

Decision Models 2018/19

Bagnalasta Matteo - 833349

Biondi Stefano - 839358

Pierri Luca - 846597



Packing Santa's Sleigh

Risoluzione tramite Algoritmo Genetico

Packaging Problems

- Studiati a fondo negli ultimi decenni
- Focus su particolare variante 3D
 - Ordine di inserimento
 - Minimizzazione dello spazio occupato in altezza





1. Preprocessing

Dataset

- File CSV
- Le righe rappresentano i pacchetti
- 3 dimensioni (+1)
 - Dimension1: dimensione x del pacchetto
 - Dimension2: dimensione y del pacchetto
 - Dimension3: dimensione z del pacchetto
 - PresentId: ID del regalo



Preprocessing

- Utilizzo di Pandas per una veloce analisi descrittiva

	Dimension1	Dimension2	Dimension3
count	1000000.000000	1000000.000000	1000000.000000
mean	49.230199	49.260762	51.901871
std	61.875137	61.949176	61.981952
min	2.000000	2.000000	2.000000
25%	7.000000	7.000000	7.000000
50%	24.000000	24.000000	28.000000
75%	61.000000	61.000000	67.000000
max	250.000000	250.000000	250.000000

- Eterogeneità dei pacchetti



```
class Pack:
    def __init__(self, x, y, z, pack_id):
        self.x = x
        self.y = y
        self.z = z
        self.pack_id = pack_id
```

- Conversione delle righe in oggetti Pack
 - Facilità di manipolazione
 - Codice «parlante»
- Ordinamento dei pacchetti per PresentId decrescente





2. Analisi Competizione

Misura d'errore

$$M = 2 \max_i(z_i) + \theta(\Gamma) \quad \text{con } i = 1, \dots, N_p$$

- N_p = numero di pacchi da inserire
- z_i = coordinata z dell'i-esimo pacco
- $\theta(\Gamma) = \sum_{i=1}^{N_p} |i - \Gamma_i|$ con $\Gamma(i)$ ID del regalo nell'i-esima posizione
- Due addendi
 - Contributo di ordini di grandezza differenti



Analisi dei worst cases

- Altezza massima con pacchi impilati l'uno sull'altro

$$s_1 = 2 \cdot \sum_{i=1}^{N_p} \min_{j=1,2,3} \text{Dimension}(j) = 65.068.208 \simeq 6.5 \cdot 10^7$$

- Errore sull'ordinamento massimo con pacchi inseriti in ordine inverso

$$s_2 = 2 \cdot \sum_{i=1}^{500.000} (2i - 1) = 2 \cdot 500.000^2 = 5 \cdot 10^{11}$$

