

— Algoritmi Avansați 2023

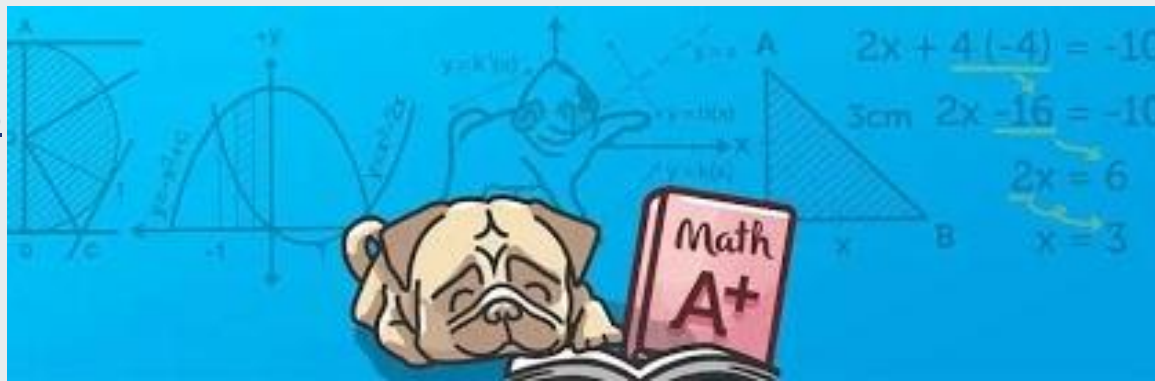
C-11

Genetic Algorithms

Lect. Dr. Ștefan Popescu

Email: stefan.popescu@fmi.unibuc.ro

Grup Teams:



What are they used for?

Sunt utilizați în probleme de optim, pentru care

- spațiul de căutare a soluțiilor posibile este mare
- nu se cunosc algoritmi exacți mai rapizi Furnizează o soluție care nu este neapărat optimă.
- Căutarea în spațiul soluțiilor candidat – euristică, bazată pe principii ale evoluției în genetică

Denumirea lor se datorează preluării unor mecanisme din biologie: moștenirea genetică și evoluția naturală pentru populații de indivizi



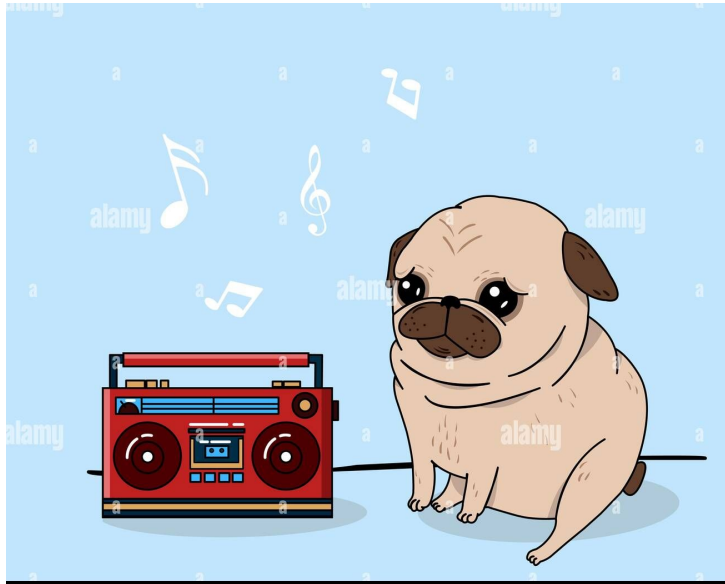
What are they used for?

