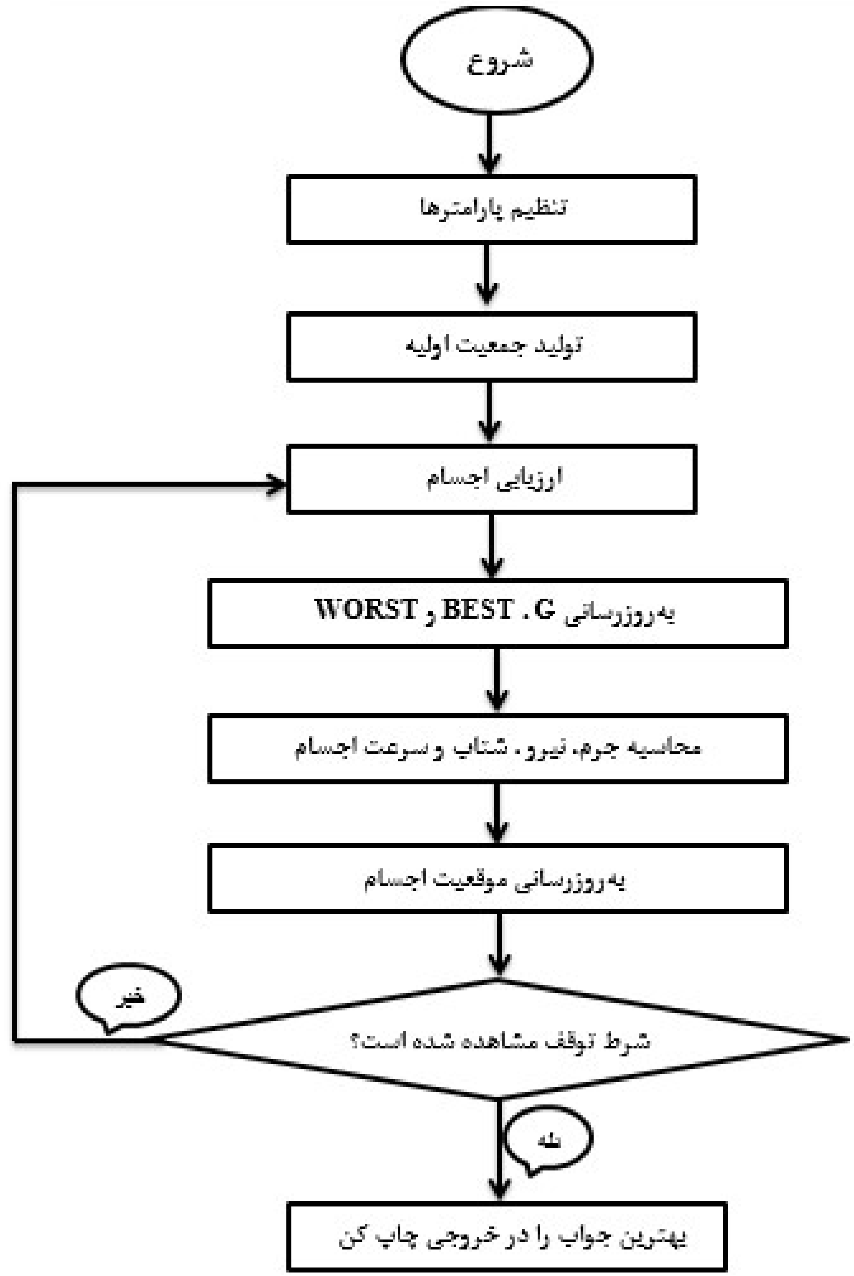
GSA

اجسام در الگوریتم گرانشی از دو قانون پیروی می کنند:

۱- قانون گرانش (law of gravity): هر ذره، ذرات دیگر را با اعمال نیروی که نسبت مستقیم با جرم ذرات و نسبت عکس با مجذور فاصله بین آنها دارد به سوی خود می کشد.

۲- قانون حرکت (law of motion) : سرعت فعلی هر جسم برابر مجموع ضریبی از سرعت قبلی و تغییرات آن است. شتاب هر جسم برابر است با نیروی اعمال شده بر سیستم تقسیم بر جرم اینرسی.

GSA



شروع
تنظیم پارامترها
تولید جمعیت اولیه
ارزیابی اجسام
به‌روزرسانی G ، BEST و WORST
محاسبه جرم، نیرو ، شتاب و سرعت اجسام
به‌روزرسانی موقعیت اجسام
شرط توقف مشاهده شده است؟
خیر
بله
بهترین جواب را در خروجی چاپ کن

Optimization problem

- فرض کنيد که هدف از بهينه‌سازي، پيدا کردن بيشينه تابع f در يک دامنه مشخص باشد:

$$fobj(x^1, x^2, \ldots, x^m)$$

$$x^{i,lo} \leq x^i \leq x^{i,hi} \qquad i=1,2,\ldots,m$$

- در اين وضعيت، پيدا کردن مقاديري براي متغيرهاي x_1 تا x_m مد نظر است که تابع f، به ازاي آنها بيشترين مقدار را به خود بگيرد.