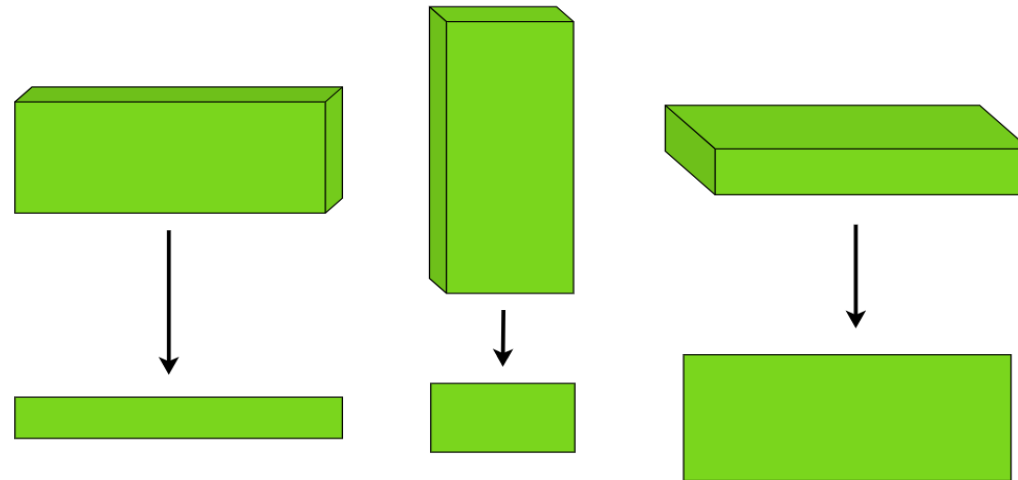




3. Approccio Metodologico

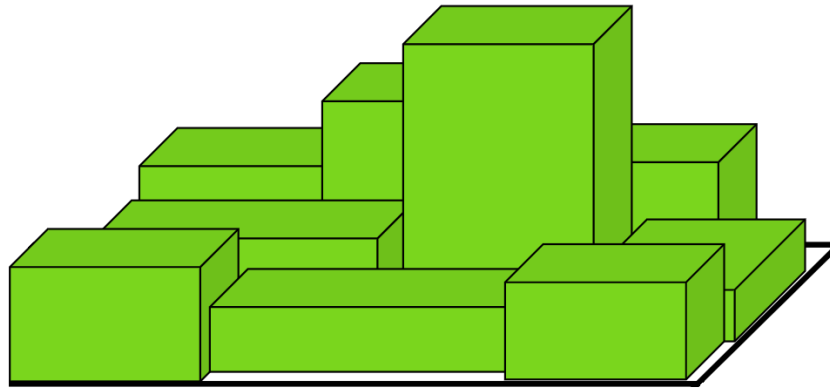
Strutturazione del problema

- Focalizzazione sulla minimizzazione dell'errore dato dall'ordinamento
- 3 superfici differenti per ogni pacco



- *Livello 1*

- Combinazione che minimizza la misura d'errore scelta tra 3^{n_1} combinazioni
- n_1 numero di pacchi da inserire al primo livello



- *Livello K*

- *Più complicato*
- *Inserimento basato sulla configurazione del piano precedente*



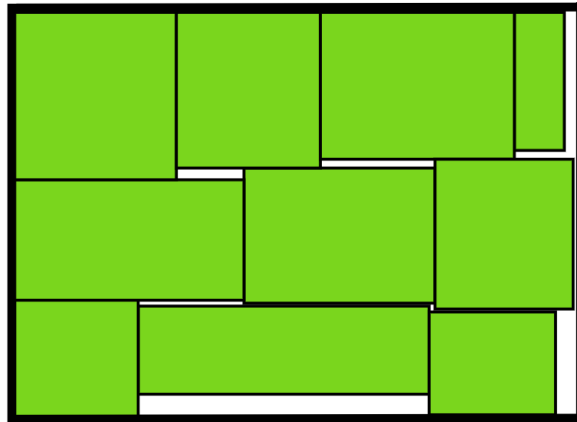
- Problema a layer

- Inserimento segue la relazione:

$$id_i > id_j \quad \forall i = 1, \dots, n_k \quad \forall j = 1, \dots, n_{k+1} \quad \forall k = 1 \dots N_{t-1}$$

- N_k = pacchi inseriti al livello k
- N_{k+1} = pacchi inseriti al livello k + 1
- N_t = numero totale dei layer

- Da 3D a 2D



Scelta ed Implementazione del Modello

- Algoritmo Genetico che sceglie quale delle 3^{n_i} combinazioni minimizzi

$$FitnessFunction_i = M_i/n_i$$

- Misura di errore finale

$$M = \sum_{i=1}^{N_t} M_i$$

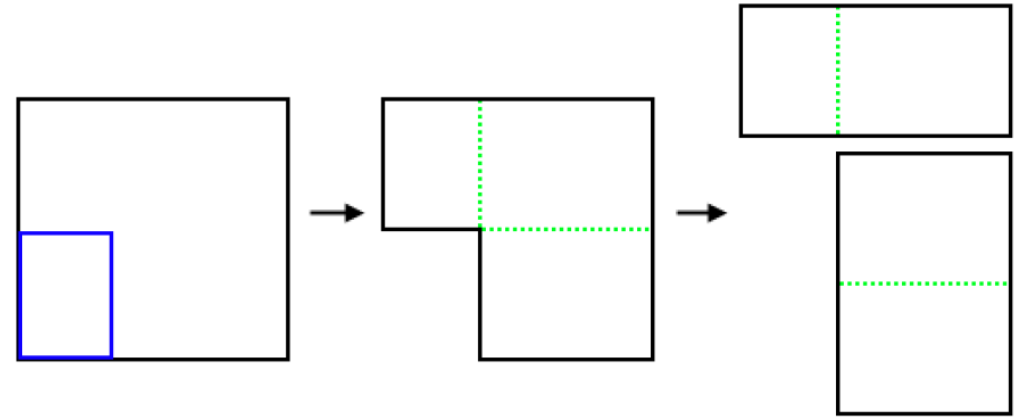
- N_t = numero totale di Layer utilizzati