Aplicații

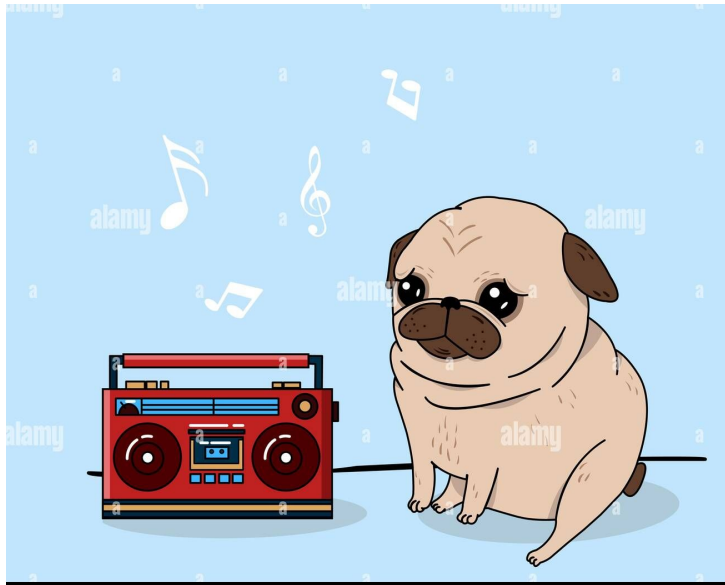
- Robotică, bioinformatică, inginerie
- Probleme de trafic, rutare, proiectare
- Criptare, code-breaking
- Teoria jocurilor
- Clustering etc



Informal talk



Informal talk



Exemplu ilustrativ (scop didactic)



Maximul unei funcții pozitive

Fie $f:D \rightarrow \mathbb{R}$. Să se calculeze

$$\max\{ f(x) \mid x \in D \}, \text{ unde } D = [a, b].$$

- Presupunem $f(x) > 0, \forall x \in D$.

Algoritmi Genetici: Noțiuni

Cromozom = mulțime ordonată de elemente (gene) ale căror valori (alele) determină caracteristicile unui individ

1	0	1	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---



Algoritmi Genetici: Noțiuni

Cromozom = mulțime ordonată de elemente (gene) ale căror valoare (alele) determină caracteristicile unui individ

1	0	1	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Populație = mulțime de indivizi care trăiesc într-un mediu la care trebuie să se adapteze



Algoritmi Genetici: Noțiuni

Cromozom = mulțime ordonată de elemente (gene) ale căror valoare (alele) determină caracteristicile unui individ

1	0	1	1	1	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---

Populație = mulțime de indivizi care trăiesc într-un mediu la care trebuie să se adapteze

Fitness (adecvare) = măsură a gradului de adaptare la mediu pentru fiecare individ (funcție de fitness)



Algoritmi Genetici: Noțiuni

Generație = etapă în evoluția populației



Algoritmi Genetici: Noțiuni

Generație = etapă în evoluția populației

Selecție = proces prin care sunt promovați indivizii cu grad ridicat de adaptare la mediu



Algoritmi Genetici: Noțiuni

Generație = etapă în evoluția populației

Selecție = proces prin care sunt promovați indivizii cu grad ridicat de adaptare la mediu

Operatori genetici:

- **încrucișare** (combinare, crossover) - indivizii din noua generație moștenesc caracteristicile părinților



Algoritmi Genetici: Noțiuni

Generație = etapă în evoluția populației

Selecție = proces prin care sunt promovați indivizii cu grad ridicat de adaptare la mediu

Operatori genetici:

- **încrucișare** (combinare, crossover) - indivizii din noua generație moștenesc caracteristicile părinților
- **mutație** - indivizii din noua generație pot dobândi și caracteristici noi



Structura pas-cu-pas a unui Algoritm Genetic



Algoritm

- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D

- $t=t+1$



Algoritm

- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D
- Cât timp nu există condiția de terminare:

- $t=t+1$



Condiții de terminare

- număr maxim de iterații / durată de execuție
- stabilizarea performanței medii /maxime
- am obținut o soluție suficient de bună