- به عبارتي هدف از بيشنه‌سازي، يافتن X^* است به گونه‌اي که

$$f(X^*) \geq f(X) \quad , \quad \forall X \in D_f$$

Initialization

- موقعیت اولیه اجسام

موقعیت اولیه هر جسم به صورت تصادفی در فضای جستجو با یک توزیع یکنواخت در محدوده تعریف مسئله تعیین می‌شود

$x^{d,hi}$ و $x^{d,low}$ به ترتیب حد پایین و حد بالا از بعد d ام فضای جستجو می‌باشند.

$$x_i^d(t=1) \sim U(x^{d,\text{low}}, x^{d,\text{hi}})$$

$$x_i^d = x_i^{d,lo} + \left(x_i^{d,hi} - x_i^{d,lo}\right) \times x_i^n \qquad \text{for } d = 1, 2, \ldots, m$$

$$i = 1, 2, \ldots, N$$

$$X = \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 \ldots\ldots\ldots x_1^m \\ x_2^1 & x_2^2 \ldots\ldots\ldots x_2^m \\ & . \\ & . \\ & . \\ x_N^1 & x_N^2 \ldots\ldots\ldots x_N^m \end{bmatrix} \qquad X = \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_N \end{bmatrix} = \begin{bmatrix} 0.8 & 3 & 0.1 \ldots\ldots 10 \\ 1.2 & 2.9 & 0.3 \ldots\ldots 8 \\ & . \\ & . \\ & . \\ 0.7 & 3.2 & 0.2 \ldots\ldots 11 \end{bmatrix}$$

Evaluation

- در هر تکرار، به ازای هر عضو یکبار مساله حل می شود.
- برای حل مساله، غالبا مساله در کامپیوتر شبیه سازی می شود.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} \xrightarrow{\text{حل مساله}} \begin{bmatrix} Fit_1 \\ Fit_2 \\ \vdots \\ Fit_N \end{bmatrix}$$

- تعداد کل ارزیابی ها

Total Number of fitness evaluations (FEs)

Mass calculation

اجرام اینرسی و گرانشی با ارزیابی برازندگی قابل محاسبه هستند. جرم سنگین تر به معنی کارآیی بیشتر جرم است. این بدین معنا است که جسم بهتر دارای قدرت جذب بیشتر و حرکت کندتر است.

با فرض برابری جرم اینرسی و گرانشی ، مقدار اجرام می تواند با استفاده از نگاشت تابع هدف محاسبه شود.

$$q_i(t) = \frac{fobj_i(t) - worst(t)}{best(t) - worst(t)}$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{N} q_j(t)}$$

$$M_{ai}(t) = M_{pi}(t) = M_{ii}(t) = M_i(t)$$

$fobj_i(t)$ مقدار تابع هدف جسم i در زمان t است.

Mass calculation

در مسائل کمینه یابی می توان برای محاسبه بهترین و بدترین مقدار شایستگی از روابط زیر استفاده کرد.

$$best(t) = \min_{j \in \{1,.., N\}} fobj_j(t)$$

$$worst(t) = \max_{j \in \{1,.., N\}} fobj_j(t)$$

در مسائل بیشینه یابی، روابط به صورت زیر تعریف می شود.

$$best(t) = \max_{j \in \{1,.., N\}} fobj_j(t)$$

$$worst(t) = \min_{j \in \{1,.., N\}} fobj_j(t)$$

G constant

ثابت گرانش G

$$G(t) = G(G_0, t)$$

یک تابع از مقدار اولیه (G0) و زمان t است

ثابت گرانشی G با یک مقدار اولیه در ابتدا شروع می شود و با زمان کاهش می یابد. از این روش برای کنترل دقت جستجوی الگوریتم استفاده می شود.

T تعداد کل تکرار ها است.

$$G(t) = G_0 e^{-\alpha \frac{t}{T}}$$

Force calculation

در زمان t، نیروی وارد بر جسم i از سوی جسم j در بعد d بصورت رابطه زیر بیان می شود

$$F_{ij}^{d}(t) = G(t)\frac{M_{aj}(t) \times M_{pi}(t)}{R_{ij}(t)^{P} + \varepsilon}\left(x_{j}^{d}(t) - x_{i}^{d}(t)\right)$$

$$d = 1, 2, \ldots, m \quad and \quad i = 1, 2, \ldots, N$$

فاصله اقلیدسی (نرم ۲) بین دو جسم i و j است.

p=1

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2.$$

Force calculation

نیروی مجموع که بر جسم i در بعد d اعمال می شود:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t),$$

i=1,…,N d=1,…,m

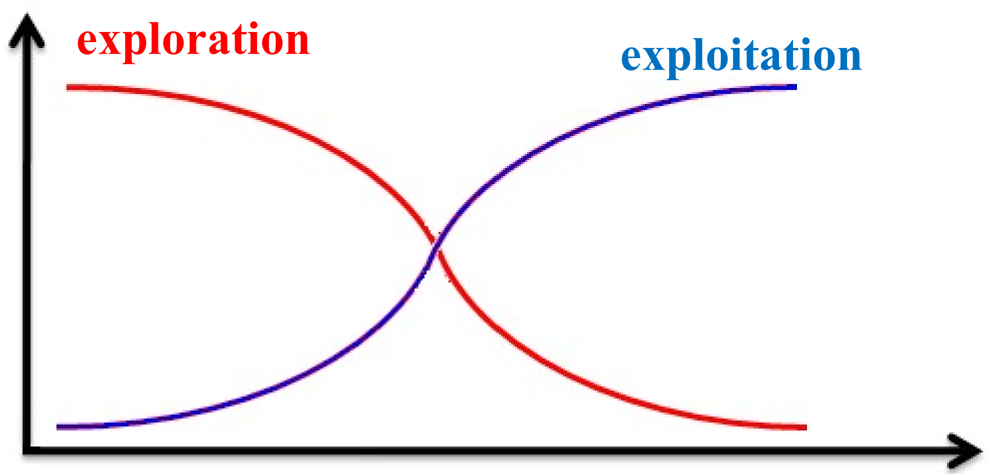
ضریب rand تصادفی بودن الگوریتم را تضمین می کند.

