

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.

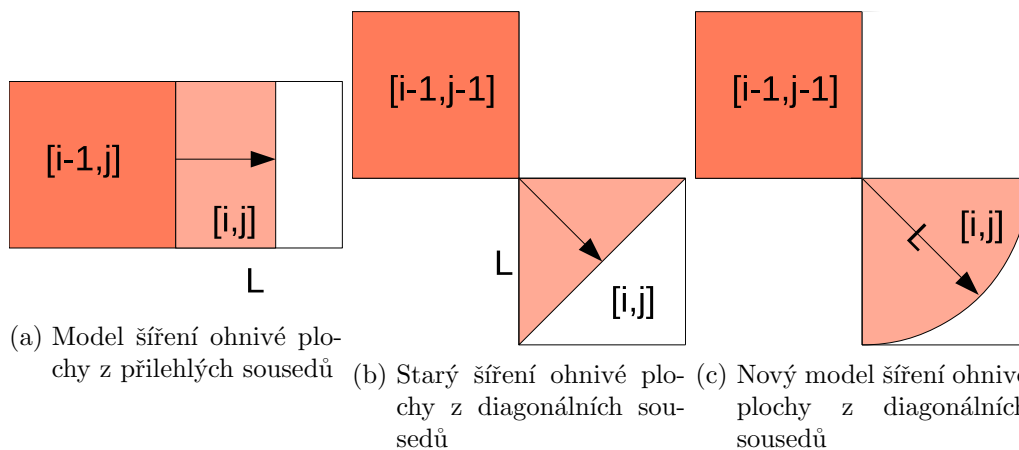
2 Teorie

Práce implementuje model šíření lesního požáru celulárním automatem tak, jak byl navržen v [1]. Původní simulace šíření požáru, která je dostupná v knihovnách NetLogu, byla založená na šíření ohně 'želvičkami'. Model navržený v článku předpokládá pouze buněčný automat a je vhodný po použití simulace homogenního i nehomogenního lesa (což je reálnější případ).

Les je interpretován jako dvourozměrné pole čtvercových buněk o délce L . Každá buňka $[i, j]$ má v čase t stav $a_{i,j} = \frac{\text{spalena plocha } [i,j]}{\text{celková plocha } [i,j]}$ vyjadřující část spálené plochy buňky. Pokud je $a_{i,j} = 0$, je buňka netknutá, pokud je $a_{i,j} = 1$, je buňka úplně vyhořelá.

2.1 Model šíření ohně

Nejzákladnější model šíření ohně je založen na sčítání spálené plochy v osmi-okolí jedné buňky. Přírůstky od jednotlivých sousedů jsou násobeny koeficientem μ , který v sobě zahrnuje parametry jako hořlavost, vítr, nebo výška terénu. V případě přilehlých sousedů je doba potřebná k úplnému spálení $t = \frac{L}{R}$ (naznačeno na obrázku 1a), kde L je rozměr plochy lesa (jedné 'buňky') a R je hořlavost terénu.

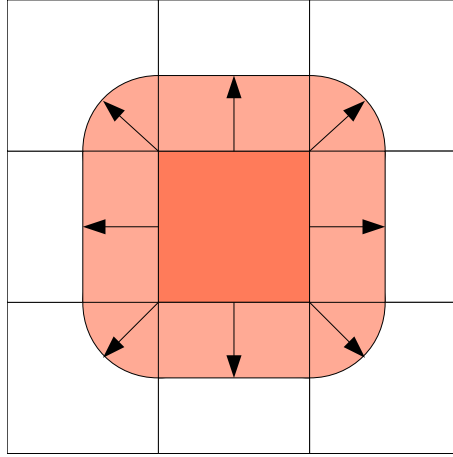


Obrázek 1: Porovnání staršího a nového modelu šíření ohně z diagonálních sousedů

V článku je popsána první modifikace základního modelu (podle [2]), která rozlišuje

přírůstek spálené plochy od přilehlých a diagonálních sousedů. Tato modifikace plyne z předpokladu šíření ohně po úhlopříčce v případě diagonálních sousedů, viz obrázek 1b.

Tvůrci článku tento model dále posouvají a počítají spálenou oblast diagonálních sousedů jako část kruhu (viz obrázek 1c). Podle tvůrců článku tento přístup více odpovídá realitě. U nového modelu tedy oheň z diagonálního souseda spálí za dobu t plochu buňky rovnou $\frac{\pi L^2}{4L^2} \approx 0.785 \approx 78.5\%$. Pro úplnost je na obrázku 2 znázorněn úplný model šíření ohně z jedné buňky do svých sousedů.



Obrázek 2: Úplný model šíření ohně z buňky

Parametry ovlivňující šíření ohně Prvním parametrem je rychlost šíření $R[\frac{m}{s}]$, který popisuje hořlavost buňky. V případě nehomogenních lesů hraje v rovnici roli podíl $\frac{R_{i,j}}{R}$, kde R je nejvyšší hořlavost v rámci simulovaného světa.