Algoritm

- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D
- Cât timp nu există condiția de terminare:
 - construim o populație nouă $P(t+1)$ pe baza indivizilor din $P(t)$ astfel:
 - selecție: generează o populație intermediară $P^1(t)$ selectând indivizi din $P(t)$ după un anumit criteriu de selecție
- $t=t+1$



Algoritm

- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D
- Cât timp nu există condiția de terminare:
 - construim o populație nouă $P(t+1)$ pe baza indivizilor din $P(t)$ astfel:
 - selecție: generează o populație intermediară $P^1(t)$ selectând indivizi din $P(t)$ după un anumit criteriu de selecție
 - aplicăm operatorul de încrucișare pentru (unii) indivizi din $P^1(t)$ obținând populația intermediară $P^2(t)$
- $t=t+1$



Algoritm

- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D
- Cât timp nu există condiția de terminare:
 - construim o populație nouă $P(t+1)$ pe baza indivizilor din $P(t)$ astfel:
 - selecție: generează o populație intermediară $P^1(t)$ selectând indivizi din $P(t)$ după un anumit criteriu de selecție
 - aplicăm operatorul de încrucișare pentru (unii) indivizi din $P^1(t)$ obținând populația intermediară $P^2(t)$
 - aplicăm operatorul de mutație peste (unii) indivizi din $P^2(t)$ obținând populația $P(t+1)$
- $t=t+1$



Algoritm

- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D
- Cât timp nu există condiția de terminare:
 - construim o populație nouă $P(t+1)$ pe baza indivizilor din $P(t)$ astfel:
 - **selecție**: generează o populație intermediară $P^1(t)$ selectând indivizi din $P(t)$ după un anumit **criteriu de selecție**
 - aplicăm **operatorul de încrucișare** pentru (unii) indivizi din $P^1(t)$ obținând populația intermediară $P^2(t)$
 - aplicăm operatorul de mutație peste (unii) indivizi din $P^2(t)$ obținând populația $P(t+1)$
 - *opțional: la $P(t+1)$ se adaugă elementul/elementele elitiste din $P(t)$*
- $t=t+1$



Algoritm

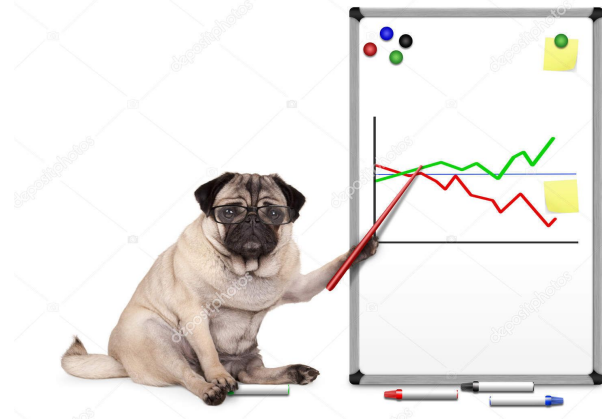
- $t=0$
- Consideră, o populație inițială $P(0)$: alegem aleator indivizi din intervalul D
- Cât timp nu există condiția de terminare:
 - construim o populație nouă $P(t+1)$ pe baza indivizilor din $P(t)$ astfel:
 - **selecție**: generează o populație intermediară $P^1(t)$ selectând indivizi din $P(t)$ după un anumit **criteriu de selecție**
 - aplicăm **operatorul de încrucișare** pentru (unii) indivizi din $P^1(t)$ obținând populația intermediară $P^2(t)$
 - aplicăm operatorul de mutație peste (unii) indivizi din $P^2(t)$ obținând populația $P(t+1)$
 - *opțional: la $P(t+1)$ se adaugă elementul/elementele elitiste din $P(t)$*
- $t=t+1$



Exemplu: maximizarea unei funcții pozitive

Date de intrare + parametri de control

- intervalul $[a, b]$
- precizia p (numărul de zecimale)
- dimensiunea populației n
- numărul de generații
- probabilitatea de încrucișare p_c
- probabilitatea de mutație p_m



Populația

Dimensiune (număr de cromozomi) :

n - fixă, dată

constantă pe parcursul algoritmului



Populația

Dimensiune (număr de cromozomi) :

n - fixă, dată

constantă pe parcursul algoritmului

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom

În general: codificare binară, lungime fixă



Populația

Dimensiune (număr de cromozomi) :

n - fixă, dată

constantă pe parcursul algoritmului

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom

În general: codificare binară, lungime fixă

Cum calculăm lungimea pentru puncte din $D = [a,b]$?