Randj یک عدد تصادفی با توزیع یکنواخت دربارهٔ [۱و۰] است

Force

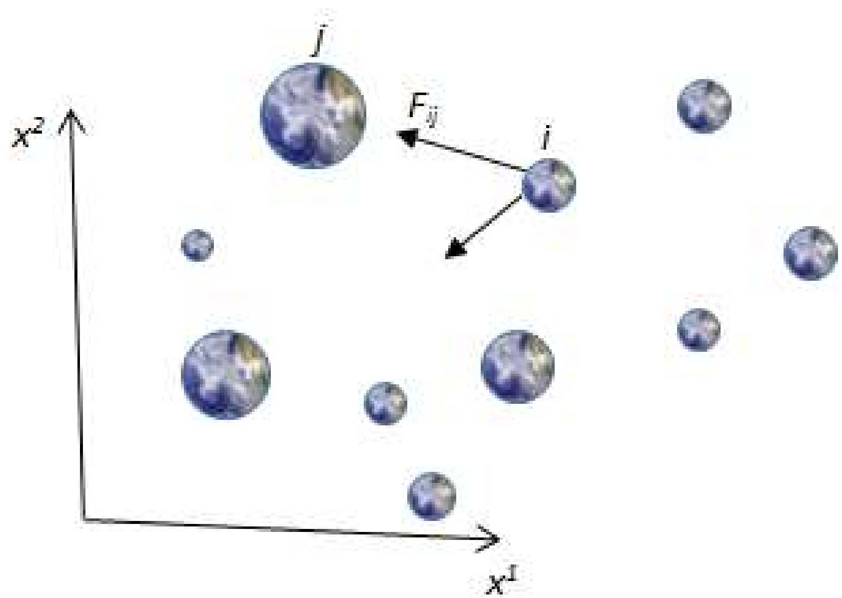
- یک راه برای مصالحه بین کاوش و بهره گیری اینست که تعداد عامل هایی که به سایرین نیرو وارد می کنند را به مرور زمان کاهش دهیم .
- حجم محاسبات نیز کاهش می یابد.
- kbest تعداد k عضو برتر (سنگینتر) جمعیت است. مقدار آن در ابتدا N است و با گذشت زمان به ۲ تا کاهش می یابد.

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t),$$



Force

- m=2
- Kbest=2



Acceleration and velocity calculation

- شتاب جسم i در جهت بعد d در زمان t با $a_i^d(t)$

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}$$

- سرعت جسم را می توان بصورت زیر محاسبه کرد:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$$

$$V = \begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_N \end{bmatrix} = \begin{bmatrix} v_1^1 & v_1^2 \ldots \ldots \ldots v_1^m \\ v_2^1 & v_2^2 \ldots \ldots \ldots v_2^m \\ & \cdot \\ & \cdot \\ & \cdot \\ v_N^1 & v_N^2 \ldots \ldots \ldots v_N^m \end{bmatrix}$$

