Firefly Algorithm

مقدمه

الگوریتم کرم شب تاب (Firefly Algorithm) به اختصار, FA الگوریتم فراکاوشی است در سال ۲۰۰۷ و توسط Xin-She Yang معرفی شده است. ایده اصلی این الگوریتم، از ارتباط نوری میان کرم های شب تاب الهام گرفته شده است. این الگوریتم را می توان از مظاهر هوش ازدحامی دانست که در آن از همکاری (و احتمالا رقابت) اعضای ساده و کم هوش، مرتبه بالاتری از هوشمندی ایجاد می شود که قطعا توسط هیچ یک از اجزا قابل حصول نیست.

کرم های شب تاب برخلاف نامشان جزء خانواده حشرات می باشند و توانایی تولید نور سرد را دارند. در حقیقت کرم های شب تاب پترن های خاصی را میتوانند تولید کنند که در گونه های مختلف کرم شب تاب این پترن ها منحصر بفرد هستند.هدف کرم های شب تاب از تولید این پترن ها و تولید نور, جذب سایر کرم های شب تاب و شکار و یا حتی میتوانند از این پترن ها بعنوان وسیله دفاعی استفاده کنند و مهاجمان را فراری بدهند که هدف اصلی در این الگوریتم جذب سایرین است.

تئوری کار الگوریتم

با توجه به فیزیک نور, شدت نور (I) کرم شب تاب با مجذور فاصله رابطه عکس دارد, یعنی هرچه قدر فاصله بیشتر باشد از شدت نور کاسته میشود:

رابطه 1: $I \propto 1/r^2$

در این الگوریتم شدت نور متناسب با تابع هدف مسئله ای است که باید بهینه شود. به منظور فرمول سازی الگوریتم کرم شب تاب ، خالق این الگوریتم در مورد کرم شب تاب ها سه قانون زیر را در نظر گرفته است :

- همه ی کرم های شب تک جنسیتیی هستند . (یا نر یا ماده)
- جذابیت آنها متناسب با شدت نور آنها می باشد .
- شدت نور یک کرم شب تاب توسط تابع هدف تحت تاثیر قرار می گیرد و یا تعیین می شود .

توجه داشته باشید که شدت نور (I) به عنوان مقدار مطلقی از نور منتشر شده توسط کرم شب تاب اشاره می شود ، جذابیت یک سنجش نسبی از نوری است که باید توسط کرم های شب تاب دیگر قضاوت شود . شدت نور(I) با فاصله r با توجه به رابطه زیر تغییر میکند :

$$I(r) = I_0 e^{-\gamma r^2}$$

رابطه 2 :

در این جا I_0 شدت نور را در منبع نشان می دهد و γ ضریب جذب نور ثابت می باشد . به طور مشابه جذابیت (β) نیز بر فاصله ی r بستگی دارد و با توجه به معادله ی کلی زیر محاسبه می شود :

$$\beta(r) = \beta_0 e^{-\gamma r^2}.$$

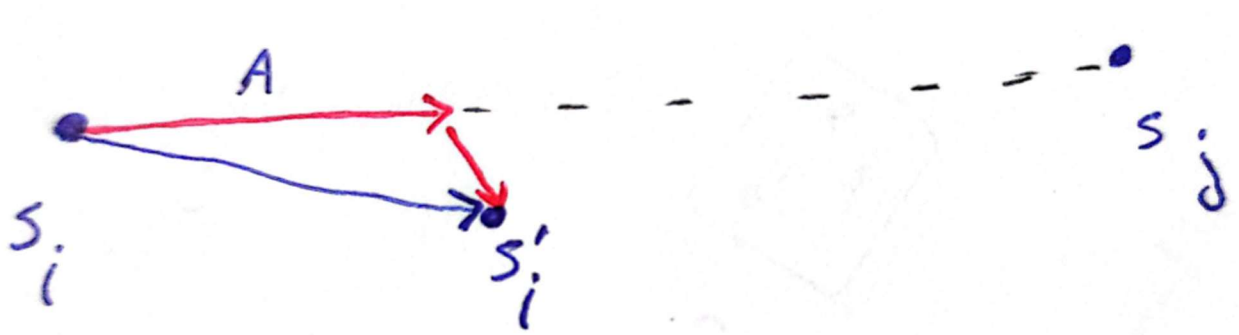
رابطه 3 :

