

---

## Criteri calcolo soglia anomalia DLgs.50/2016

L'applicazione gestisce i 5 metodi di calcolo soglia anomalia previsti dal DLgs 50/2016, aggiornati secondo le modifiche introdotte dal DLgs.56/2017 "Correttivo codice dei contratti", entrato in vigore il 20 maggio 2017.

In fase di aggiudicazione viene sorteggiato il metodo da applicare. Se viene selezionato il metodo E (5), il coefficiente da utilizzare nel calcolo viene ulteriormente sorteggiato tra i valori possibili.

Mediante parametro di configurazione (parametro A1128 - *Calcolo aggiudicazione: (DLgs.50/2016) modalità sorteggio calcolo soglia*) il sorteggio può essere effettuato manualmente (default) o automaticamente dal software. Dettagli tecnici sul sorteggio in modalità automatica sono riportati alla fine del manuale.

Di seguito si specificano le modalità di calcolo della soglia di anomalia per ciascun metodo.

### 1. Metodo A:

- esclusione ali 20%: viene accantonato il 20% delle ditte ammesse in gara, arrotondato all'unità superiore, sia tra quelle di maggior ribasso che tra quelle di minore ribasso. Mediante parametro di configurazione (parametro A1134 - *Calcolo aggiudicazione: (DLgs.50/2016) criterio accorpamento offerte uguali nel taglio ali*) è possibile scegliere se accorpare o meno le offerte uguali ricadenti nelle ali. Il criterio impostato di default è in linea con le indicazioni della delibera ANAC n.206 del 01/03/2018. Il parametro può assumere i seguenti valori:

- 'maniera unitaria' (0 - default): vengono considerate come unica offerta le ditte con uguale ribasso, ricadenti sia all'interno delle ali che 'a cavallo'.
- 'maniera individuale' (1), vengono considerate come unica offerta solo le ditte con uguale ribasso che sono 'a cavallo' delle ali
- 'nessun accorpamento' (2)

Si sottolinea che se non previsto l'accorpamento delle offerte uguali ricadenti nelle ali, diversamente dagli altri criteri, una volta completato il calcolo della soglia di anomalia, non viene data evidenza delle ditte che sono rientrate nelle ali.

- calcolo media ribassi dopo esclusione ali
- calcolo media scarti dei ribassi superiori alla media dopo esclusione ali
- soglia anomalia = media ribassi + media scarti

### 2. Metodo B:

- esclusione ali 20%: come metodo A
- calcolo media ribassi dopo esclusione ali: come metodo A
- calcolo somma ribassi delle ditte ammesse in gara

Mediante parametro di configurazione (parametro A1132 - *Calcolo aggiudicazione: (DLgs.50/2016) criterio somma ribassi metodo B*), il cliente in base alla propria interpretazione può decidere se fare la somma dei ribassi considerando solo le ditte in gara dopo l'esclusione delle ali (default) oppure se considerare tutte le ditte in gara.

- si considera la prima cifra decimale della somma dei ribassi (X):
  - se X è numero pari o vale 0, soglia anomalia = media ribassi
  - se X è numero dispari, soglia anomalia = media ribassi - X% (la media viene decrementata)

### 3. Metodo C:

- calcolo media ribassi ditte ammesse in gara

- 
- soglia anomalia = media ribassi + 15%

4. **Metodo D:**

- calcolo media ribassi ditte ammesse in gara
- soglia anomalia = media ribassi + 10%

5. **Metodo E:**

- esclusione ali 10%: come metodo A, ma con percentuale 10 anziché 20
- calcolo media ribassi dopo esclusione ali: come metodo A
- calcolo media scarti dei ribassi superiori alla media dopo esclusione ali: come metodo A
- soglia anomalia = media ribassi + media scarti \* coefficiente

**Procedure pubblicate prima del 20 maggio 2017**

L'applicativo continua a gestire il metodi di calcolo soglia anomalia anche per le procedure pubblicate prima dell'entrata in vigore del decreto correttivo. Riportiamo di seguito i punti in cui la gestione differisce rispetto alla normativa corrente:

- **Metodo A:** la percentuale per il calcolo dell'ala inferiore e superiore è 10 anziché 20 %.
- **Metodo B:** come per il metodo A, la percentuale per il calcolo delle ali è 10 anziché 20 %
- **Metodo C:** la percentuale di incremento della media è 20 anziché 15 %
- **Metodo D:** il calcolo della media e della soglia viene fatta in termini assoluti, ovvero considerando l'importo del ribasso anziché direttamente i ribassi. Inoltre la media ottenuta non viene incrementata del 10 % ma decrementata del 20 %.
- **Metodo E:** i possibili coefficienti da applicare alla media degli scarti sono 0,6 – 0,8 – 1 – 1,2 – 1,4 anziché 0,6 – 0,7 – 0,8 - 0.9.

---

## Algoritmo di estrazione casuale del metodo di calcolo soglia e del coefficiente per il metodo E.

Il sorteggio automatico del metodo di calcolo soglia e del coefficiente nel caso del metodo E viene gestito dall'applicativo mediante l'utilizzo della funzione "Java Random", che restituisce un numero casuale compreso tra 1 e 5 per la selezione del metodo e tra 1 e 4 per la selezione del coefficiente. Tale funzione assicura che:

- o ogni numero ha la stessa probabilità di essere estratto ogni volta che viene lanciata la funzione sullo stesso intervallo di numeri;
- o non c'è mai dipendenza fra un numero estratto e il successivo

Si riporta di seguito un estratto del codice del software:

```
// Si istanzia la classe Java Random
Random r = new Random();
// Si estrae casualmente un indice da 1 a 5,
// l'indice estratto viene assegnato alla variabile "metsoglia".
int rand = r.nextInt(5) + 1;
metsoglia = new Long(rand);
if(rand==5){
    // Se selezionato il metodo E, si estrae casualmente un indice tra 0 e 4, se applicato il correttivo, e tra 0 e
    // se applicata la normativa prima del correttivo
    //Mediante l'indice estratto viene recuperato il coefficiente, considerando la corrispondente descrizione nel
    //tabellato All40 nel caso del correttivo e All27 altrimenti. Il coefficiente viene assegnato alla variabile
    "metcoeff".
    r = new Random();
    if ("true".equals(isModalitaDL2017)){
        rand = r.nextInt(4) + 1;
        codiceTabellato="All40";
    }else{
        rand = r.nextInt(5) + 1;
        codiceTabellato="All27";
    }
}
String valoreString = (String)sqlManager.getObject("select tabldesc from tabl where tablcod=? and tabltip=? ",
new Object[]{codiceTabellato,new Long(rand)});
if(valoreString!=null)
    metcoeff = UtilityNumeri.convertiDouble(valoreString, UtilityNumeri. FORMATO_DOUBLE_CON_ VIRGOLA_DECIMALE);
```

Specifica pubblica della funzione (classe) Java Random.

java.util

### Class Random

[java.lang.Object](#)

└ [java.util.Random](#)

#### All Implemented Interfaces:

[Serializable](#)

#### Direct Known Subclasses:

[SecureRandom](#)

---

```
public class Random
extends Object
implements Serializable
```

An instance of this class is used to generate a stream of pseudorandom numbers. The class uses a 48-bit seed, which is modified using a linear congruential formula. (See Donald Knuth, *The Art of Computer Programming, Volume 2*, Section 3.2.1.)

If two instances of `Random` are created with the same seed, and the same sequence of method calls is made for each, they will generate and return identical sequences of numbers. In order to guarantee this property, particular algorithms are specified for the class `Random`. Java implementations must use all the algorithms shown here for the class `Random`, for the sake of absolute portability of Java code. However, subclasses of class `Random` are permitted to use other algorithms, so long as they adhere to the general contracts for all the methods.

The algorithms implemented by class `Random` use a protected utility method that on each invocation can supply up to 32 pseudorandomly generated bits.