Randi متغیر تصادفی یکنواخت
در محدودهٔ [۰-۱] است

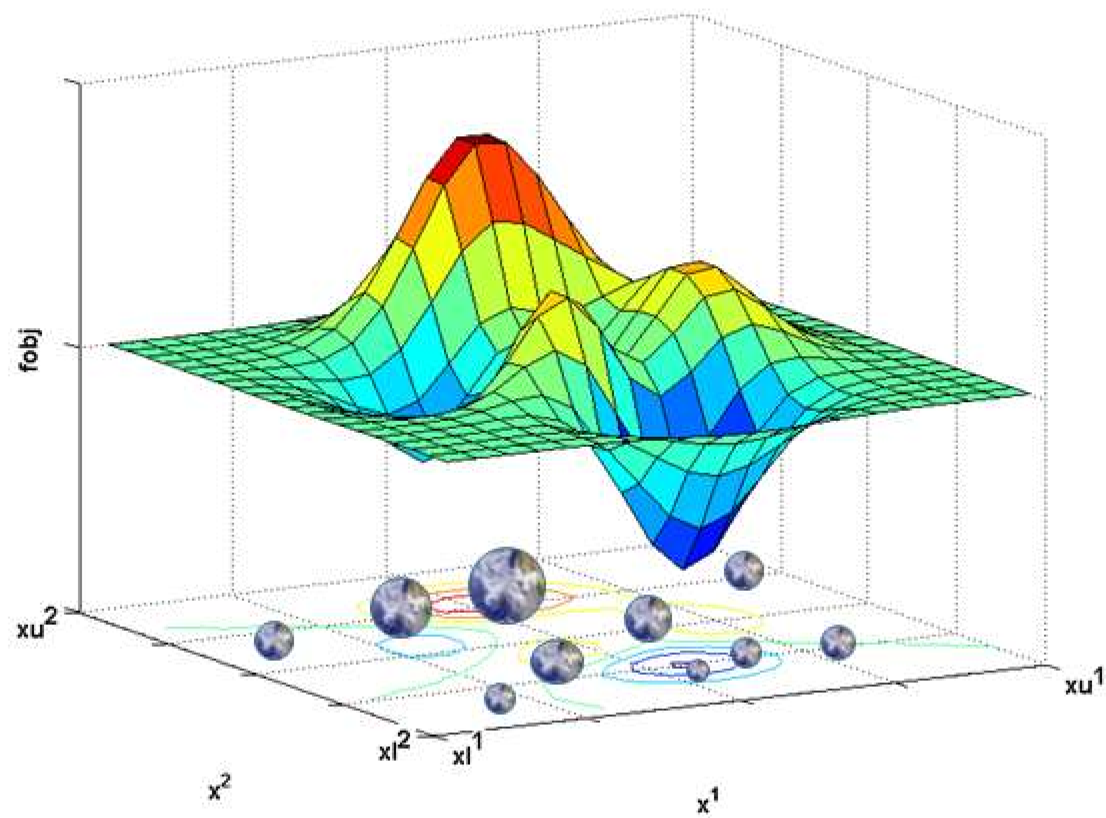
Position updating

- موقعیت جدید جسم را می توان بصورت زیر محاسبه کرد:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$$

Visualization

- m=2
- N=10
- maximization



Various Versions of GSA

- Single objective تک هدفه
- Multi-objective چند هدفه

- Un-constraint نامقید
- Constraint مقید

- Real حقیقی
- Binary باینری
- Discrete گسسته


BGSA

BGSA

الگوریتم جستجوی گرانشی باینری

- بسیاری از مسائل بهینه سازی ماهیت گسسته دارند از این رو نیاز به مدل باینری الگوریتم احساس می شود .
- در فضای باینری، ذرات جستجوگر در یک فضای صفر و یک حرکت می کنند. یک ابر مکعب را در نظر بگیرید که مختصات هر یک از گوشه های آن با صفر و یک مشخص می شود. هر کدام از گوشه های این ابر مکعب یک جواب مسئله است. برای جستجو لازم است اجرام بین گوشه ها حرکت کنند.

Population in BGSA

- هر object یک بردار L بیتی است. با مقادیر ۰ و ۱

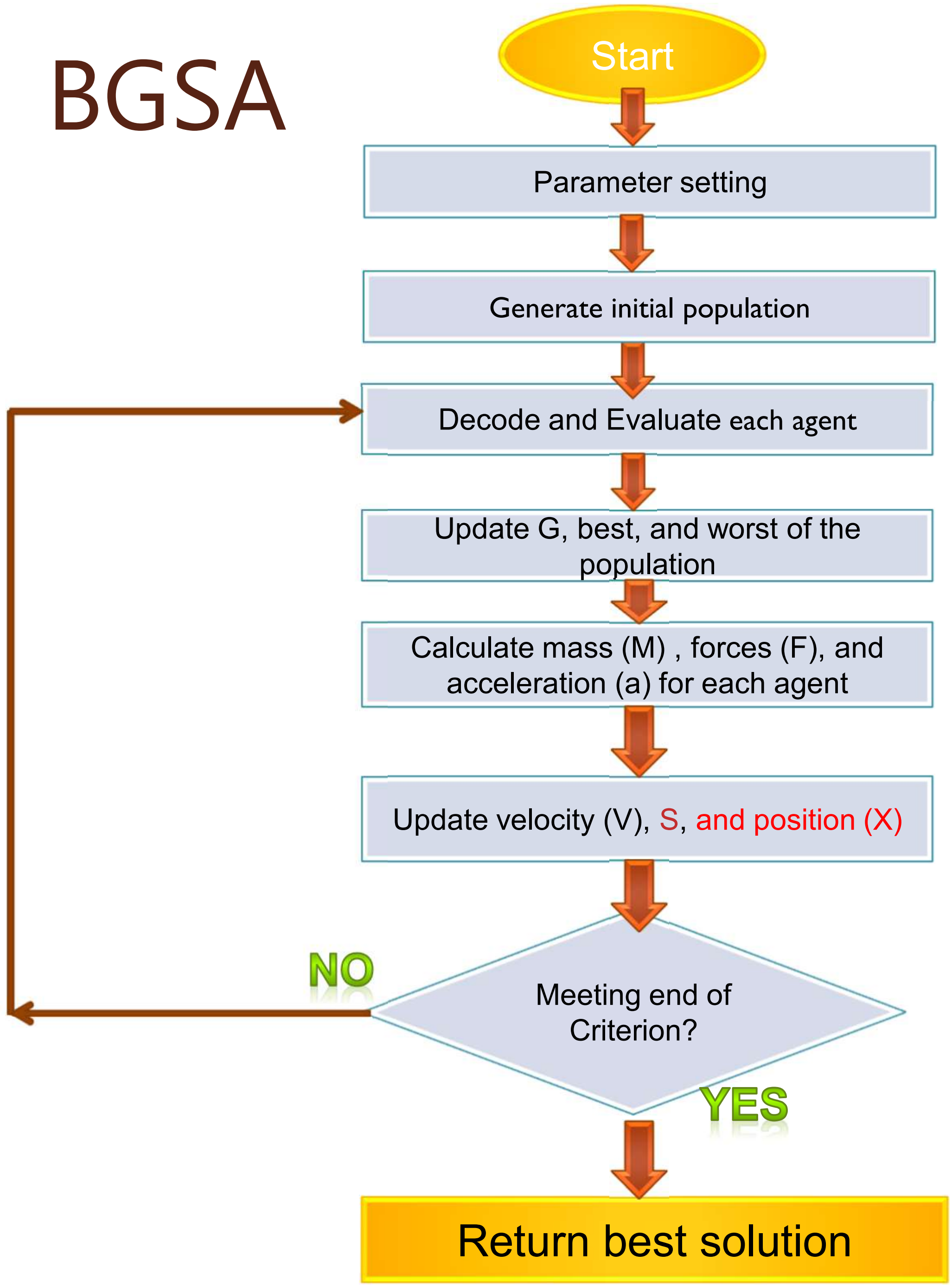
$$X=\begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_N \end{bmatrix}=\begin{bmatrix} x_1^1 & x_1^2 \ldots \ldots \ldots x_1^L \\ x_2^1 & x_2^2 \ldots \ldots \ldots x_2^L \\ & \cdot \\ & \cdot \\ & \cdot \\ x_N^1 & x_N^2 \ldots \ldots \ldots x_N^L \end{bmatrix}$$