فاصله دو کرم شب تاب i , j :

$$r_{ij} = \|s_i - s_j\|$$

رابطه 4 :

در حقیقت, هر کرم شب تاب i به سمت کرم شب تاب جذاب تر j حرکت می کند ، مانند زیر :



که در اینجا بردار A با رابطه $\beta_0 e^{-\gamma r_{ij}^2}(s_j - s_i)$ بدست می آید و با جمع کردن با بردار تصادفی مسیر حرکت کرم شب تاب را از راستای کرم شب تاب جاذب منحرف میکنیم که این بردار هم با رابطه $\alpha \cdot N_i(0, 1)$ بدست می آید. این بخش شامل پارامتر تصادفی α و اعداد تصادفی $N_i(0,1)$ حاصل از توزیع گاوسی است .

رابطه کلی حرکت کرم شب تاب به صورت گونه جذاب تر به صورت زیر است:

رابطه 5 :

$$s_i = s_i + \beta_0 e^{-\gamma r_{ij}^2}(s_j - s_i) + \alpha \cdot N_i(0, 1).$$

در ادامه با استفاده از این الگوریتم قصد داریم تابع محک sphere را بهینه بکنیم.

بهینه سازی تابع sphere

تابع محک sphere به صورت زیر است :

```
1    function z = Sphere(x)
2
3        z = sum(x.^2);
4
5    end
```

که در آن مجموع مربعات ورودی ها محاسبه میشود.هدف ما از بهینه کردن این تابع این است که بتوانیم به کمترین مقدار دست بیابیم.(ارزشمندی ما در این مسئله مقدار کمتر است.یعنی کرم شب تابی که شدت نور کمتری دارد)

مراحل الگوریتم کرم شب تاب در این مسئله

- تشخیص تابع هدف
- تولید پاسخ های اولیه و ارزیابی آنها (تولید جمعیت اولیه کرم شب تاب و ارزیابی شدت نور آنها)
- به ازای هر کرم شب تاب i و j :
  - اگر $I_i<I_j$ => با توجه رابطه 5 کرم شب تاب به سمت کرم با ارزش حرکت میکند.
  - ارزیابی کرم شب تاب جدید
  - تعیین بهترین پاسخ های یافت شده
- تکرار مرحله سه به دفعات لازم

که در این مسئله تابع sphere تابع هدف ما است و قرار است خروجی تابع sphere را بهینه کنیم.(هر چه جواب ما کمتر باشد بهتر است)

مراحل الگوریتم نویسی

- تعریف مسأله و تعیین تابع هدف
- تعریف پارامتر ها
- آماده سازی
- حلقه اصلی الگوریتم
- نمایش نمودار ها و نتایج

اکنون که تابع هدف مشخص شده است,شروع به نوشتن برنامه نویسی الگوریتم میکنیم.برای این منظور ابتدا فایل fa ایجاد میکنیم.

مرحله 1 : تعریف مسأله

```
 7 -     costFunction = @(x) Sphere(x);
 8
 9 -     n_x = 2;
10
11 -     xSize = [1 n_x];
12
13 -     xMin = -10;
14 -     xMax = 10;
```

- در خط 7 تابع هدف مشخص شده است که همان تابع sphere است.
- در خط 9 تعداد پارمتر های ورودی تابع هدف مشخص شده است.
- در خط 11 اندازه ماتریس که جلوتر در بخش توزیع نورمال و تولید جمعیت استفاده خواهد شد ,مشخص شده است.
- در خط 13 و 14 به ترتیب مینیمم مقدار و ماکسیمم مقدار ورودی تابع هدف مشخص شده است.

