

KIV/VSS

Simulace šíření lesího požáru buněčným automatem

Zdeněk Valeš

1.1. 2019

1 Zadání

Pomocí nástroje NetLogo naprogramujte simulaci lesního požáru. Základem pro semestrální práci bude existující příklad simulace v programu NetLogo, který rozšířím o další parametry (vítr, hořlavost terénu, typy terénu, jiný model šíření ohně, ...) tak, jak je popsáno v článku Simulation of forest fire fronts using cellular automata[1]. Práce bude umět simulovat šíření ohnivé stěny i šíření ohně z předem vybraného bodu. Ideálně by práce měla umět načíst mapu ze souboru a na té provést simulaci.

2 Teorie

TODO: Popis stávajícího modelu v NetLogo. Popis modelu v článku. Popis mého modelu.

2.1 Základy pravděpodobnosti

Název	Značení	Spojité	Diskrétní
Střední hodnota	$E(X)$	$\int xf(x)$	$\sum s_i p_i$
Rozptyl	$\sigma^2, D(X)$	$\int (x - E(X))^2 p(x)$	$\sum p_i (x_i - E(X))^2$
Směrodatná odchylka	σ, s_x	$\sqrt{D(X)}$	
Variační koeficient	v_x	$\frac{s_x}{\bar{x}}$	

Tabulka 1: Tabulka základních vzorečků

Název	Hustota ppsti (PDF)	Distribuční fce (CDF)	Střední hodnota	Rozptyl
Rovnoměrné	$\frac{1}{b-a}$	$\frac{x-a}{b-a}$	$\frac{a+b}{2}$	$\frac{1}{12}(b-a)^2$
Normální	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$	μ	σ^2
Exponenciální	$\lambda e^{-\lambda x}$	$1 - e^{-\lambda x}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Poissonovo	$\frac{\lambda^x}{x!} e^{-\lambda}$	exp. schody	λ	λ
Trojúhelníkové (a-c-b)	0 pro $x < a$ $\frac{2(x-a)}{(b-a)(c-a)}$ pro $x < c$ $\frac{2}{(b-a)}$ pro $x = c$ $\frac{2(b-x)}{(b-a)(b-c)}$ pro $x \leq b$ 0 pro $x > b$	0 pro $x \leq a$ $\frac{(x-a)^2}{(b-a)(c-a)}$ pro $x \leq c$ $1 - \frac{(b-x)^2}{(b-a)(b-c)}$ pro $c < b$ 1 pro $x \geq b$	$\frac{a+b+c}{3}$	$\frac{a^2+b^2+c^2-ab-ac-bc}{18}$

Tabulka 2: Tabulka se vzorečky pro základní rozdělení

2.2 Generování náhodných čísel

- Kvazirozměrné rozložení, generátor uniformního rozložení (celé číslo na n bitech)

- Metoda prostředních čtverců
- Lineární rovnice + modulo aritmetika ($y_{i+1} = (ay_i + c) \bmod m$)

2.2.1 Generování dalších rozdělení

Transformační metoda Uniformní rozdělení transformujeme podle inverzní distribuční fce $F^{-1}(u)$. Vhodné pokud je $F^{-1}(u)$ snadno zjistitelná.

Vylučovací metoda Musí být známa hustota ppsti $f(x)$. Dvěma uniformními generátory dostanu čísla v prostoru:

- G_1 s uniformním rozdělením na $[a, b]$
- G_2 s uniformním rozdělením na $[0, M]$
- $G_1 \Rightarrow y_1 \Rightarrow x_i = (b - a)y_1 + a$
- $G_2 \Rightarrow y_2 \Rightarrow z_i = My_2$
- Pokud $z_i < f(x_i)$ pak x_i je náhodné číslo s rozdělením $f(x)$; jinak opakuj

Obecné diskrétní rozdělení Pokud znám schodovou distribuční fci, generuji 1 číslo s uniformním rozdělením podle tabulky (CDF) určím výslednou hodnotu.

