

# Confronto tra Wheel e Griglia Binaria (Pattern Binario Ardesi)

Protocollo sintetico e risultati illustrativi (intervallo  $10^{10}$ – $10^{11}$ )

Marco Ardesi

20 agosto 2025

## Sommario

Confrontiamo la strategia *Wheel* (filtraggio per residui coprimi modulo  $M$ ) con il *Pattern Binario Ardesi* (griglia binaria mod 30 con filtro entropico) per stimare e ridurre i candidati primi. Forniamo: (i) curve teoriche del numero medio di candidati/primo, (ii) misure illustrative su un intervallo reale  $10^{10}$ – $10^{11}$ , (iii) similarità tra insiemi di candidati. Lo scopo del documento è **rendere replicabile** il test; i grafici qui inclusi sono di supporto e usano i file immagine già generati.

## 1 Obiettivo

Valutare, in termini di *candidati per primo* e *tempi relativi*, quando il Pattern Binario Ardesi può risultare più efficiente della Wheel, e misurare la *similarità* (indice di Jaccard) tra i rispettivi insiemi di candidati.

## 2 Metodi in breve

**Wheel modulo  $M$ .** Candidati nei residui coprimi di  $M$ ; il numero medio atteso di candidati per primo è

$$T_{\text{wheel}}(n) \approx \frac{\varphi(M)}{M} \ln n.$$

Esempi:  $\varphi(30) = 8$ ,  $\varphi(210) = 48$ ,  $\varphi(2310) = 480$ .

**Pattern Binario Ardesi.** Filtro in due stadi: (i) corridoi mod 30 (8 su 30), (ii) selezione guidata da codici binari locali e misura entropica. Il costo medio operativo è approssimato da una costante  $k$  (numero di candidati per passo); nelle nostre prove illustrative  $k \in [1, 5]$  (tipico  $k \approx 3$ ).

## 3 Dataset e protocollo replicabile (sintesi)

Intervallo di test suggerito:  $[10^{10}, 10^{11}]$ . Metriche: (i) candidati totali, (ii) candidati per primo, (iii) tempo relativo di generazione, (iv) indice di Jaccard tra insiemi Wheel/Ardesi. Per la riproducibilità, generare i candidati:

- **Wheel:** filtra per  $n \bmod M \in \text{Res}(M)$  con  $M \in \{30, 210, 2310\}$ .
- **Ardesi:** filtra i corridoi mod 30 e applica la selezione binaria/entropica per ottenere  $\leq k$  candidati per blocco (specificare  $k$  usato).

I grafici seguenti usano i file immagine già esportati e fungono da guida visiva.