مرحله 2 : تعریف پارمتر های مسأله

```
18 -     It = 500;
19
20 -     nPop = 40;
21
22 -     beta0 = 2;
23
24 -     gama = 1;
25
26 -     alpha = 0.2;
27
28 -     delta = 0.05*(xMax-xMin);
29
30 -     alphaRatio = 0.99;
```

- در خط 18 تعداد تکرار های حلقه اصلی را مشخص میکنیم.
- در خط 20 تعداد جمعیت کرم های شب تاب را مشخص کرده ایم.
- در خط بعدی پارامتر بتای صفر که در رابطه 5 بود را تعریف میکنیم و در این مسئله عدد 2 برای این پارامتر خوب است.
- در خط 24 پارامتر گاما را مشخص میکنیم و در این مسئله برابر 1 قرار میدهیم.
- در خط 26 مقدار اولیه آلفا که برابر 0.2 است را قرار داده ایم.

- در خط 28 حد بالا و پایین بردار تصادفی را مشخص میکنیم.این مرحله در تعیین بردار تصادفی به صورت توزیع یکنواخت به کار ما می آید.
- در خط بعدی هم نرخ کاهش نمایی آلفا را مشخص میکنیم.در حقیقت در این مسأله آلفا را به صورت نمایی کاهش میدهیم.در خطوط جلوتر به کارمان خواهد آمد.

مرحله 3 : آماده سازی

```
34 -    Firefly.pos = [];
35 -    Firefly.cost = [];
36
37 -    bestSol.cost = inf;
38
39 -    for i=1:nPop
40 -        Firefly(i).pos = unifrnd(xMin,xMax,xSize);
41 -        Firefly(i).cost = Sphere(Firefly(i).pos);
42
43 -        if Firefly(i).cost <= bestSol.cost
44 -            bestSol = Firefly(i);
45 -        end
46 -    end
```

- در خطوط 34 و 35 شروع به ساختن ساختاری برای ذخیره سازی کرم شب تاب های تولید شده میکنیم.اسم این ساختار را Firefly گذاشتیم و دارای دو ستون pos (موقعیت-ورودی تابع sphere) و cost (ارزش - خروجی تابع sphere)
است.
- در خط 37 برای بهترین جواب ها یک ساختار ایجاد میکنیم و چون اول کار خیلی مهم نیست که مقدارش چقدر باشد,ما هم بی نهایت در نظر میگیریم.
- خط 39 تا 46 یک حلقه برای تولید کرم شب تاب ها است.این حلقه که به تعداد کرم های شب تاب تکرار خواهد شد,در هر تکرار به صورت یکنواخت بین ماکسیمم و مینیمم مقدار ورودی که مشخص کردیم,به تعداد ورودی ها اعداد تصادفی تولید میکند و در قالب ماتریسی که در خط 11 تعریف کرده بودیم در ساختار Firefly ذخیره میکند(خط 40).و همچنین همین موقعیت هایی که ساخته شده را در خط 41 به تابع هدف میدهیم و ارزش هر کرم شب تاب را مشخص کرده و در ساختار Firefly ذخیره میکنیم.
- در هر تکرار همچنین چک میکنیم اگر ارزش کرم شب تاب تولید شده اگر بهتر باشد در ساختار bestSol ذخیره میکنیم.

تا الان, مسئله تعریف شده و جواب های تصادفی(کرم شب تاب ها)تولید شده اند.اکنون وارد حلقه اصلی الگوریتم میشویم:

مرحله 4 : حلقه اصلی

```
49 -    bestCost = zeros(1,It);
50 -    copyOfFirefly= Firefly;
51
52 -    for it=1:It
53 -        for i=1:nPop
54 -            for j=1:nPop
55 -               if Firefly(j).cost <= Firefly(i).cost
56 -                   r = norm(Firefly(i).pos - Firefly(j).pos);
57 -                   beta = beta0*exp(-gama*r^2);
58 -                   e = delta*unifrnd(-1,+1,xSize);
59
60 -                   newFirefly(i).pos = Firefly(i).pos + beta * (Firefly(j).pos - Firefly(i).pos) + alpha*e; %#ok
61
62 -                   newFirefly(i).cost = costFunction(newFirefly(i).pos); %#ok
63 -                   newFirefly(i).pos = max(newFirefly(i).pos,xMin); %#ok
64 -                   newFirefly(i).pos = min(newFirefly(i).pos,xMax); %#ok
65
66 -                    if newFirefly(i).cost<=bestSol.cost
67 -                        bestSol=newFirefly(i);
68 -                    end
69 -               end
70 -            end
71 -        end
72 -        Firefly = [newFirefly Firefly bestSol];
73 -        [~,order] = sort([Firefly.cost]);
74 -        Firefly = Firefly(order);
75 -        Firefly = Firefly(1:nPop);
76
77 -        bestCost(it)=bestSol.cost;
78 -         disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(bestCost(it))]);
79
80 -        alpha=alpha*alphaRatio;
81 -    end
```

- در خط 49 یک ماتریسی ایجاد میکنیم که جواب های نهایی هر تکرار را در آن ذخیره سازی کنیم.
- در خط 50 برای ترسیم شکل در مرحله اخر یک کپی از ساختار Firefly برمیداریم.(این ساختار وقتی که وارد حلقه اصلی شود در هر تکرار آپدیت خواهد شد و مقدار اولیه را از دست خواهیم داد.برای ترسیم شکل به مقدار اولیه نیازمندیم که از این کپی استفاده خواهیم کرد.)

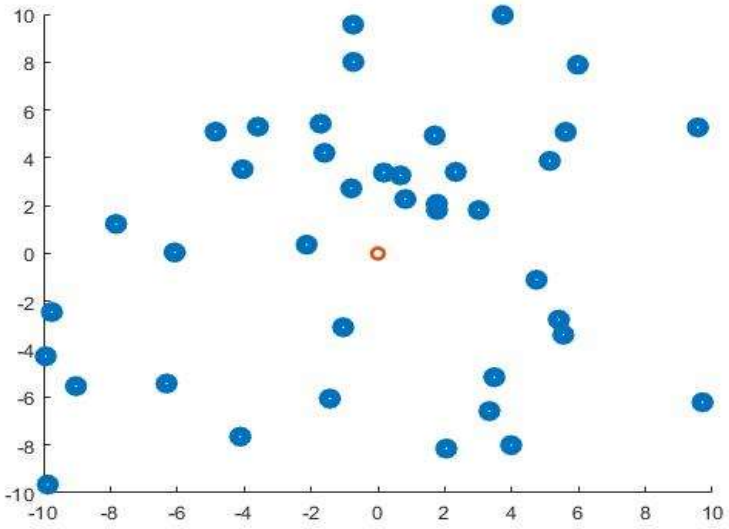
- وارد حلقه اصلی الگوریتم میشویم.همانطور که گفته شده است قرار است در هر تکرار,هر کرم شب تاب با کرم شب تاب های دیگر مقایسه شود و با توجه به شرایط و روابط گفته شده به سمت کرم شب تاب جذاب تر از خودش حرکت کند.با این حرکت موقعیت و ارزش کرم شب تاب تغییر میکند و ما این موقعیت جدید و ارزش جدید را در ساختار newFirefly ذخیره میکنیم.(خطوط 54 تا 62)
- در مورد 56 و 57 و 58 که روابط مربوطه را تعریف کرده ایم.در مورد خط 58 که بردار تصادفی را ایجاد کرده ایم باید گفت که با استفاده از توزیع نورمال انجام شده است.(متغیر دلتا در این قسمت استفاده شده است)
- در خطوط 63 و 64 با این کار فقط اطمینان حاصل میکنیم که موقعیت کرم از ماکسیمم و مینیمم مقدار تعریف شده بیشتر یا کمتر نشود.
- در خطوط بعدی با چک کردن جواب(در صورت با ارزش بودن)در ساختار bestSol ذخیره کرده ایم.
- ما الان سه ساختار از موقعیت و ارزش کرم ها داریم: (Firefly , newFirefly , bestSol).در تکرار بعدی نیازمند ساختاری یکپارچه هستیم و برای اینکه هر بار به جواب نزدیک تر شویم,ساختار Firefly را آپدیت میکنیم و هر سه ساختار را به داخل این ساختار انتقال میدهیم.(خط 72) ولی برای نزدیک تر شدن به جواب داده های موجود در ساختار Firefly آپدیت شده را منظم میکنیم(خط 73 و 74) و فقط به تعداد کرم شب تاب ها از اول این ساختار برمیداریم (خط 75) و با این Firefly جدید وارد تکرار بعدی میشویم و در هر تکرار این کار هارا انجام میدهیم.
- جواب نهایی هر تکرار همانطور که گفته شد در ساختار bestCost ذخیره میشود(خط 77)
- در خط 80 به صورت نمایی مقدار آلفا را کاهش میدهیم تا هر بار کوچک تر شود.