2.2.2 Generování normálního rozdělení

Součet n náhodných čísel s rovnoměrným rozdělením se asymptoticky blíží k normálnímu rozdělení. $s_n = \sum_{i=1}^n y_i$, hodí se volit $n = 12$ protože $E\{s_n\} = nE\{y_i\} = \frac{n}{2} = 6$ a $D\{s_n\} = nD\{y_i\} = \frac{n}{12} = 1$, je tedy snadné generovat gaussovo rozdělení se $\mu = 6$ a $\sigma = 1$. Pro zadané μ a σ : $\sigma \cdot (\sum_{i=1}^{12} y_i - 6) + \mu$

Box-Müllerova transformace Stačí dvě hodnoty x_1, x_2 s normovaným rovnoměrným rozdělením:

- $z_1 = \sqrt{-2\ln(-x_1)} \cos(2\pi x_2)$
- $z_2 = \sqrt{-2\ln(-x_1)} \sin(2\pi x_2)$

Lze ještě aplikovat parametry normálního rozdělení: $z_1 \sigma \mu$.

2.2.3 Testování generátoru

Ověřit, zda má generátor zadané vlastnosti (střední hodnotu, rozptyl, délka periody ...).

χ^2 test Testuji, že nějaká hypotéza neplatí (nebo, že ji nelze zamítnout).

- Hodnoty z $\{y_i\}_1^n$ rozdělím do k intervalů. V každém intervalu spočtu četnost θ_i , ppst p_i , že hodnota y_i spadne do intervalu.
- $\chi^2 = \sum_{i=1}^k \frac{(\theta_i - np_i)^2}{np_i}$
- Porovnání s tabulkovou hodnotou. $\chi^2 \leq \chi_{tab}^2$ pak hypotézu nelze zamítnout. Jinak hypotézu zamítnu na hladině ppsí α .

2.3 Markovské náhodné procesy

Poissonovo (počet jevů v určitém čas. intervalu) vs. Exponenciální rozdělení (délka intervalu mezi dvěma událostmi).

- Stř. doba setrvání ve stavu i : $T_i = \frac{1}{\lambda_i}$
- Stř. frekvence přechodů po hraně z i do j : $f_{i,j} = p_i \cdot \lambda_{i,j}$ (pouze bez abs stavů)
- Stř. doba cyklu průchodů stavem i : $T_{ci} = \frac{1}{f_i}$ (pouze bez abs stavů)

2.4 Systémy hromadné obsluhy

Vstupní proud: a, λ . Doby obsluh: s, μ . Fronta: w . Celý systém: q . Doba: T, t (velká písmena = stř. hodnoty, malá konkrétní). Počet požadavků: L . Zátěž systému: $\rho = \frac{T_{obsluha}}{T_{mezi\ prichody}} = \frac{T_s}{T_a}$. Pokud $\rho < 1$, pak se jedná o stacionární režim (systém není přetížený).

Charakteristiky vstupního proudu Veličina τ . Exponenciální proud lze popsat jedním parametrem λ , proto $E\{\tau\} = T_a = \frac{1}{\lambda}$ (viz vzorečky ppsti). Koeficient variace určuje, jak moc je proud náhodný, typicky v intervalu $< 0; 1 >$, kde 0 jsou pravidelné příchody. Výstupní proud má stejnou periodu a frekvenci jako vstupní (pro stac. režim). Jinak $m \cdot \mu$ a rozdělení se blíží $F_s(t)$.

Fronta požadavků Aktuální počet požadavků ve frontě: w . Střední počet požadavků ve frontě L_w . Doba čekání jednoho požadavku t_w . Stř. doba čekání požadavku ve frontě T_w .

Charakteristiky SHO $T_s = \frac{1}{\mu}$. Mezi středními hodnotami platí následující vztahy. Stř. počet prvků v systému: $L_q = L_w + L_s = L_w + m \frac{\lambda}{\mu}$. Stř. doba průchodu systémem: $T_q = T_w + T_s = T_w + \frac{1}{\mu}$.