$$X=\begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_N \end{bmatrix}=\begin{bmatrix} 0 & 1 & 1 \ldots \ldots 1 & 0 \\ 1 & 1 & 0 \ldots \ldots 0 & 0 \\ & & \cdot \\ & & \cdot \\ & & \cdot \\ 0 & 0 & 1 \ldots \ldots 1 & 1 \end{bmatrix}$$

- هر object یک بردار سرعت L بعدی دارد. با مقادیر حقیقی

$$V=\begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ \cdot \\ V_N \end{bmatrix}=\begin{bmatrix} v_1^1 & v_1^2 \ldots \ldots \ldots v_1^L \\ v_2^1 & v_2^2 \ldots \ldots \ldots v_2^L \\ & \cdot \\ & \cdot \\ & \cdot \\ v_N^1 & v_N^2 \ldots \ldots \ldots v_N^L \end{bmatrix}$$

BGSA



- در الگوریتم جستجوی گرانشی باینری ، محاسبه نیروی وارد بر هر جسم و سرعت هر جسم و روابط بروز رسانی اجرام، مطابق مدل الگوریتم حقیقی انجام می شود.
- در این الگوریتم از فاصله همینگ به جای فاصله اقلیدسی استفاده می شود.

BGSA Gravitational constant

ثابت گرانش در نسخه باینری

رابطه G به فرم رابطه زیر به صورت خطی کاهش می یابد. چرا که در فضای باینری ضریب گرانش روی تغییرات هر بیت اثر می گذارد نه هر بعد و بهتر است آهنگ کند تری داشته باشد.

$$G(t) = G_0\left(1 - \frac{t}{T}\right)$$

BGSA position updating

- حرکت ذره در هر بعد به معنای تغییر مقدار آن از صفر به یک و بالعکس خواهد بود.
- در این روش اینگونه عمل می شود که سرعت جسم در هر بعد به یک تابع احتمال تبدیل می شود و بر مبنای آن، ذره با یک احتمال در آن بعد تغییر موقعیت می دهد.
- در واقع در نسخه باینری v_i^d به جای آنکه بیانگر جابجایی ذره باشد، احتمال تغییر موقعیت x_i^d را نشان می دهد.

BGSA

- جهت تعریف تابع انتقال برای نگاشت سرعت به احتمال جابجایی ذره، چند مفهوم مهم الگوریتم جستجوی گرانشی استاندارد را متذکر می شویم:

مقدار بزرگ سرعت بیان کننده اینست که ذره تمایل بیشتری به جابجایی دارد.
مقدار کوچک سرعت بیانگر اینست که ذره تمایل کمتری به حرکت و تمایل بیشتری به سکون دارد.

براساس مفاهیم فوق ، در BGSA :

- مقدار بزرگ سرعت، باید یک احتمال بزرگ برای تغییرات مکانی جسم ایجاد کند (از صفر به یک و برعکس).
- مقدار کوچک سرعت، باید یک احتمال کوچک برای تغییرات مکانی جسم ایجاد کند.
- مقدار صفر سرعت، نشان دهنده تمایل به عدم تغییر است.

BGSA

براساس موارد ذکر شده، تابع احتمال مناسب باید طوری تعریف شود که برای یک $|v_i^d|$ کو چک، احتمال تغییر x_i^d نزدیک به صفر و برای $|v_i^d|$ بزرگ، احتمال حرکت بزرگ باشد.