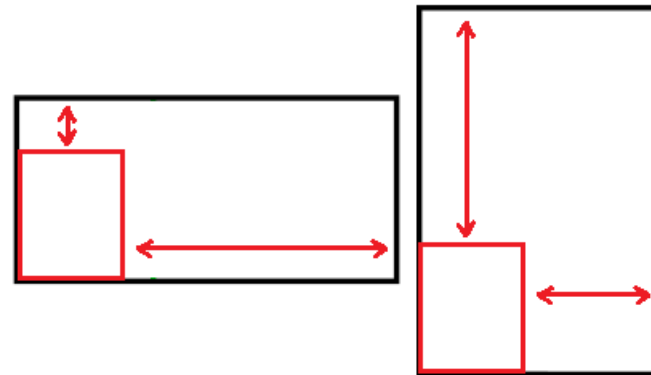
Rettangoli 2D nel singolo Layer

- *MaxRectBssf* (Maximum Rectangle – Best short side fit)

- *Primo rettangolo in basso a sinistra*
- *Calcolo dei rettangoli liberi*



- *Rettangolo successivo in basso a sinistra tale che sia minima la distanza libera*



- *RectPack modificata per garantire ordinamento*

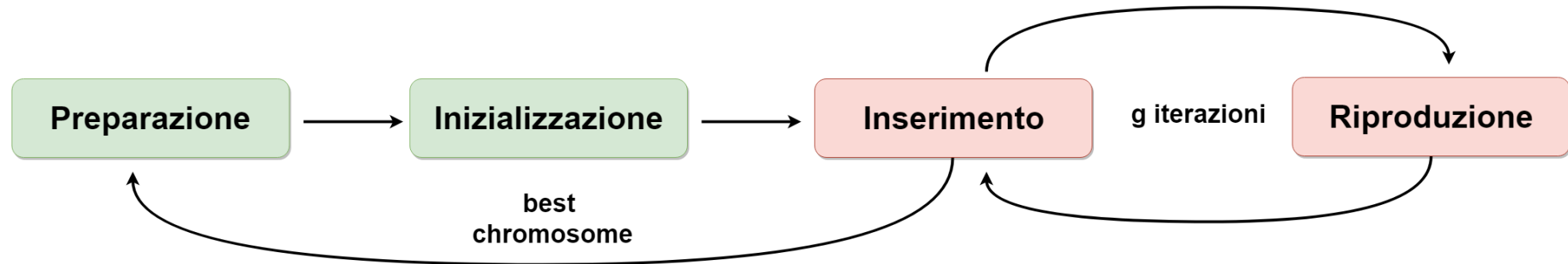


Algoritmo Genetico per Rotazioni

- Creazione dell'oggetto Chromosome

```
class Chromosome:  
    def __init__(self, packs, error=0):  
        self.packs = packs  
        self.error = error
```

