



عنوان:

# الگوریتم جهش قورباغه

Shuffled Frog Leaping Algorithm (SFLA)

ارائه دهنده:

فاطمه صالح نیا



۶. فلوجارت  
الگوریتم

۱. تعریف  
الگوریتم  
ممتیک

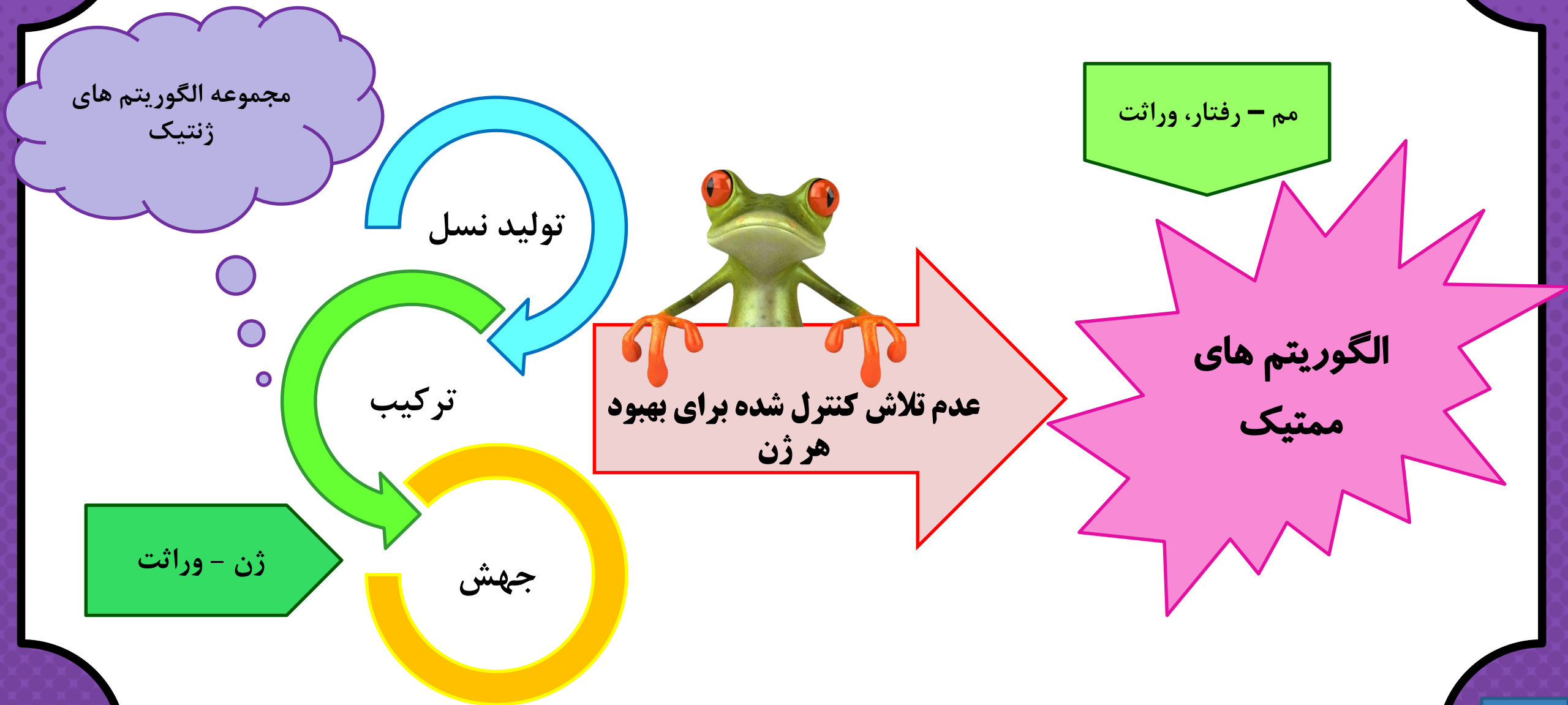
۵. خلاصه گام های  
کلی الگوریتم

فهرست

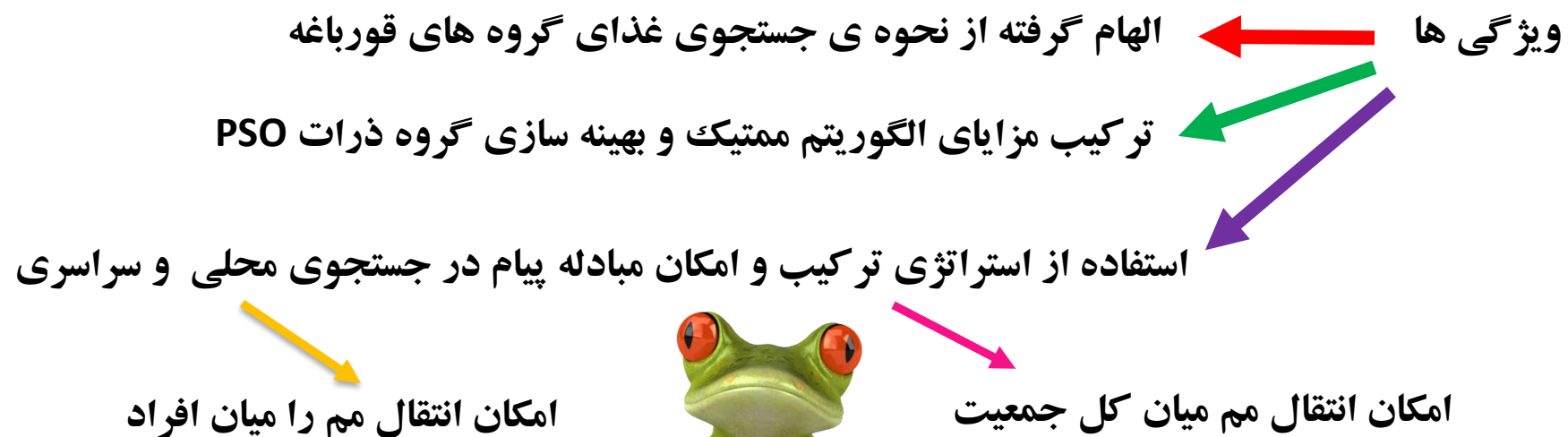
۲. معرفی  
الگوریتم

۴. الگوریتم  
محلی بهینه  
سازی FLA

۳. معرفی  
الگوریتم در فاز  
جستجوی  
سراسری



## معرفی الگوریتم



توازن بین مبادله پیام سراسری و جستجوی محلی ← امکان پرش از مینیمم محلی و دستیابی به حالت بهینه یکی از خصیصه های الگوریتم SFLA همگرایی سریع آن است.

# الگوریتم SFLA

Shuffled Frog Leaping Evolution

انتخاب تصادفی

تعداد مجتمع ها:

m

S

۱	بهترین
۲	.
۳	.
مم های جمعیت اولیه	
.	
.	
S	بدترین

۱
۴
مجتمع ۱

۲
۵
مجتمع ۲

۳
مجتمع ۳

اندازه هر مجتمع

$$S = M * N$$

۱. تعیین پارامترهای اولیه

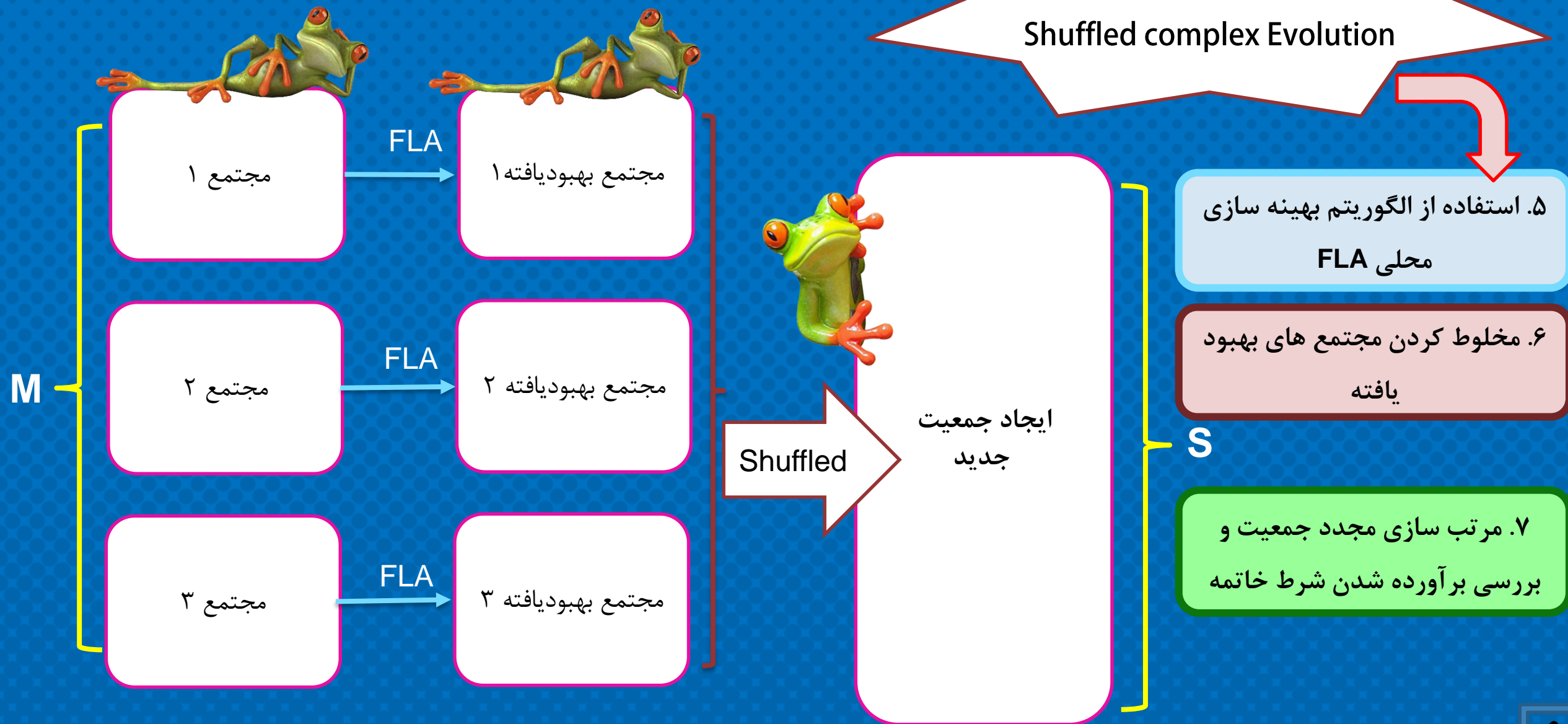
۲. تولید جمعیت اولیه به صورت تصادفی

۳. مرتب سازی پاسخ ها بر حسب fitness

۴. تقسیم جمعیت به چندین مجتمع



# الگوریتم SCE-UA Shuffled complex Evolution







# الگوریتم بهینه سازی محلی FLA:

هدف قورباغه ها: بهبود  
مم هایشان به سمت  
موقعیت های بهینه



ایجاد  
زیرمجموعه ها:

از  $n$  قورباغه ی موجود در مجتمع، انتخاب  $q$  قورباغه  
به صورت تصادفی

مرتب سازی  
قورباغه ها

مرتب سازی قورباغه ها براساس میزان شایستگی به  
صورت نزولی  
بهترین  $P_B$  و بدترین  $P_W$  موقعیت

تصحیح موقعیت  
بدترین قورباغه

تعیین میزان جهش از طریق رابطه ۱  
اصلاح موقعیت از طریق رابطه ۲ با شرط بهتر بودن  
موقعیت جدید

به روز  
کردن  
مجتمع

جایگزینی با بدترین قورباغه برمی گردد، بررسی  
شرط خاتمه (اینکه چند فرزند بسازیم؟!)

والدها

مجتمع  
با اندازه  
 $n$

$P_i$

$q$

+

-

۱

$$S = \text{rand}() * (P_B - P_W)$$

۲

$$U_q = P_W + S$$

در صورتی که موقعیت بهتر  
نشود، به جای  $P_B$  مقدار  $P_G$   
را قرار می دهیم.

جایگزینی تصادفی یک قورباغه  
جدید بجای قورباغه ای که برای  
پیشروی مناسب نیست



## مثالی از بهبود موقعیت بدترین قورباغه:

[0,1]