Littleovy vzorce:

- $L_q = \lambda \cdot T_q$
- $L_w = \lambda \cdot T_w$
- $T_w = L_w \cdot T_a$

2.4.1 Kendallová klasifikace

Znám charakteristiku vstupního proudu $F_a(t)$ a kanálu obsluhy $F_s(t)$, chci určit vlastnosti systému. Systém je popsán pěticí $X/Y/m(/I/disc)$

- **X**: prvd. rozdělení vstupního proudu
GI = obecné náhodné rozdělení, stat. nezáv.; G = obecné náhodné rozdělení;
M = exponenciální rozdělení; D = determ. intervaly
- **Y**: prvd. rozdělení dob obsluh
- **m**: počet kanálů obsluhy
- **I**: max. délka fronty (obvykle ∞).
- **disc.**: frontová disciplína (obvykle FIFO).

M/M/1 Nejjednodušší případ, charakterizováno parametry λ, μ . $\rho = \frac{\lambda}{\mu}$, pro stac. režim $\lambda < \mu$. Pokud stacionární, lze modelovat jako mark. proces: $p_k = \rho^k p_0 = \rho^k (1 - \rho)$.
 $E\{k\} = L_q = \sum_{k=0}^{\infty} k p_k = (1 - \rho) \frac{\rho}{(1 - \rho)^2}$.

M/M/m m obslužných kanálů pro 1 frontu. Koeficient vytížení: $\rho = \frac{1}{m} \frac{\lambda}{\mu} = \frac{1}{m} \frac{T_s}{T_a} = \frac{\lambda T_s}{m}$.
Pro $m \in \{1, 2\}$ přesný, jinde pouze přibližný odhad: $L_q = \frac{m\rho}{1 - \rho^m}$; $T_q = \frac{T_s}{1 - \rho^m}$.

Pokud konečná fronta, některé požadavky zahozeny, proto nelze použít vzorce pro nekonečnou délku fronty a proto platí, že $\lambda_{realne} < \lambda_{teoreticke}$. Vzorce pro konkrétní případ lze odvodit z mark. modelu.

M/G/1 Nemarkovský model. Potřebuji λ pro vstupní proud a $F_s(t)$, nebo $f_s(t)$ pro popis doby obsluhy. $\rho = \lambda T_s$ kde T_s je stř. doba rozdělení $F_s(t)$. Na zbytek potřebuji koef. variace (C_s^2). $C_s = \frac{\sigma(\tau)}{T_s}$.

- $L_w = L_{w(M/M/1)} \frac{1 + C_s^2}{2} = \frac{\rho^2}{1 - \rho} \frac{1 + C_s^2}{2}$
- $L_q = L_w + L_s (= L_w + \frac{\lambda}{\mu} = L_w + \rho)$
- $T_q = \frac{L_q}{\lambda}$
- $T_w = \frac{L_w}{\lambda}; T_w = T_q - T_s$

Markovské modely se používají jako odhad nejhoršího průběhu.

GI/G/1 Stat. nezávislost dob příchodů ve vstupu. Potřebujeme znát $F_a(t)$, nebo $f_a(t)$. Zatížení $\rho = \frac{T_s}{T_a}$. Přibližně $L_w = \frac{\rho^2}{1 - \rho} \frac{C_a^2 + C_s^2}{2}$. Pro D/D/1 je $L_w = 0$ pokud je $T_a > T_s$. Pro M/M/1 je $\frac{C_a^2 + C_s^2}{2} = 0$.

2.4.2 Složené sítě

Jsou-li všechny vstupní toky poissonovské a obsluhy exponenciální, jsou i výstupní toky exponenciální. → dílčí SHO lze řešit jako M/M/m. Littleův zákon: $L_q = \Lambda_0 \cdot T_q$ (v podstatě: co do sítě vstoupí z ní musí i vystoupit). Λ_0 je souhrnný vstupní tok do systému, Λ_i vnitřní frekvence toku v uzlu. Pro uzly ve stacionárním systému platí obdoba Kriehoffových zákonů (z nich lze pak zjistit Λ_i). Zatížení uzlu: $\rho_i = \frac{1}{m_i} \Lambda_i T_{si}$. Pokud každé $\rho_i < 1$, pak je systém stacionární.