Populația

Dimensiune (număr de cromozomi) :

n - fixă, dată

constantă pe parcursul algoritmului

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom

În general: codificare binară, lungime fixă

Cum calculăm lungimea pentru puncte din $D = [a,b]$?

Depinde de nivelul de discretizare al intervalului $[a,b]$



Codificare

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom
În general: codificare binară, lungime fixă



Pentru $D = [a, b]$ și o precizie p dată (ca număr de zecimale):

- discretizarea intervalului =>



subintervale (elemente)

Codificare

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom
În general: codificare binară, lungime fixă

Pentru $D = [a, b]$ și o precizie p dată (ca număr de zecimale):

- discretizarea intervalului $\Rightarrow (b-a) \times 10^p$ subintervale (elemente)



Codificare

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom
În general: codificare binară, lungime fixă



Pentru $D = [a, b]$ și o precizie p dată (ca număr de zecimale):

- discretizarea intervalului $\Rightarrow (b-a) \times 10^p$ subintervale (elemente)
- lungimea cromozomului este:



Codificare

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom
În general: codificare binară, lungime fixă



Pentru $D = [a, b]$ și o precizie p dată (ca număr de zecimale):

- discretizarea intervalului $\Rightarrow (b-a) \times 10^p$ subintervale (elemente)
- lungimea cromozomului este:

$$2^{l-1} < (b-a)10^p \leq 2^l \Rightarrow l = \lceil \log_2((b-a)10^p) \rceil$$

Codificare

Codificare = cum asociem unei configurații din spațiul de căutare un cromozom
În general: codificare binară, lungime fixă



Pentru $D = [a, b]$ și o precizie p dată (ca număr de zecimale):

- discretizarea intervalului $\Rightarrow (b-a) \times 10^p$ subintervale (elemente)
- lungimea cromozomului este:

$$2^{l-1} < (b-a)10^p \leq 2^l \Rightarrow l = \lceil \log_2((b-a)10^p) \rceil$$

- valoarea codificată din $D=[a, b]$ - translație liniară

$$X_{(2)} \rightarrow X_{(10)} \rightarrow \frac{b-a}{2^l-1} X_{(10)} + a$$

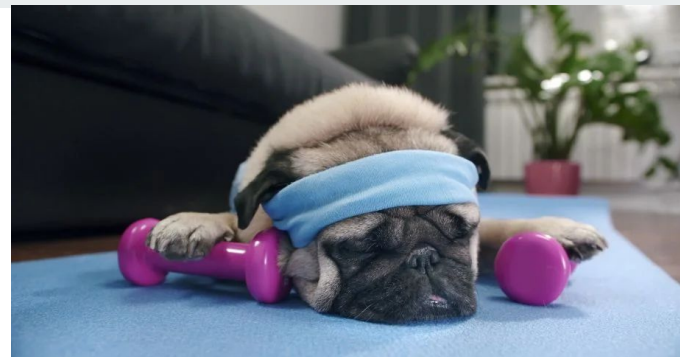
Populație

Populația inițială se generează aleator



Funcția de fitness

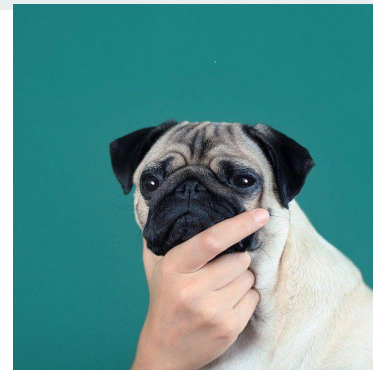
- se pot folosi distanțe cunoscute (euclidiană, Hamming)
- pentru problema de maxim funcția este chiar f



