

Ewolucja recesywów dla haploidów

$$p^{t+1} = (1 - \varepsilon)p^t + p_m(1 - p^t) - p_m p^t$$
$$p^{t=\infty} \sim p_m$$

Dla diploidów:

$$p^{t+1} = (1 - 2\varepsilon p^t)p^t + 2p_m(1 - p^t) - 2p_m p^t$$

Maskowanie!

W przybliżeniu że populacja recesywów się nie zmienia,

$$\sqrt{p_m} \sim p^\infty$$

Niestandardowe operatory genetyczne

1. Operator duplikacji

$$D_b[A, b, c, D, E] = [A, b, b, c, D, E]$$

mutacja teraz działa 2x lepiej dla cechy bB

2. Operator delecji (odwrotny do poprzedniego):

$$\bar{D}_b[A, b, b, c, d, E] = [A, b, c, d, E]$$

3. Operator inwersji: zmiana kolejności genów
4. Inwersja liniowa - losujemy w chromosomie dwa punkty, np. tu: 1 i 4 (indeksy punktów) $[A, b, c, D, e, F] \rightarrow [A, e, D, c, b, F]$

```
x, y = np.random.randint(0, len(genotype), shape=2)
genotype[x:y] = genotype[x:y:-1]
```
5. Inwersja boczna $[A, b, c, D, e, F] \rightarrow [A, c, D, F, e]$
6. Inwersja liniowo-boczna Z prawdopodobieństwem np. 0.75 liniowa, z 0.25 boczna

Cała idea to konserwacja cegiełek w genotypie.

Problem: krzyżujemy

ABc|De
AbE|dc

ABcdC - brak eE
AbEDe - brak cC

oba nazywamy nieżywoтными. Żeby im zapobiec, stosujemy *kopiowanie wg wzorca*: Tworzymy wpierw jakiś wzorzec kolejności genów. Geny w obu chromosomach ustawiamy wg tego wzorca. Wzorcem często jest osobnik o większej funkcji celu.

Pożytki z płci w algorytmach genetycznych

- specjalizacja
- podział populacji na grupy, które obie wykonują zadania
- dwie czynności konieczne do tworzenia potomstwa:

1. h - hunting
2. n - nursing

- $S(n, h) = h + n$ zła bo może być $h = 0$ i dla dużych n dalej ok $S(n, h) = hn$
lepsza $S(n, h) = h + n + ahn = 1$ - unormowanie $h(1 + ahn) = 1 - n$
 $h = \frac{1-n}{1+an}$

$$S = n \frac{1 - n}{1 + an}$$

co przyjmuje wartość maksymalną dla $a = 0, n = 0.5$. Dla niezerowego a mamy $n_{\pm} = (-1 \pm \sqrt{1 + a})/a$, a w granicy