Uzavřené sítě front lze řešit převedením na markovský model. V případě modelů interaktivního systému (n terminálů, 1 fronta, 1 server) se přechody do prava postupně zmenšují ($n\lambda, (n-1)\lambda, \dots, \lambda$). Propustnost pak je $x = \frac{1}{T_s}(1 - p_0)$ (ve stavu p_0 není co obsluhovat). $T_q = (n \frac{T_s}{1-p_0}) - T_t$ ($T_t = \frac{1}{\lambda}$ je doba 'přemýšlení' terminálu).

Nepoissonovské sítě front Toky obecně charakterizovány λ a variačním koef. C_τ . Pro stacionární platí $\lambda_{in} = \lambda_{out}$. Platí $C_{out}^2 = 1 + \rho^2(C_s^2 - 1) + (1 - \rho^2)(C_\tau^2 - 1)$. ρ určuje vliv rozdělení dob vstupů a dob obsluh. Tok s λ, C_τ lze dělit na n toků: $\lambda_i = \lambda p_i$, $C_i^2 = 1 + p_i(C_\tau^2 - 1)$. Slučovat toky lze podle: $\lambda = \sum_{i=1}^n \lambda_i$, $C^2 = 1 + \sum_{i=1}^n (\frac{\lambda_i}{\lambda})^2 (C_i^2 - 1)$. Pro jednotlivé elementární SHO lze použít vztahy pro GI/G/n.

2.5 Základy teorie spolehlivosti

Systematické (bugy) vs. náhodné poruchy. Opravitelné vs. neopravitelné systémy. Zálohy: cold (vypnutá, aktivuje se po výpadku), teplá (zapnutá, neaktivní až do výpadku), horká (paralelně s hlavním výpočtem). MTTF = střední doba do poruchy (v modelech často $\lambda = \frac{1}{MTTF}$). MTBF = střední doba mezi poruchami. MTTR = střední doba do opravy (v modelech často $\lambda = \frac{1}{MTTR}$).

Ukazatele spolehlivosti $Q(t)$ je distribuční fce ppsti poruchy. $R(t) = 1 - Q(t)$ je distribuční fce ppsti bezporuch. provozu. Hustota ppsti: $f(t)$. Intenzita poruch: $\lambda(t) = \frac{f(t)}{R(t)}$. Výpočty vychází z empiricky zjištěné vanové křivky (čeká se na ustálení systému kdy je λ cca konstantní), pak platí:

- $R(t) = e^{-\lambda t}$
- $Q(t) = 1 - e^{-\lambda t}$ (distrib. fce exp. rozdělení)
- $f(t) = \lambda e^{-\lambda t}$ (hustota ppsti exp. rozdělení)

Neobnovované objekty: $T_s = \frac{1}{\lambda}$. Obnovované objekty: $T_s = \frac{t_p}{n} = \frac{1}{n} \sum_{i=1}^n \tau_{pi}$ kde τ_{pi} jsou dby jednotlivých poruch. Obdobně střední doba cyklu (MTBF) je $T_c = \frac{1}{n} \sum_{i=1}^n (\tau_{pi} + \tau_{oi})$. Ppst, že objekt bude fungovat v libovolné době (koeficient pohotovosti): $K_p = \frac{t_p}{t_p + t_o}$. MTTR lze zavést jako $T_o = \frac{t_o}{n}$ (pro exp. $\frac{1}{\mu}$). $K_p = \frac{T_s}{T_s + T_q} = \frac{\mu}{\lambda + \mu}$. Součinitel prostoje: $K_n(t) = 1 - K_p(t)$.