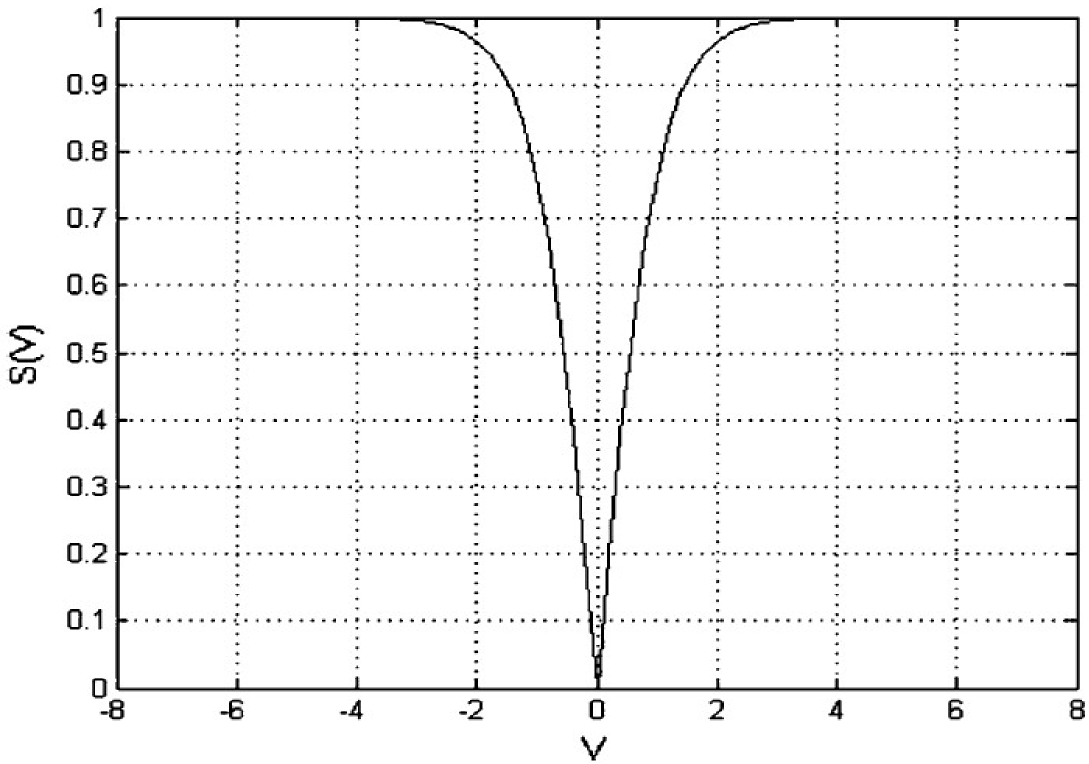
تابع $S(v_i^d)$ را برای تبدیل v_i^d به تابع احتمال تعریف می کنیم. این تابع باید در محدوده [۰-۱] قرار گیرد و با افزایش $|v_i^d|$ افزایش یابد.

رابطه زیر نحوه محاسبه را نشان می دهد.

$$S(v_i^d(t)) = \left|\tanh(v_i^d(t))\right|$$

BGSA

شکل تابع S(Vid)



برای همگرایی مناسب الگوریتم، v_i^d باید به یک بازه مناسب محدود شود.

مقدار V_{max} برابر ۶ در نظر گرفته می شود.

$$|v_i^d| < v_{max}$$

BGSA

جسم در هر بعد مطابق رابطه زیر حرکت می کند:

$$if \quad rand < S(v_i^d(t+1)) \quad then$$
$$x_i^d(t+1) = complement(x_i^d(t))$$
$$else \quad x_i^d(t+1) = x_i^d(t)$$

طبق این رابطه، جسم با یک احتمال تغییر موقعیت می دهد که هرچه سرعت جسم در یک بعد بیشتر باشد، احتمال حرکت جسم در آن بعد بیشتر می شود.

Rand یک عدد تصادفی با توزیع یکنواخت در بازه [۱ و ۰] است.

Evaluation and Comparison

ارزیابی کارآیی الگوریتم

- ارزیابی کارآیی الگوریتم، از طریق تحلیل همگرایی و زمان رسیدن به پاسخ بهینه بررسی می‌شود. برای مشاهده وضعیت پیشرفت الگوریتم و پایش معیارهای فوق، از روشهای ترسیمی کمک گرفته می‌شود.
- متوسط شایستگی جمعیت mean-fitness
- بهترین جواب دیده شده تا آن لحظه best-so-far

