

# Modelovanie Monte Carlo

# Predaj špeciality

- Objednané množstvo (O) = ~~100~~ ks
- Nákupná cena (Cn) = ~~0,60~~ €
- Predané množstvo (P) = ~~80~~ ks
- Predajná cena (Cp) = 0,89 €
- Zostatok na konci dňa (Z) = 100 - 80 = ~~20~~ ks
- Výkupná cena (Cv) =  $C_n / 2 = \text{~~0,30~~ €}$
- Hospodársky výsledok = ?

Takýto fixný model nahradíme modelom, kde sú pre niektoré veličiny známe rozdelenia pravdepodobnosti.

# Predaj špeciality

- Objednané množstvo ( $O$ ) = 80 ks
- Nákupná cena ( $C_n$ ) ~  $UNIF(0,40; 0,70)$  €
- Dopyt ( $D$ ) ~  $TRIA(70, 90, 110)$  ks
- Predané množstvo ( $P$ ) =  $\min(O, D)$  ks
- Predajná cena ( $C_p$ ) = 0,89 €
- Zostatok na konci dňa ( $Z$ ) =  $P - D$  ks
- Výkupná cena ( $C_v$ ) =  $C_n / 2$  €
- Hospodársky výsledok = ?

# Hospodársky výsledok

- $HV = (P * C_p) + (Z * C_v) - (O * C_n)$
- $HV > 0 = \text{zisk}$
- $HV < 0 = \text{strata}$
- Uskutočníme veľké množstvo experimentov a môžeme stanoviť záver.

# Postup

Ak chceme získať 10000000 hodnôt HV:

- Nastav spolu = 0
- 10000000 -krát opakuj:
  - Vypočítaj  $HV_i$  z vygenerovaných hodnôt
  - $spolu = spolu + HV_i$
- Odhad priemerného  $HV = spolu / 10000000$
- Na základe získaných údajov môžeme stanoviť:
  - smerodajnú odchýlku
  - interval spoľahlivosti

# Hospodársky výsledok pri rôznej veľkosti objednávky

- Môžeme preveriť rôzne scenáre – 60, 80, 100 objednaných kusov špeciality.
- Výsledky je potrebné vždy porovnať cez OutputAnalyzer (t-test).

# Výberová smerodajná odchýlka

$$s = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (X_i - \bar{X})^2}$$

$n$  = počet získaných hodnôt

$X_i$  =  $i$ -ta získaná hodnota

$\bar{X}$  = priemer hodnôt  $X_i$

# Výberová smerodajná odchýlka

$$s = \sqrt{\left( \frac{1}{n-1} \cdot \sum_{i=1}^n X_i^2 \right) - \left( \frac{1}{n-1} \cdot \sum_{i=1}^n X_i \right)^2}$$

$n$  = počet získaných hodnôt

$X_i$  =  $i$ -ta získaná hodnota

**Tento tvar vzorca je pre programové spracovanie dát podstatne výhodnejší. Umožňuje vypočítať smerodajnú odchýlku bez potreby pamätať si výsledky jednotlivých experimentov!**



# Interval spoľahlivosti

$$\left\langle \bar{X} - \frac{s \cdot t_{\alpha}}{\sqrt{n}}, \bar{X} + \frac{s \cdot t_{\alpha}}{\sqrt{n}} \right\rangle$$

Hodnota  $t_{\alpha}$  je hodnota z tabuliek Studentovho rozdelenia s  $n-1$  stupňami voľnosti. Ak je  $n \geq 30$ , teda máme viac ako 30 experimentov môžeme použiť tabuľky normovaného normálneho rozdelenia, označovaného ako  $N(0, 1)$ .

Ak napríklad požadujeme 95% obojstranný interval spoľahlivosti bude hodnota  $t_{\alpha}$  rovná 1,96.

# Intervaly spoľahlivosti

- Ak majú intervaly spoľahlivosti pre rôzne objednané množstvá prienik, rozdiel medzi výsledkami experimentov nie je štatisticky významný.
- Nemôžeme s istotou tvrdiť, že jedno zo skúmaných množstiev prinesie väčší zisk!

# Generovanie „náhodných“ čísel

Pre vytvorenie generátora náhodných čísel s ľubovoľným rozdelením pravdepodobnosti nám postačí existencia generátora spojitého rovnomerného rozdelenia na intervale  $(0, 1)$ .

# Generovanie „náhodných“ čísel v jazyku Java

- Pri modelovaní náhodných javov využijeme triedu Random.
- Vybrané metódy:
  - `double nextDouble()` – vráti desatinné číslo spojitého rovnomerného rozdelenia na intervale  $<0, 1)$
  - `int nextInt(int n)` – vráti celé číslo rovnomerného diskretného rozdelenia na intervale  $<0, n)$

# Generovanie „náhodných“ čísiel

- Rovnomerné spojité rozdelenie (UNIF):

$$\text{UNIF}(a, b) = a + (b - a) \cdot \text{UNIF}(0, 1)$$

- Trojuholníkové (symetrické) rozdelenie (TRIA):  
a – minimum, b – modus, c- maximum

$$\text{TRIA}(a, b, c) =$$

$$a + \frac{(c - a)}{2} \cdot (\text{UNIF}(0, 1) + \text{UNIF}(0, 1))$$

# Pravidlá pre použitie generátora náhodných čísel

Triedu Random (a s ňou podobné triedy) sú v jednotlivých jazykoch založené na použití lineárneho kongruenčného generátora. Pre jeho správne fungovanie je potrebné zabezpečiť aspoň:

- pre každý náhodný jav vytvoríme práve jeden generátor náhodných čísel (tento sa môže skladať z viacerých pomocných generátorov)
- jednotlivé generátory inicializujeme násadami získanými cez jedinou inštanciu triedy Random (generátor násad)
- nevytvárame pre každý experiment nové inštanície generátorov

Ďakujem za pozornosť