Selecția

determinarea unei populații intermediare, ce conține indivizi care vor fi supuși operatorilor genetici

- Selecție proporțională
- Selecție elitistă
- Selecție turneu
- Selecție bazată pe ordonare



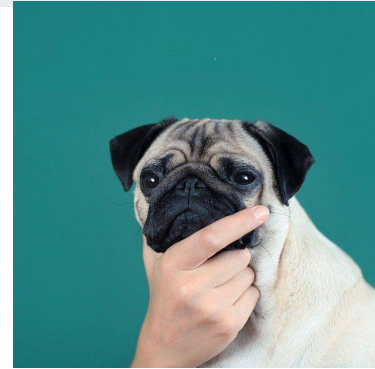
Selecția proporțională

Presupunem $P(t) = \{X_1, \dots, X_n\}$

asociem fiecărui individ X_i o probabilitate p_i de a fi selectat, în funcție de performanța acestuia (dată de funcția de fitness f)



VERSIRIS



Selecția proporțională

Presupunem $P(t) = \{X_1, \dots, X_n\}$

asociem fiecărui individ X_i o probabilitate p_i de a fi selectat, în funcție de performanța acestuia (dată de funcția de fitness f)

$$p_i = \frac{f(X_i)}{F}$$

$$F = \sum_{j=1}^n f(X_j) = \text{performanța totală a populației}$$



Selecția proporțională

Presupunem $P(t) = \{X_1, \dots, X_n\}$

asociem fiecărui individ X_i o probabilitate p_i de a fi selectat, în funcție de performanța acestuia (dată de funcția de fitness f)

$$p_i = \frac{f(X_i)}{F}$$

$$F = \sum_{j=1}^n f(X_j) = \text{performanța totală a populației}$$

folosind **metoda ruletei** selectăm n indivizi (!copii), cu distribuția de probabilitate (p_1, p_2, \dots, p_n)

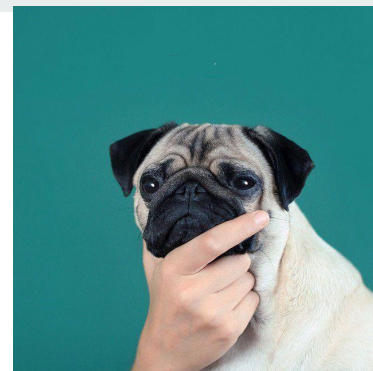


Selecția proporțională - metoda ruletei

Folosind metoda ruletei selectăm n indivizi (!copii), cu distribuția de probabilitate (p_1, p_2, \dots, p_n)

Etapă de Selecție:

- $P^1(t) = \emptyset$
- repetă de n ori:
 - generează j cu probabilitatea (p_1, p_2, \dots, p_n) folosind metoda ruletei
 - generează u variabilă uniformă pe $[0,1)$
 - determină indicele j astfel încât u este între $q_{j-1} = p_1 + \dots + p_{j-1}$ și $q_j = p_1 + \dots + p_j$ (cu convenția $q_0 = 0$)
 - adaugă la populația selectată $P^1(t)$ o copie a lui X_j

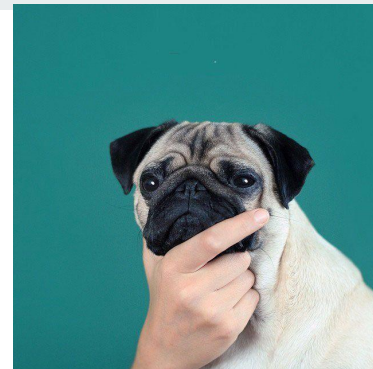


Selecția

Selecție elitistă = trecerea explicită a celui mai bun individ în generația următoare