ارزیابی کارایی الگوریتم

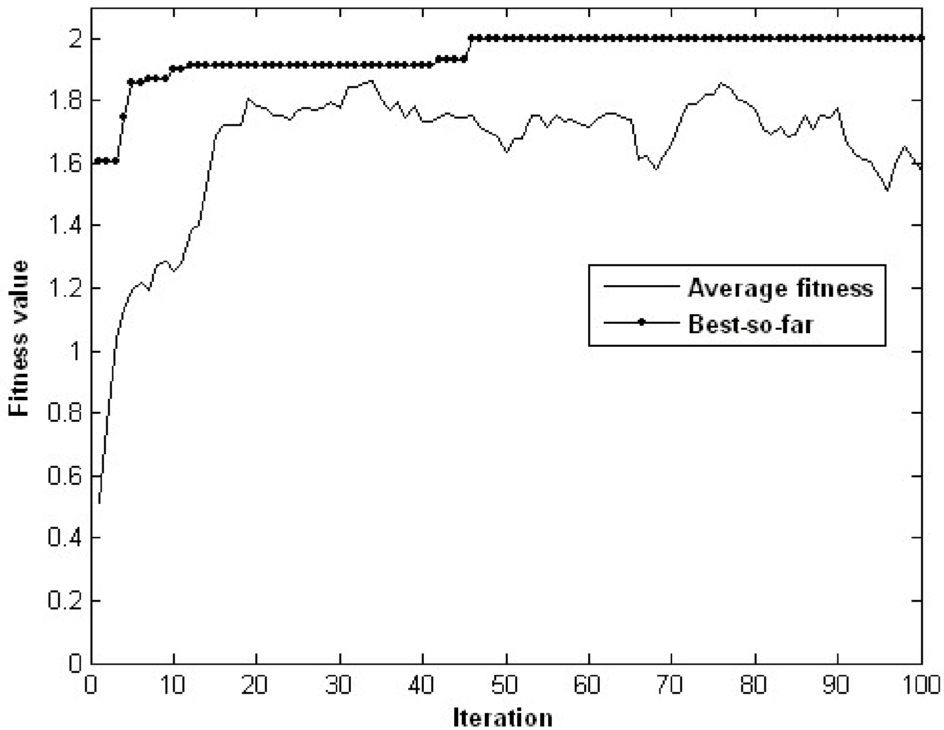
$$
\begin{array}{ccccc}
t=1 & t=2 & & t=T \\
\begin{bmatrix} fit_1(1) \\ fit_2(1) \\ fit_3(1) \\ fit_4(1) \\ . \\ . \\ . \\ fit_{N-1}(1) \\ fit_N(1) \end{bmatrix} &
\begin{bmatrix} fit_1(2) \\ fit_2(2) \\ fit_3(2) \\ fit_4(2) \\ . \\ . \\ . \\ fit_{N-1}(2) \\ fit_N(2) \end{bmatrix} &
\ldots\ldots\ldots\ldots &
\begin{bmatrix} fit_1(T) \\ fit_2(T) \\ fit_3(T) \\ fit_4(T) \\ . \\ . \\ . \\ fit_{N-1}(T) \\ fit_N(T) \end{bmatrix}
\end{array}
$$

mean Fitness=[mf(1) mf(2) .... mf(T)] میانگین هر تکرار

best So far =[bsf(1) bsf(2) .... bsf(T)] بهترین تا هر تکرار

ارزیابی کارآیی الگوریتم در یکبار اجرا

- Maximization example



T=100

ارزیابی و مقایسه کارآیی الگوریتم ها

آیا مقایسه الگوریتم ها با یک بار اجرای هر الگوریتم عادلانه است؟

ارزیابی و مقایسه کارآیی

- الگوریتم ها تصادفی هستند. و با هر بار اجرا جواب های متفاوتی می دهند.
- برای مقایسه عادلانه، هر الگوریتم را چند بار اجرا کرده و میانگین نتایج را گزارش می کنند.

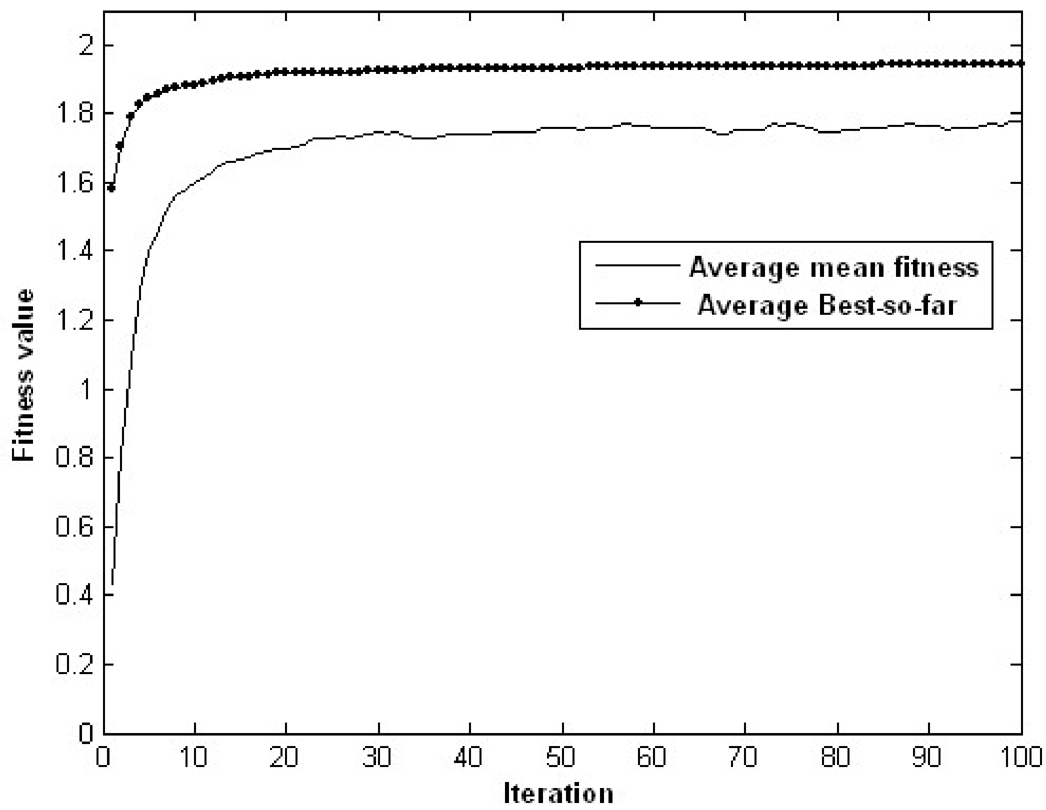
ارزیابی در چند اجرای مستقل الگوریتم

Run1	[bsf(1)	bsf(2)	....	bsf(T)]
Run2	[bsf(1)	bsf(2)	....	bsf(T)]
Run3	[bsf(1)	bsf(2)	....	bsf(T)]
.				
.				
.				
.				
Run20	[bsf(1)	bsf(2)	....	bsf(T)]