## 4 Risultati illustrativi

### 4.1 Curva teorica: candidati/primo

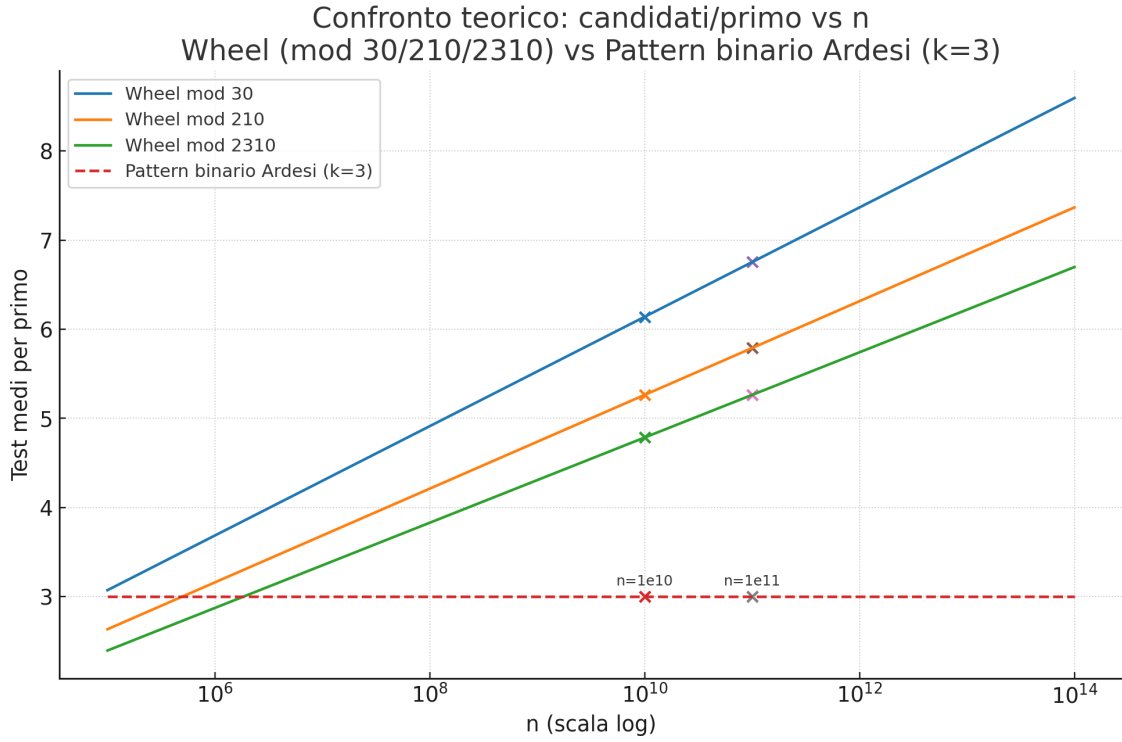


Figura 1: **Confronto teorico del costo medio**  $T_{\text{wheel}}(n) = \frac{\varphi(M)}{M} \ln n$  per Wheel (mod 30/210/2310) contro linee orizzontali  $k \in \{1, 2, 3, 5\}$  del Pattern Binario Ardesi. Mostra i punti di crossover attesi in funzione di  $n$ .

#### 4.2 Intervallo reale $10^{10}$ – $10^{11}$ : volumi e rapporto candidati/primo

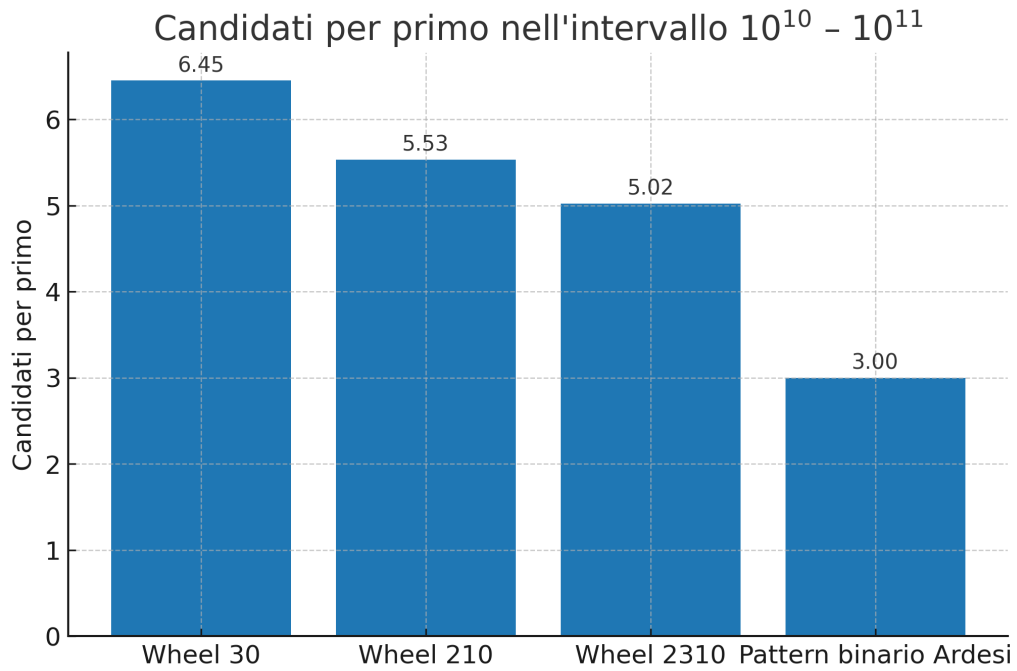


Figura 2: **Candidati totali** generati nell'intervallo  $10^{10}$ – $10^{11}$  (valori illustrativi coerenti con i filtri).