Selecție turneu = se aleg aleatoriu k indivizi din populație și se selectează cel mai performant dintre ei

Selecție bazată pe ordonare = se ordonează indivizii după performanță și li se asociază câte o probabilitate de selecție în funcție de locul lor după ordonare

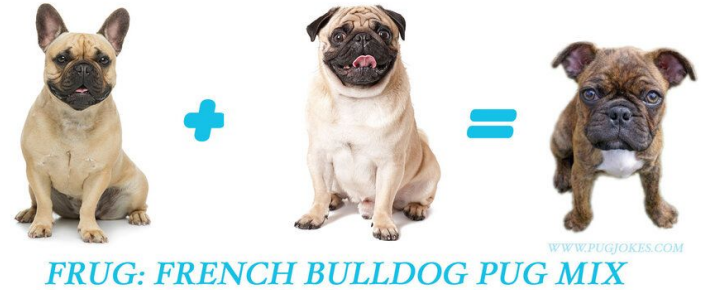


Încrucișarea

Permite combinarea informațiilor de la părinți

Doi părinți dau naștere la doi descendenți

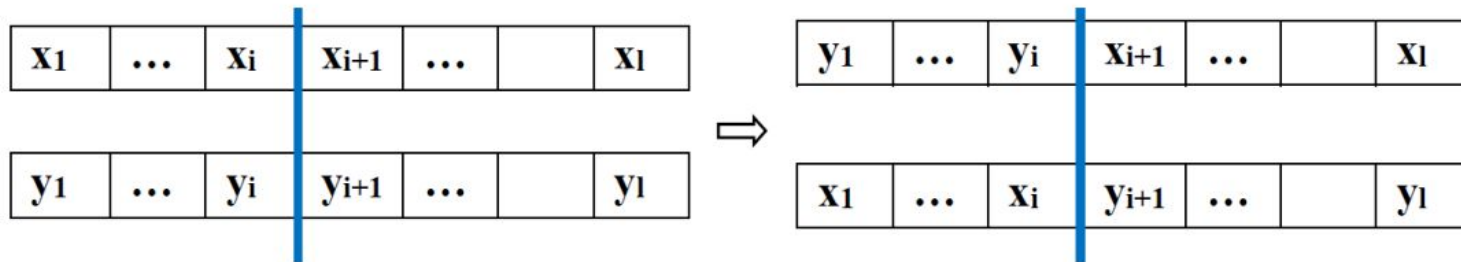
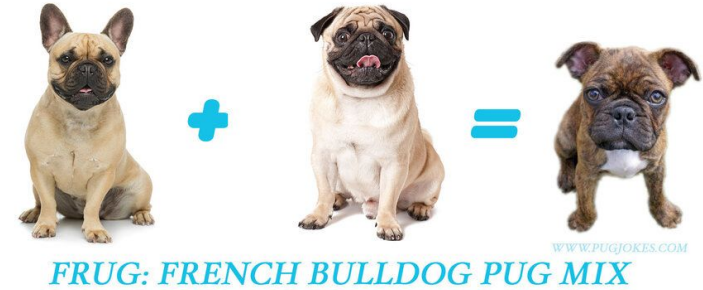
- cu un punct de tăietură (de rupere)
- cu mai multe puncte de rupere
- uniformă
- etc



Încrucișarea

Cu un punct de tăietură (de rupere)

2 părinți => 2 indivizi noi care iau locul părinților în populație



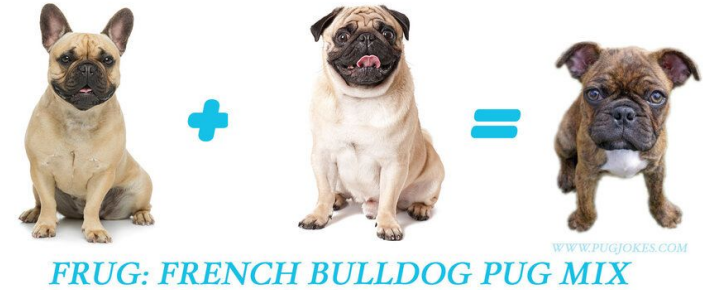
i – punct de rupere generat aleator

Încrucișarea

Cu un punct de tăietură (de rupere)

Nu toți cromozomii din $P^1(t)$ participă la încrucișare.