- Per ogni layer

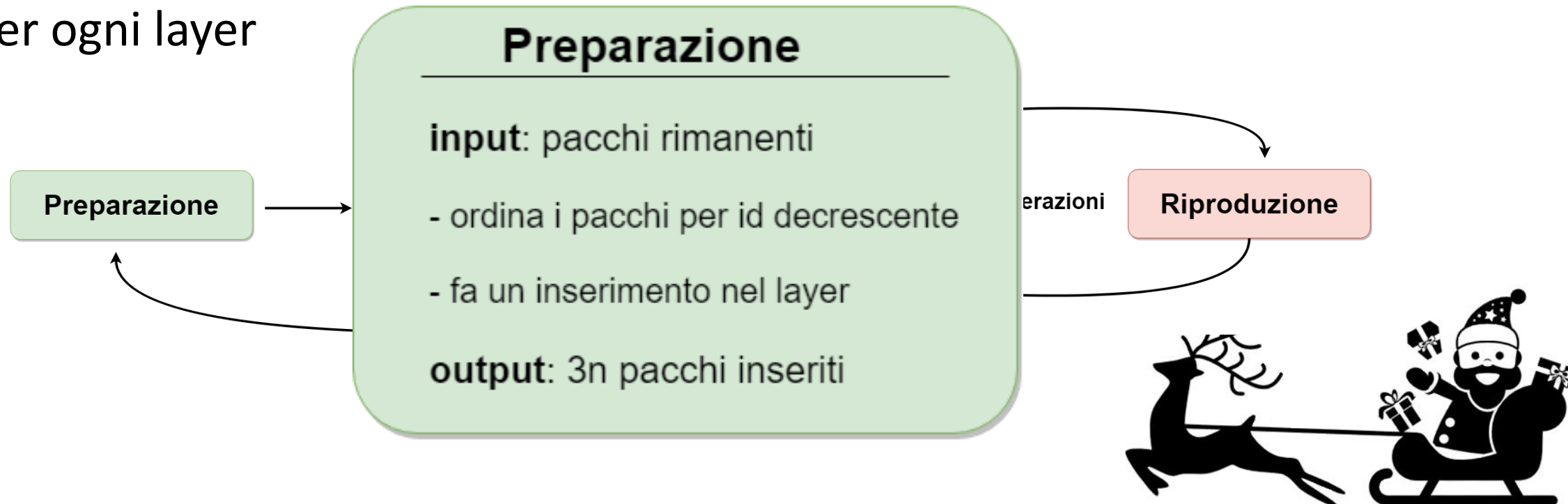


Algoritmo Genetico per Rotazioni

- Creazione dell'oggetto Chromosome

```
class Chromosome:  
    def __init__(self, packs, error=0):  
        self.packs = packs  
        self.error = error
```

- Per ogni layer

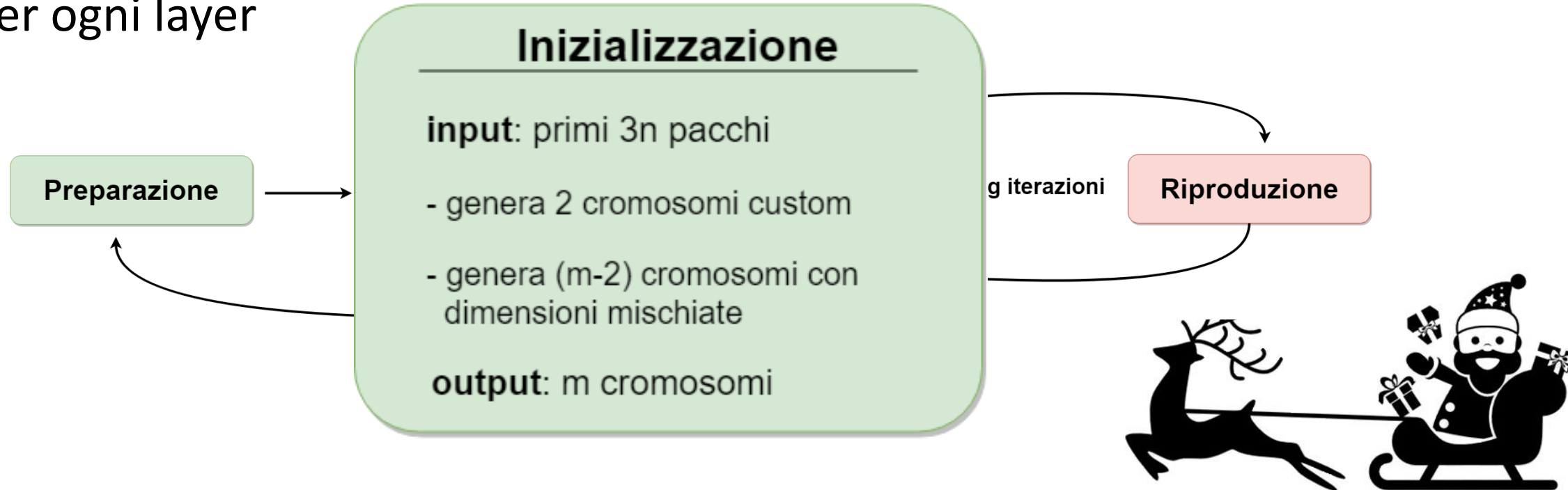


Algoritmo Genetico per Rotazioni

- Creazione dell'oggetto Chromosome

```
class Chromosome:  
    def __init__(self, packs, error=0):  
        self.packs = packs  
        self.error = error
```