۱

$$S = \text{rand}() * (P_B - P_W)$$

۲

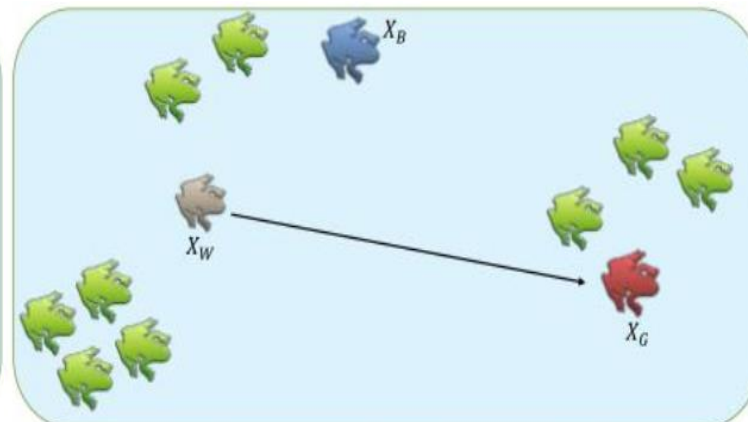
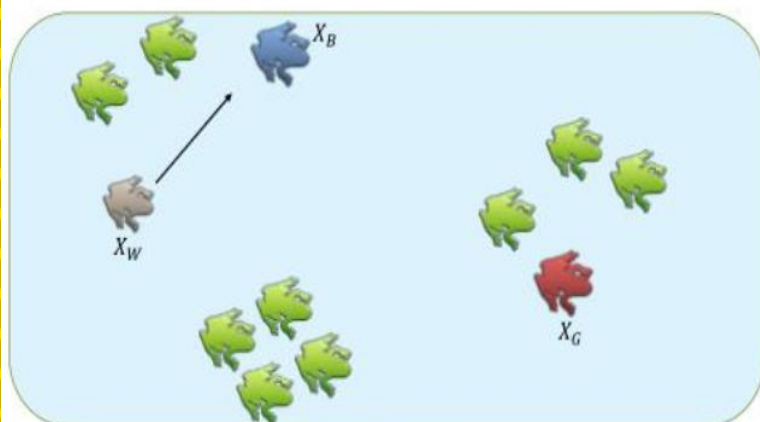
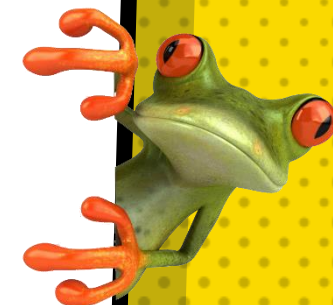
$$U_q = P_W + S$$

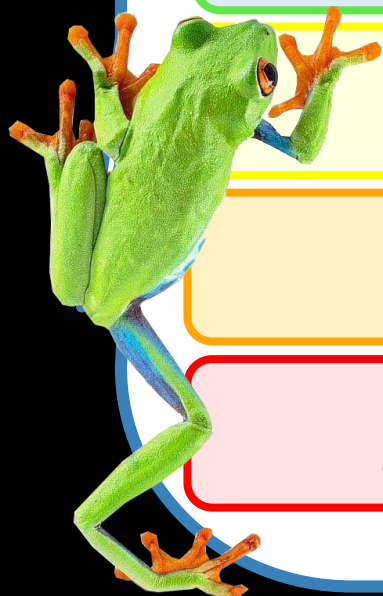
best	3	2	6	8	9	4
------	---	---	---	---	---	---

worst	1	5	2	4	7	3
-------	---	---	---	---	---	---



new	3	5	6	4	9	4
-----	---	---	---	---	---	---





## ۱- مقداردهی اولیه

انتخاب  $M$  تعداد مجتمع ها و  $n$  تعداد قورباغه های هر مجتمع

## ۲- تولید جمعیت

انتخاب جمعیت تصادفی به اندازه  $f$  از کل جمعیت، محاسبه مقدار شایستگی هر قورباغه

## ۳- درجه بندی قورباغه ها

مرتب سازی قورباغه ها به ترتیب نزولی براساس میزان شایستگی  
ثبت موقعیت بهترین قورباغه  $P_x$  در کل جمعیت

## ۴- تقسیم قورباغه ها در مجتمع ها

تقسیم جمعیت فعلی در  $m$  مجتمع، هر کدام شامل  $n$  قورباغه

## ۵- تکامل ممیتک در هر مجتمع

تکامل هر مجتمع، به وسیله ی جستجوی محلی (FLA)