مرحله 5 : نمایش نتایج

```
 87 —    figure(1);
 88 —    semilogy(bestCost,'LineWidth',2);
 89 —    xlabel('It');
 90 —    ylabel('Best Cost');
 91
 92 —    bb = zeros(1,nPop);
 93 —    cc = zeros(1,nPop);
 94 —    dd = zeros(1,nPop);
 95 —    ee = zeros(1,nPop);
 96 —    for i=1:nPop
 97 —        bb(i) = copyOfFirefly(i).pos(1);
 98 —        cc(i) = copyOfFirefly(i).pos(2);
 99 —        dd(i) = Firefly(i).pos(1);
100 —        ee(i) = Firefly(i).pos(2);
101 —    end
102 —    figure(2);
103 —    hold on
104 —    scatter(bb,cc,'LineWidth',5);
105 —    scatter(dd,ee,'LineWidth',2);
106 —    hold off
```

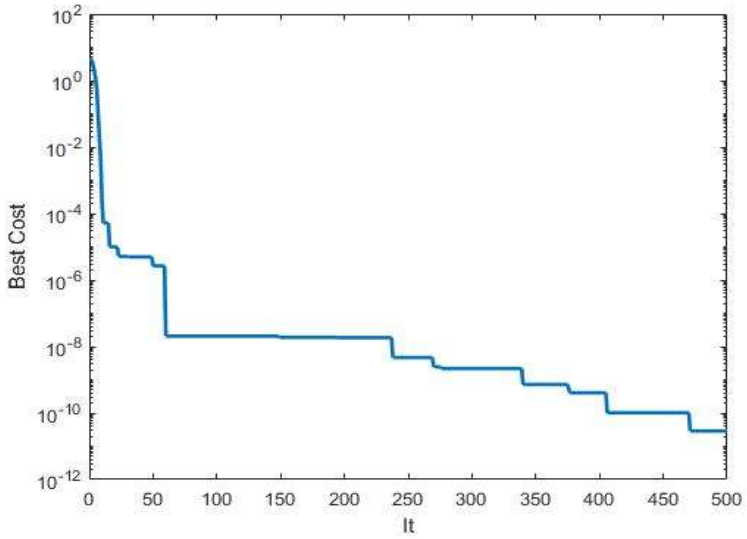
- در خطوط 96 تا 106 با تفکیک موقعیت های کرم شب تاب ها,در شکل 2 موقعیت اولیه کرم شب تاب ها (رنگ آبی) و همچنین موقعیت نهایی کرم شب تاب ها بعد از پایان یافتن تکرار ها (رنگ نارنجی) نشان داده شده است

خروجی کار



پراکندگی جمعیت اولیه کرم شب تاب (آبی)

تجمع نهایی جمعیت کرم شب تاب (نارنجی)



منابع

- **Iztok Fister, Xin-She Yang, Janez Brest and Iztok Fister Jr.On the Randomized Firefly Algorithm**