- Per ogni layer

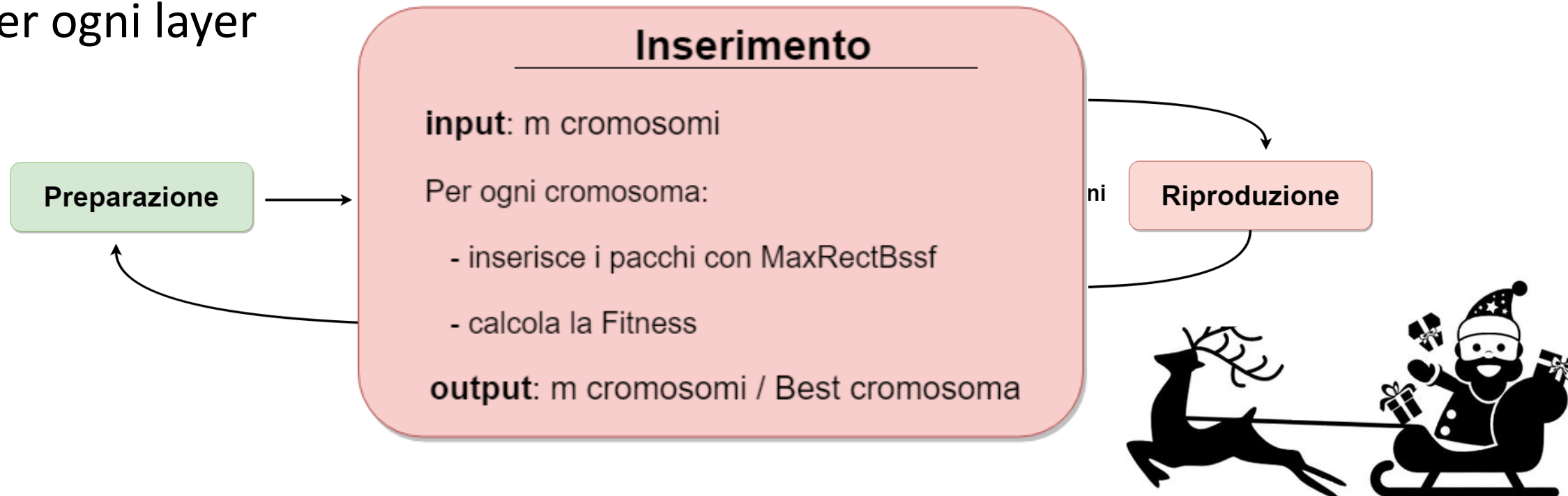


Algoritmo Genetico per Rotazioni

- Creazione dell'oggetto Chromosome

```
class Chromosome:  
    def __init__(self, packs, error=0):  
        self.packs = packs  
        self.error = error
```