Systémy s nezávislými prvky Paralelní vs. sériové zapojení prvků. Pro sériové platí: $R(t) = \prod_{i=1}^n R_i(t)$, $T_s = \sum_{i=1}^n \frac{1}{\lambda_i}$. Pro paralelní zapojení platí: $Q(t) = \prod_{i=1}^n Q_i(t)$. V případě kombinovaných zapojení postupně redukuje. Pro exp. doby poruch platí: $T_s = \int_0^{\infty} R(t)dt = \frac{1}{\lambda_1 + \dots} + \frac{1}{\lambda_1 + \dots}$ (integruji $e^{-(\lambda_1 + \dots)t}$ podle t). Mohu použít stavový graf, je n prvků, musí fungovat alespoň k . Pro výsledné R sečtu všechny stavy ($R_1 R_2 R_3$ pro stav 111, $R_1 Q_2 R_3$ pro stav 101, ...).

Markovský model pro neob. systém Podle problému vytvořím graf. Najdu všechny cesty z 1. do n (absorpčního stavu), pro každou cestu určím T_{ci} a p_{ci} . $T_s = \sum T_{ci} p_{ci}$. T_{ci} určím jako součet $\sum \frac{1}{\lambda_{odchozi}}$ pro každou hranu v cestě. p_{ci} určím jako součin $\frac{\lambda_{hrana}}{\sum \lambda_{odchozi}}$ pro každou hranu v cestě.

Markovský model pro ob. systém Klasika. $K_p = \sum_{ok\ stavy} p_i$, $T_s = \frac{\sum_{ok\ stavy}}{ppst\ prechodu\ do\ neok\ stavu}$.

2.6 Diskrétní stochastické modely

Stochastická = založena na náhodných číslech. Diskrétní = stav jen v diskrétních okamžicích. Událostní = čas založený na událostech (ne na pravidelném korku). Stoch. simulace = výsledky se blíží realitě s větším počtem pokusů (Buffonova jehla).

Metoda pseudoparalelních procesů Založeno na objektové dekompozici simulačního modelu. Aktivní objekty = pracují podle vlastního programu; Pasivní objekty = poskytují služby ostatním. Základní objekty pro simulaci: prvek seznamu (**LINK**) = objekt, který lze řadit do seznamů. Hlava seznamu (**HEAD**) = objekt reprezentující seznam. Proces (**PROCESS**) = aktivní prvek, může vykonávat činnost (generátory, kanály obsluhy).

LINK:

- into(seznam) vloží objekt do zadaného seznamu
- follow(prvek) zařadí objekt za daný prvek do seznamu
- precede(prvek) zařadí objekt před daný prvek
- out() vyjme prvek ze seznamu

HEAD:

- empty() je prázdný
- cardinal() jak je velký
- first() první prvek
- last()
- clear()

Prvky propojují tak, že nový liunk přidám do fronty následujícího uzlu (přímá reference vs. všechny prvky v hash + identifikátor). Jednotný interface pro uzly akceptující požadavky (accept(link)), zpracování závisí na daném objektu. Kanál obsluhy zpožďuje cestu požadavku v síti (metoda hold(time)).

Měření charakteristik sítě Doba průchodu systémem T_q : pro každý prvek času vstupu do systému a čas výstupu ze systému \rightarrow mohu počítat $E(x)$, $D(x)$ a histogram.

Statistika toku v místě: počítám rozdíl času v accept(). Mohu zjišťovat délky front L_q, L_w (vzorkování). Zatížení serveru: $\rho = \frac{\text{doba obsluh}}{\text{celková doba}}$.

Mechanismus bariéry Proces: střídání hold() a barrier.arrived(this). Bariéra: Postupně čeká až přijdou všechny procesy, pak všechny aktivuje.

Bariéra + KS (rendez-vous) Proces: střídání hold() a barrier.arrived(this). Bariéra: Postupně čeká až přijdou všechny procesy, pak proběhne KS (hold()), pak se uvolní všechna vlákna.