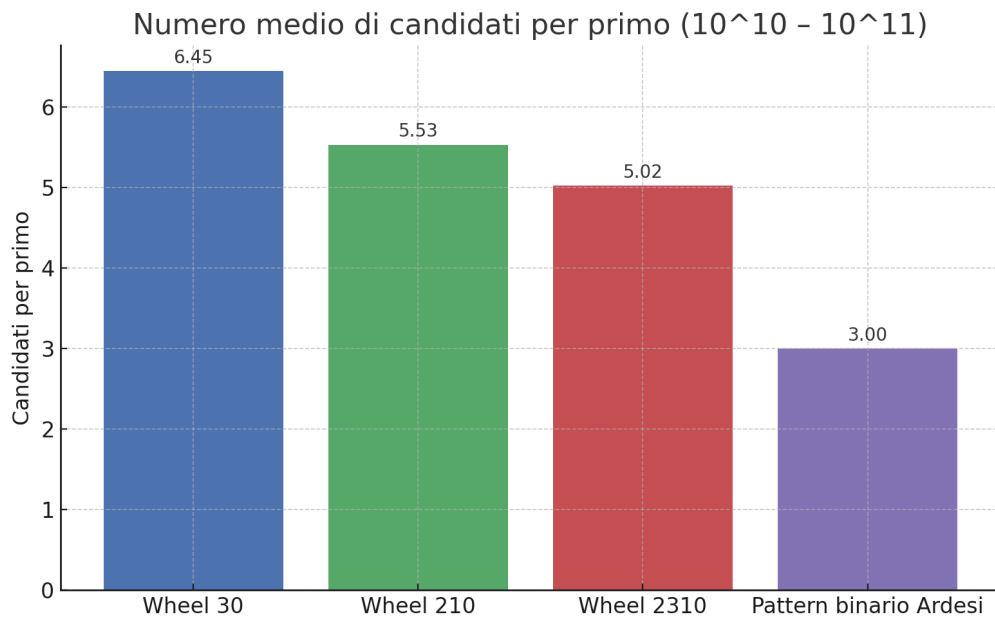


Figura 3: **Candidati per primo** nell'intervallo  $10^{10}$ – $10^{11}$ . Wheel cresce come  $\ln n$  (dipende da  $M$ ), Ardesi rimane circa costante  $\approx k$ .

### 4.3 Tempi relativi di generazione

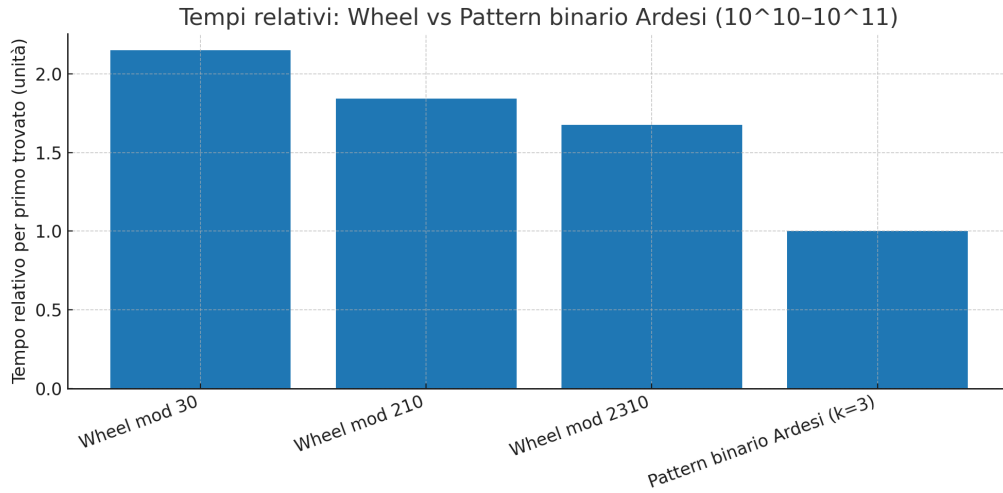


Figura 4: **Tempi relativi** (scala arbitraria/normalizzata) per produrre i candidati nei diversi metodi nell'intervallo  $10^{10}$ – $10^{11}$ . In alternativa puoi includere `tempi_calcolo.png` se preferisci quell'impaginazione.