- Per ogni layer

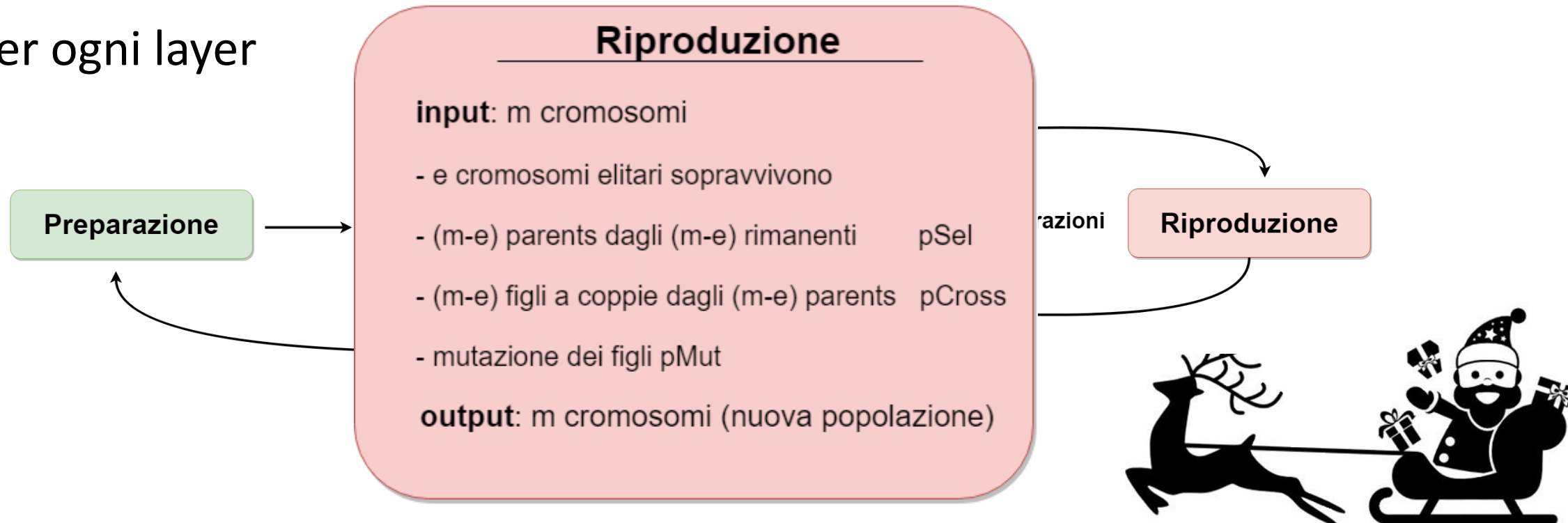


Algoritmo Genetico per Rotazioni

- Creazione dell'oggetto Chromosome

```
class Chromosome:  
    def __init__(self, packs, error=0):  
        self.packs = packs  
        self.error = error
```

- Per ogni layer



Algoritmo Genetico per Rotazioni

- Creazione dell'oggetto Chromosome

```
class  
def
```

- Per ogni layer

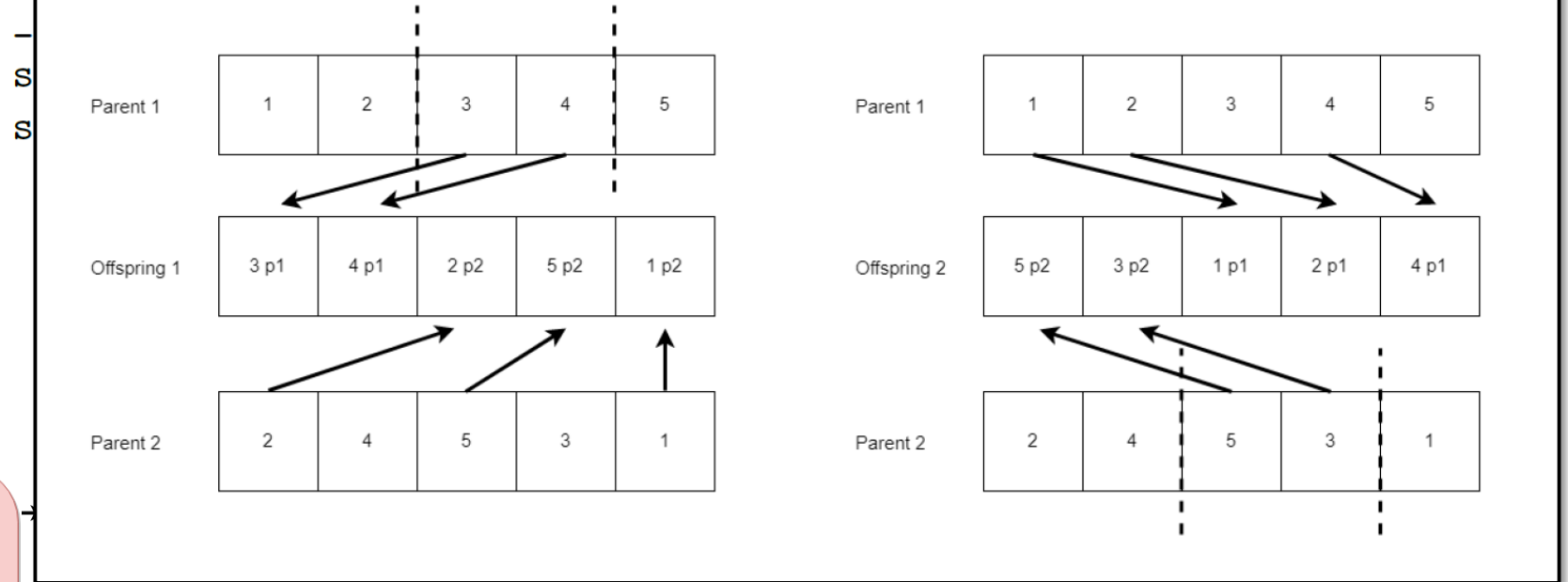
Riproduzione

input: m cromosomi

- e cromosomi elitari sopravvivono
- (m-e) parents dagli (m-e) rimanenti pSel
- (m-e) figli a coppie dagli (m-e) parents pCross
- mutazione dei figli pMut

output: m cromosomi (nuova popolazione)

Crossover:



best
chromosome





4. Discussioni e Conclusioni

Risultati

<i>Iterazioni</i>	<i>m</i>	<i>e</i>	<i>pSel</i>	<i>pCross</i>	<i>pMut</i>
3	20	5	0.9	0.6	0.3