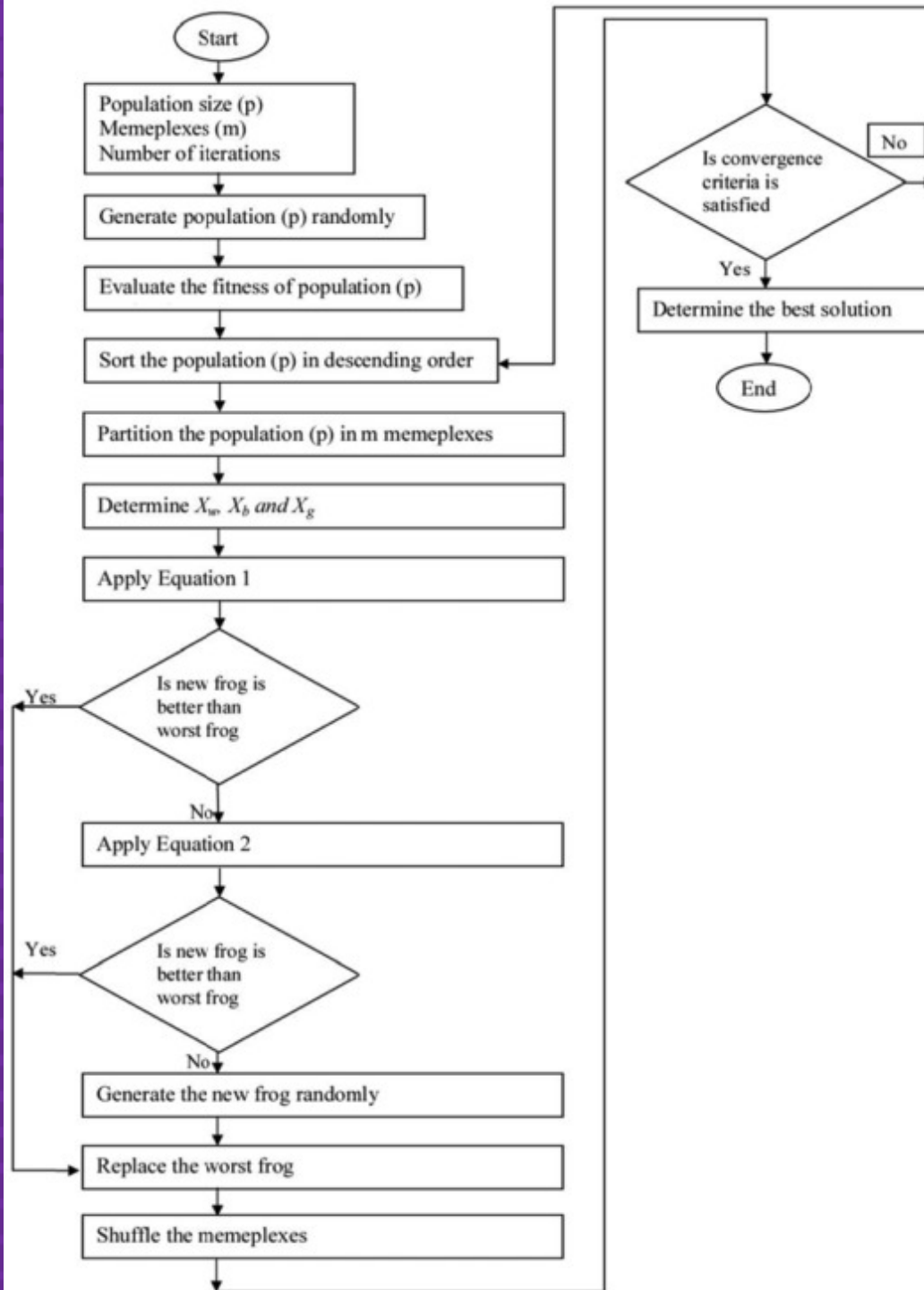
## ۶- ترکیب مجتمع ها

پس از تعداد معینی تکامل ممیتک در هر مجتمع، ترکیب مجدد مجتمع ها  
و بروزرسانی موقعیت بهترین قورباغه ی موجود در جمعیت ( $P_x$ )

## ۷- بررسی شرایط

بررسی شرایط اتمام الگوریتم:  
عدم برآورده شدن شرایط، رفتن به گام ۴ - در صورت اتمام  $P_G$  جواب بهینه بدست آمده

# فلو چارت الگوریتم جهش قورباغه



## منابع

- Eusuff, Muzaffar, Kevin Lansey, and Fayzul Pasha. "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization." Engineering Optimization 38.2 (2006): 129-154.
- Fang, Chen, and Ling Wang. "An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem." Computers & Operations Research 39.5 (2012): 890-901.
- <http://matlabfile.ir/687/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%A%D9%85-%D8%AC%D9%87%D8%B4-%D9%82%D9%88%D8%B1%D8%A8%D8%A7%D8%BA%D9%87-%D8%A7%DB%8C/>
- [https://www.youtube.com/watch?v=uDC\\_NxCE\\_38](https://www.youtube.com/watch?v=uDC_NxCE_38)







با سپاس از توجه شما

## روند الگوریتم SFLA

```
Begin;  
  Generate random population of  $P$  solutions (frogs);  
  For each individual  $i \in P$ : calculate fitness ( $i$ );  
  Sort the population  $P$  in descending order of their  
  fitness;  
  Divide  $P$  into  $m$  memeplexes;  
  For each memeplex;  
    Determine the best and worst frogs;  
    Improve the worst frog position using Eqs. (1) and (2);  
  Repeat for a specific number of iterations;  
End;  
Combine the evolved memeplexes;  
Sort the population  $P$  in descending order of their  
fitness;  
Check if termination = true;  
End;
```