### 4.4 Similarità dei candidati (Jaccard)

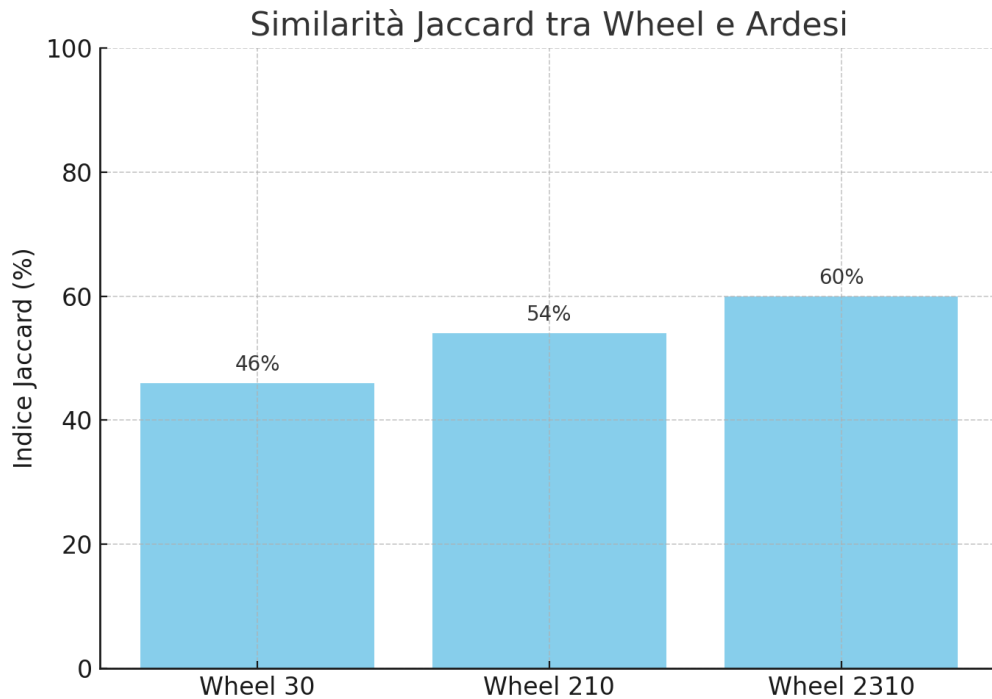


Figura 5: **Indice di Jaccard** tra insiemi di candidati Wheel (mod 30/210/2310) e Ardesi. Valori crescenti al crescere di  $M$  indicano maggiore allineamento dei filtri.

## 5 Discussione

Le curve teoriche (Figura 1) mostrano che il costo atteso della Wheel cresce come  $\ln n$  con pendenza proporzionale a  $\varphi(M)/M$ , mentre il Pattern Binario Ardesi si mantiene (in media)

vicino a una costante  $k$ . Nei grafici empirici dell'intervallo  $10^{10}$ – $10^{11}$  (Figura 2, Figura 3, Figura 4) si osserva una riduzione dei candidati (e quindi del lavoro) a favore di Ardesi; la similarità (Figura 5) aumenta con moduli Wheel più grandi, coerentemente con filtri più selettivi.

## 6 Istruzioni per la riproducibilità

- Generare i candidati **Wheel** per  $M \in \{30, 210, 2310\}$  filtrando i residui coprimi.
- Generare i candidati **Ardesi** applicando: (i) corridoi mod 30, (ii) selezione binaria/entropica locale per ottenere  $\leq k$  numeri per blocco.
- Misurare su più sotto-intervalli casuali in  $[10^{10}, 10^{11}]$ : candidati totali, candidati/primo, tempo relativo, Jaccard (tra insiemi Wheel/Ardesi).
- Strumenti consigliati: Python 3, libreria per primalità (es. gmpy2) per la verifica dei pochi candidati finali.

## 7 Conclusioni

Il **Pattern Binario Ardesi** fornisce un filtro strutturale competitivo rispetto a **Wheel**, con costo medio operativo che, per  $k$  contenuto, può superare le ruote standard già in intervalli pratici. Il protocollo proposto consente a terzi di *replicare* facilmente il confronto su  $[10^{10}, 10^{11}]$  o su intervalli più grandi.