Monitor KS se spouští pro každé vlákno ale smí v ní být jen jedno. Proces: hold(λ); section.enter(); hold(μ); section.exit(). Monitor: nové procesy uspí, pokud je KS obsazená, při odchodu procesu probere první čekající.

Producent, konzument Producent: Dává (produkce = hold()) linky do bufferu, dokud může, pak se uspí. Pokud je něco v bufferu, vzbudí konzumenta. Konzument: pokud je buffer prázdný, uspí se. Jinak konzumuje. Pokud je buffer prázdný a producent spí, vzbudí jej. Konzumace = hold().

2.7 Benchmarky

Benchmark musí být opakovatelný a měřitelný. Dělení od nejsložitějších ale nejpřesnějších k nejsnáze naimplementovatelným: program, kvli kterému testujeme \rightarrow reálné programy \rightarrow jádra \rightarrow syntetické benchmarky \rightarrow specifické algoritmy.

Hodnocené vlastnosti Rychlost (odezva, MIPS), Spolehlivost, Dostupnost, Cena. Typické metriky: čas (něčeho), kapacita (celého/části systému), výkon (efektivita), spolehlivost (MTBF), dostupnost (MTTF, MTTR). Reálná vs. syntetická zátěž. Parametry = předem dané, faktor = parametry zátěže měnící se v čase.

Návrh testu S minimálním úsilím získat co nejvíc informací. Přesný popis zátěže a posloupnosti kroků. Přesná volba cílů (z toho vychází volba testů a interpretace výsledků). Systém = souhrn použitého SW, HW. Uživatel = netita využívající systém. Metrika = kritérium zvolené pro hodnocení, Zátěž = požadavky zasílané uživateli. Měření na reálném systému vs. analytický model vs. simulace.

Příklady benchmarků Erastothenovno síto: hledání prvočísel, výkon závisí na rychlosti paměti, implementaci polí, velikosti cache. Ackermanova funkce: test schopnosti překladače optimalizovat rekurzi. Whetstone (1972 - staré): sada instrukcí podle statistiky (50% jump, 30% aritmetika, ...) pro vědecké programy, měří počet float instrukcí za sekundu. Závisí na kvalitě překladače. Dhrystone (1984): realističtější než wheatstone (=použita jiná statistika pro instrukce), měří cykly/s nebo MIPS. CoreMark (2009): zpracování seznamů, matice, konečný automat, CRC, vlastní skóre. Linpack: řešení float soustavy lin. rovnic gauss. metodou, TFLOPS, porovnání superpočítačů. Debit/Credit: test databází, v každé transakci zápis do 4 tabulek, cena/1 transakci (TPC-C V5 modernější). SPEC Suite: sada různých reálných programů, testují různé komponenty/vlastnosti PC (pro Javu SPEC JVM2008). PCMark(domácí PC), 3DMark (grafické karty)

Základní chyby Korelace není kauzalita. Aritmetický průměr vs. harmonický průměr (lepší pro převrácené hodnoty). Špatná tvorba grafu (měřítko, počátek os, popis os, špatný typ grafu). Porovnávání výsledků různých benchamrků. Povaha benchmarku neodpovídá povaze reálného použití stroje/SW. Nesystematický přístup (např. při volbě parametrů). Nevhodně zvolená zátěž systému. Benchmark je známý a výrobci pro něj optimalizují.

3 Použití

Možnost nahrávat mapy z obrázků. Možnost nastavení hořlavosti druhů terénu, možnost nastavení matice výšky terénu.

4 Závěr

Reference

- [1] ENCIAS, Hernández, Hoya WHITE, Martín del RAY a Rodríguez SANCHÉZ. Simulation of forest fire fronts using cellular automata. *Advances in Engineering Software: Advances in Numerical Methods for Environmental Engineering*. 2007, 2007(6), 372-378. ISSN 0965-9978. Dostupné také z: <https://www.sciencedirect.com/science/article/pii/S0965997806001293>