- Riempimento senza algoritmo genetico:

Errore Medio	Numero pacchi Medio	Tempo Medio (s)
576×10^3	1329.4	33.8

- Riempimento con algoritmo genetico:

Errore Medio	Numero pacchi Medio	Tempo Medio (s)
244×10^3	867.7	2100.7



Discussione

- Errore medio normalizzato $\rho = \text{errore medio} / \text{numero medio di pacchi}$
 - Approccio genetico = 281.81
 - Approccio non genetico = 433.53
- Best Fitness costante per ogni iterazione
 - Miglioramento in fase di inizializzazione (2° cromosoma custom)
 - Algoritmo genetico non effettivamente migliorativo



Discussione

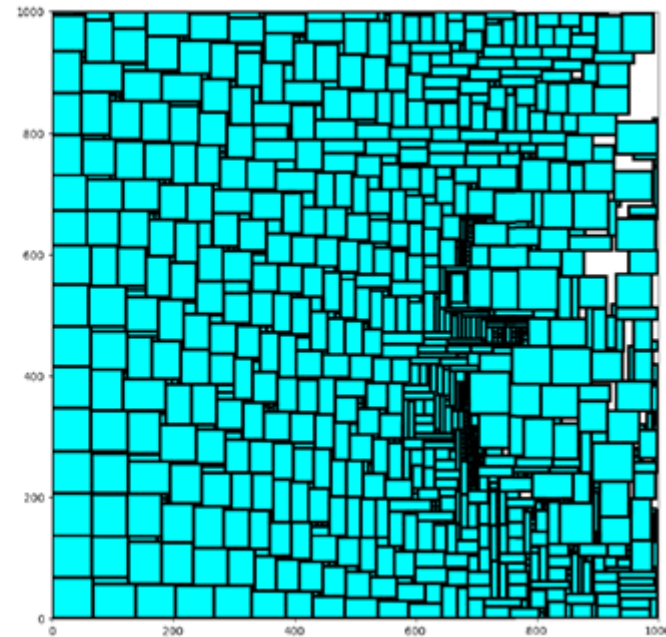
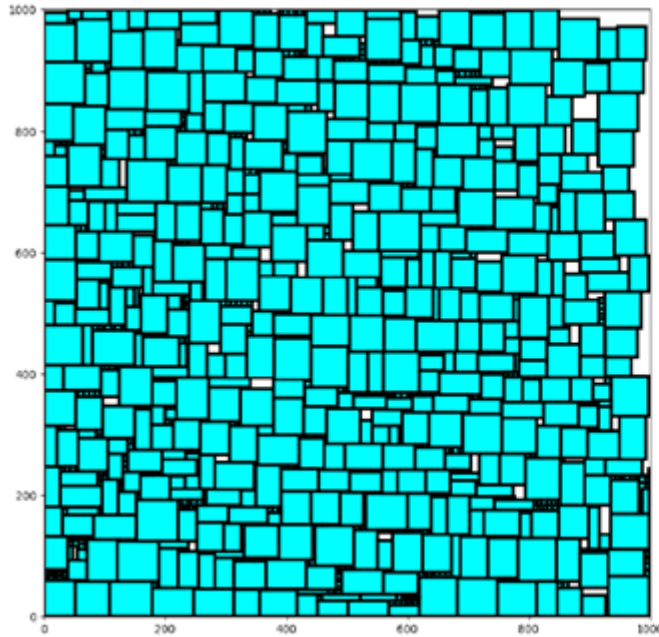
- Cambiamento parametri di controllo
 - Probabilità Crossover = 0.75
 - Probabilità di Mutazione = 0.4
 - Numero di Iterazioni = 5, Numero di Cromosomi = 20
 - Numero di Iterazioni = 3, Numero di Cromosomi = 30

Tuttavia...



Discussione

- Cambio di approccio: riempimento per **area**



Tuttavia...



Conclusioni

- Errore totale

$$\rho_{best} \cdot N_p = 281.81 \cdot 10^6 \simeq 2.8 \cdot 10^8$$

- Possibili Miglioramenti
 - ordinamento per area con tuning del parametro %
 - Tuning dei parametri di controllo
- $2 \times 10^6 < \text{Errore competizione} < 5 \times 10^7$
 - Riempimento per layer probabilmente troppo grezzo

