

## مسائل في الخوارزميات الجينية

### المسألة الأولى:

لدينا وحدة انتاج تتالف من ثلاث ورشات تنتج كل ورشة منتج مختلف، المطلوب تحديد خطة او جدول الإنتاج لهذه الورشات لزيادة الربح، لتبسيط المسألة افترض ان كل ورشة اما ان تنتج كمية محددة من المنتجات (هذه الحالة يمكن ان تمثل بالرقم الثنائي 1) او لا تنتج شيء (تمثل بالرقم 0)، اذا يمثل جدول الإنتاج عندئذ بسلسلة مؤلفة من ثلاث بتات، على سبيل المثال:

(101) means (make,not make,make) This may yield a total profit of 5

بفرض كل حل يمثل يسلسلة من ثلاث بتات وحجم التجمع السكاني (population) هو 4 و بفرض ان قيمة اللياقة للحل هي العدد الثنائي الممثل بالسلسلة، اوجد الحل الذي يعطي اكبر قيمة لياقة أي 111

الحل:

#### Step 0 : initialization of the population

101, 001, 010, 110

#### Iteration1, Step1 : parent selection

(a) يتم الاختيار عن طريق تحديد اللياقة لكل حل أي لكل سلسلة ( $A_i$ ) ومن ثم اللياقة الكلية، الجدول التالي يوضح ذلك:

i	$A_i$	$f_i$
1	101	5
2	001	1
3	010	2
4	110	6

$$\text{Total fitness} = F = \sum_{i=1}^4 f_i = 14$$

ثم نقوم بحساب احتمال اللياقة وتوقع الإحصاء ويعطى كل منهما بالعلاقات التالية:

$$\text{Fitness probability} = p_i = \frac{f_i}{\sum_{i=1}^n f_i}$$

$$\text{Expected count} = n \cdot p_i = \frac{f_i}{\text{Avg}}$$

$i$	$A_i$	$f_i$	$p_i$	$n \cdot p_i$	Actual count
1	101	5	0.357	1.429	?
2	001	1	0.071	0.286	?
3	010	2	0.143	0.571	?
4	110	6	0.429	1.714	?
	total	14	1.000	4	
	Avg	3.5	0.25	1	
	Max	6	0.429	1.714	

(b) نختار عشوائياً مجموعة جديدة مؤلفة من 4 سلاسل

Random No	Probability	String $i$
(000 356)	0.357	1
(357 427)	0.071	2
(428 570)	0.143	3
(571 999)	0.429	4

بفرض أننا قمنا بتوليد 4 أرقام عشوائية 725,652,091,438 فإن الحلول المختارة هي 4,4,1,3

$i$	$A_i$	$n \cdot p_i$	Actual count
1	101	1.429	1
2	001	0.286	0
3	010	0.571	1
4	110	1.714	2

وبالتالي فان الإباء التي تم اختيارها تكون كما هو موضح في الجدول

Newi	$A_i$
1	010
2	101
3	110
4	110

### Iteration1, Step2 : Crossover

a. يتم اختيار زوج من الآباء بشكل عشوائي للتزاوج حيث يتم تحديد فيما اذا ستحدث عملية عبور كروموسومي بناء على قيمة احتمال محدد مسبقا. سنفترض فيما يلي انه ستحصل عملية عبور.

b. يتم اختيار نقطة العبور  $k$  عشوائيا لكل سلسلة في المجال  $[1, L-1]$  حيث  $L$  هي طول السلسلة ، حيث يتم تبادل أجزاء السلسلتين عند نقطة العبور.

فعلى سبيل المثال السلسلتين التاليتين قد يتم اختيارهما بشكل عشوائي كما في الخطوة a

$$A_2=101$$

$$A_3=110$$

b: قد يتم اختيارها عشوائيا كما في الخطوة  $k=2$  بعد ذلك

$$A_2=101$$

$$A_3=110$$


---


$$A'_2=100$$

$$A'_3=111$$

السلسلتين الباقيتين يتم التزاوج بينهما مع اختيار  $k=1$  كنقطة عبور عشوائية حيث يتم توليد سلسلتين جديدتين:

$$A'_1=010$$

$$A'_4=110$$

i	$A_i$	Crossover site,k	New population $A'_i$
1	010	1	010
2	101	2	100
3	110	2	111
4	110	1	110

### Iteration1, Step3: Mutation

تحصل الطفرة بشكل نادر. على سبيل المثال تحصل طفرة واحدة لكل 1000 عملية نقل للبتات. في مثالنا فقط 12 بت تم نقلها من جيل لجيل وبالتالي لا تحصل عملية الطفرة حيث انه هنا احتمال حصول الطفرة لهذا الدور حيث الاحتمال هو:  $12/1000 = 0.012$

i	$A_i$	$f_i$
1	010	2
2	100	4
3	111	7
4	110	6
	Total	19
	Avg	4.75
	Max	7

نلاحظ حصول تحسن من الجيل الأول الى هذا الجيل على اللياقة الاجمالية والمتوسطة والأعظمية، بما أن الحل الأمثل الموافق لقيمة اللياقة الاعظمية 7 قد تم الحصول عليه فلا حاجة لاجراء دور آخر.

## المسألة الثانية: Genetic Algorithm for Hello world!

هو مثال للتسلية حيث يبدأ ببضع مئات من العبارات العشوائية بإتجاه عبارة Hello world!

Constraction the chromosome:

[97,98,99]  [a,b,c]

**The fitness function:**

نريد العبارات الأقرب إلى Hello world! أن تكون لياقتها أعلى ,

يوجد لدينا عدة طرق من أجل تحديد تابع اللياقة منها:

1. إرجاع عدد المحارف من الكروموسوم والتي تكون موجودة في العبارة الهدف أي :

"ehllo world!" أكثر لياقة من "Gello world!" ولكنها تماثل

لياقة "Hello world!" .

2. إرجاع عدد المحارف من الكروموسوم والتي تقع في المكان الصحيح أي :

"Hxxloxxwxxxx" لياقتها تساوي 4 و "Hello world!" لياقتها 12.

3. إرجاع مجموع بعد المحارف في الحل عنها في الحل المثالي أي:

"Hello Xord!" لديها لياقة 1- لأن X تبعد بمقدار محرف واحد

عن w بينما "Hello yord!" لديها لياقة 2- .