Average best So far =[absf(1)	absf(2)	....	absf(T)]

ارزیابی کارآیی الگوریتم

- متوسط معیارهای قبلی در چند اجرای مستقل



T=100
maximization

مقایسه کارآیی الگوریتم های مختلف

- رسم کردن نمودارهای الگوریتم ها در یک شکل
- استفاده از جداول

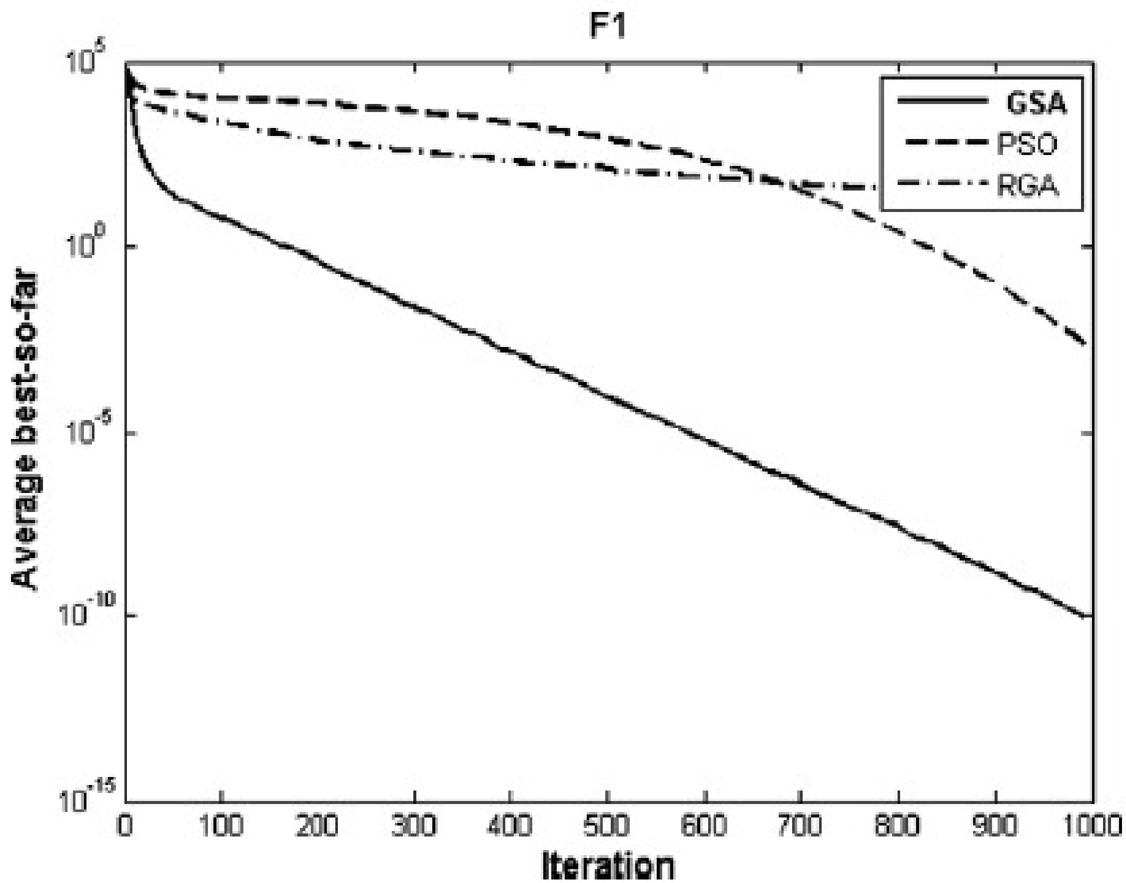
Comparison

- Experiments and comparison

- الگوریتم ها از طریق آزمایش و مقایسه با سایر روش ها بررسی می شوند.
- معیار مقایسه دقت جواب ها، زمان رسیدن به پاسخ، و زمان اجرا و پیچیدگی الگوریتم ها است.

مقایسه کارآیی الگوریتم های مختلف

- مقایسه با رسم کردن نمودارهای الگوریتم ها در یک شکل
- Minimization example



T=1000

مقایسه کارآیی الگوریتم های مختلف

- مقایسه با استفاده از جداول

Run1	[bsf(1)	bsf(2)	....	bsf(T)]
Run2	[bsf(1)	bsf(2)	....	bsf(T)]
Run3	[bsf(1)	bsf(2)	....	bsf(T)]
.				
.				
.				
.				
Run20	[bsf(1)	bsf(2)	....	bsf(T)]

Best

Mean±var

مقایسه کارآیی الگوریتم های مختلف

- مقایسه با استفاده از جداول

Algorithm	Mean ± Var	Best	Time (sec)
A	1.0 ± 0.2	1.22	2
B	1.7 ± 0.1	1.72	1.5
C	0.9 ± 0.05	0.95	2.5

GSA

[1] E.Rashedi, H.Nezamabadi-pour, S.Saryazdi, "A gravitational search algorithm", Information Science, 2009

[2] E.Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: binary gravitational search algorithm", Natural Computing, 2010.