Un cromozom participă la încrucișare cu o probabilitate fixată p_c (probabilitate de încrucișare – dată de intrare)



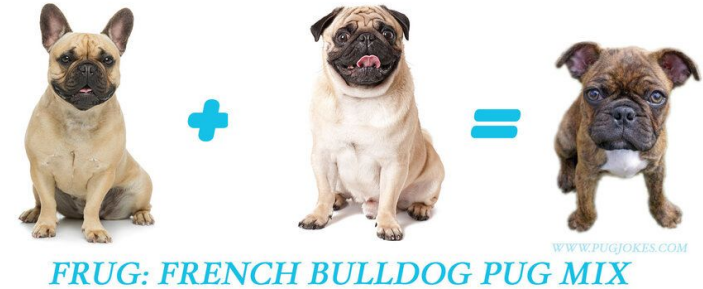
Încrucișarea

Cu un punct de tăietură (de rupere)

Un cromozom participă la încrucișare cu o probabilitate fixată p_c (probabilitate de încrucișare – dată de intrare)

Etapă de încrucișare:

- Notăm $P^1(t) = \{X_1, X_2, \dots, X_n\}$



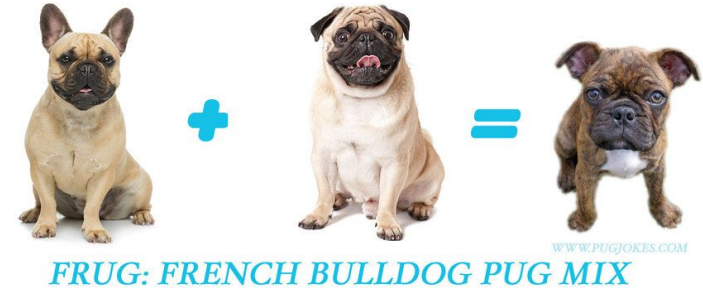
Încrucișarea

Cu un punct de tăietură (de rupere)

Un cromozom participă la încrucișare cu o probabilitate fixată p_c (probabilitate de încrucișare – dată de intrare)

Etapă de încrucișare:

- Notăm $P^1(t) = \{X_1, X_2, \dots, X_n\}$
- for $i = 1, n$
 - generează u variabilă uniformă pe $[0, 1]$



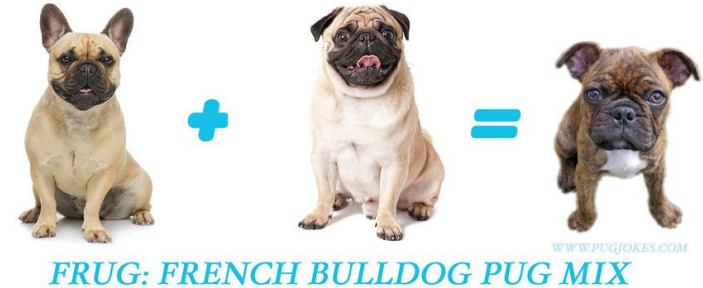
Încrucișarea

Cu un punct de tăietură (de rupere)

Un cromozom participă la încrucișare cu o probabilitate fixată p_c (probabilitate de încrucișare – dată de intrare)

Etapă de încrucișare:

- Notăm $P^1(t) = \{X_1, X_2, \dots, X_n\}$
- for $i = 1, n$
 - generează u variabilă uniformă pe $[0, 1]$
 - dacă $u < p_c$ atunci marchează (va participa la încrucișare)



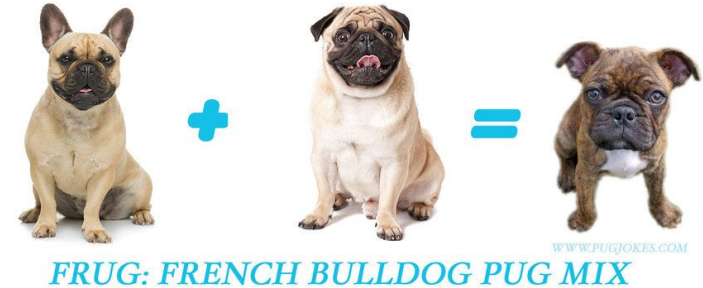
Încrucișarea

Cu un punct de tăietură (de rupere)

Un cromozom participă la încrucișare cu o probabilitate fixată pc (probabilitate de încrucișare – dată de intrare)

Etapă de încrucișare:

- Notăm $P^1(t) = \{X_1, X_2, \dots, X_n\}$
- for $i = 1, n$
 - generează u variabilă uniformă pe $[0, 1]$
 - dacă $u < pc$ atunci marchează (va participa la încrucișare)
- formează perechi disjuncte de cromozomi marcați și
- aplică pentru fiecare pereche operatorul de încrucișare;



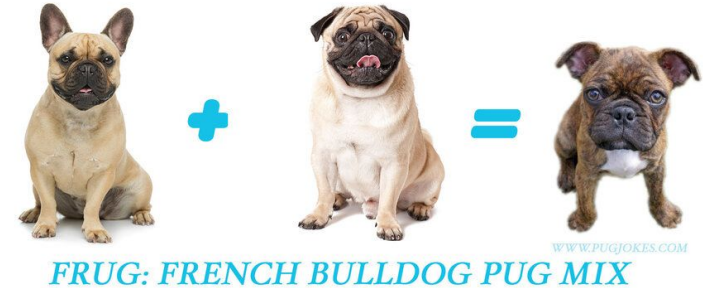
Încrucișarea

Cu un punct de tăietură (de rupere)

Un cromozom participă la încrucișare cu o probabilitate fixată pc (probabilitate de încrucișare – dată de intrare)

Etapă de încrucișare:

- Notăm $P^1(t) = \{X_1, X_2, \dots, X_n\}$
- for $i = 1, n$
 - generează u variabilă uniformă pe $[0, 1]$
 - dacă $u < pc$ atunci marchează (va participa la încrucișare)
- formează perechi disjuncte de cromozomi marcați și
- aplică pentru fiecare pereche operatorul de încrucișare;
- **descendenții rezultați înlocuiesc părinții în populație**



Mutația

schimbarea valorilor unor gene din cromozom

asigură diversitatea populației

probabilitatea de mutație pm – dată de intrare



Mutația

Etapa de mutație - **Varianta 1** (mutație rară):

- Notăm $P^2(t) = \{X_1, \dots, X_n\}$ populația obținută după încrucișare
- for $i = 1, n$
 - generează **u** variabila uniformă pe $[0, 1)$
 - dacă **u** < **p** atunci **generează o poziție aleatoare p** și
 - trece gena p din cromozomul X_i la complement $0 \leftrightarrow 1$



Mutația

Etapa de mutație - **Varianta 2**:

- Notăm $P^2(t) = \{X_1, \dots, X_n\}$ populația obținută după încrucișare
- for $i = 1, n$
 - for $j = 1, \text{len}(X_i)$
 - generează u variabilă uniformă pe $[0, 1)$
 - dacă $u < p_m$ atunci
 - trece gena j din cromozomul X_i la complement $0 \leftrightarrow 1$



Alte exemple





Next Time