Vliv terénu je pro každou buňku popsán maticí $\Phi_{i,j}$ uvedenou v 1. Hodnoty matice by měly být dány rozdílem výšek sousedních buněk od centrální buňky: $h_{i+\alpha, j+\beta} = F(H_{i+\alpha, j+\beta} - H_{i,j})$. Přesný tvar funkce $F()$ se mi v článcích [1] a [2] nepodařilo najít. Z publikovaných výsledků testů však plyne, že vyšší hodnoty v matici odpovídají nižším sousedům. Pokud je tedy buňka $[i + \alpha, j + \beta]$ vyšší než buňka $[i, j]$, bude platit, že $h_{i+\alpha, j+\beta} < h_{i,j}$.

$$\Phi_{i,j} = \begin{pmatrix} h_{i-1,j-1} & h_{i-1,j} & h_{i+1,j+1} \\ h_{i,j-1} & 1 & h_{i,j+1} \\ h_{i+1,j-1} & h_{i+1,j} & h_{i+1,j+1} \end{pmatrix} \quad (1)$$

Obdobně je řešen i vliv větru, který je popsán maticí $W_{i,j}$ uvedenou v 2. Matice funguje stejně, jako v případě matice s terénem. Pokud vítr fouká od severu k jihu, měly by hodnoty $w_{i-1,j-1}, w_{i,j-1}, w_{i+1,j-1}$ být větší než zbytek matice $W_{i,j}$.

$$W_{i,j} = \begin{pmatrix} w_{i-1,j-1} & w_{i-1,j} & w_{i+1,j+1} \\ w_{i,j-1} & 1 & h_{i,j+1} \\ w_{i+1,j-1} & w_{i+1,j} & w_{i+1,j+1} \end{pmatrix} \quad (2)$$

Po sestavení všech parametrů dohromady vznikne rovnice 3, která platí pro nehomogenní les. Parametry terénu a větru jsou zastoupeny koeficientem μ , jehož přesný tvar je uveden v rovnici 4. Množiny V_{adj} a V_{diag} v sumách označují přilehlé, respektive diagonální okolí buňky. Z rovnice a popisu modelu plyne, že výsledné $a_{i,j}$ může vyjít větší než 1, v takovém případě je $a_{i,j}$ přiřazena hodnota 1 [1]. O záporných hodnotách se článek nezmiňuje a i když prakticky nedávají smysl, teoreticky mohou nastat.

$$a_{i,j}^{(t+1)} = \frac{R_{i,j}}{R} a_{i,j} + \sum_{\alpha, \beta \in V_{adj}} \mu_{\alpha, \beta} \frac{R_{i+\alpha, j+\beta}}{R} a_{i+\alpha, j+\beta}^{(t)} + \sum_{\alpha, \beta \in V_{diag}} \mu_{\alpha, \beta} \frac{\pi R_{i+\alpha, j+\beta}^2}{4R^2} a_{i+\alpha, j+\beta}^{(t)} \quad (3)$$

$$\mu_{\alpha, \beta} = w_{i+\alpha, j+\beta} h_{i+\alpha, j+\beta} \quad (4)$$

Zbytek této kapitoly představuje přípravu na zkoušku.

2.2 Základy pravděpodobnosti

Název	Značení	Spojité	Diskrétní
Střední hodnota	$E(X)$	$\int x f(x)$	$\sum s_i p_i$
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

2.3 Generování náhodných čísel

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

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

2.3.1 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.

2.3.2 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: $z_1 \sigma \mu$.

2.3.3 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 .

2.4 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)

2.5 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:

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

2.5.1 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 $\frac{C_a^2 + C_s^2}{2} = 0$.

2.5.2 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 Kriehhoffových 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.

2.6 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 $\frac{1}{\sum \lambda_{odchozi}}$ 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. $K_p = \sum_{ok\ stavy} p_i$, $T_s = \frac{\sum_{ok\ stavy}}{ppst\ prechodu\ do\ neok\ stavu}$.

2.7 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: $\rho = \frac{\text{doba obsluh}}{\text{celkova doba}}$.

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()`.

2.8 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 Erastothenenovo 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 Svět interpretovaný v NetLogu

Svět simulovaný v NetLogu se rozkládá na ploše 256x256 pixelů. Jeden pixel odpovídá čtvercové ploše 10x10 metrů v reálném světě. Model v článku počítá s časovým krokem určeným podle hořlavosti terénu, v mé implementaci je časový krok konstantní (1 minuta) a hořlavost je zadávána v rozmezí $0; 1 > \frac{m}{min}$.

Vzhledem k tomu, že jedna buňka mého modelu má rozměr 10m, je celkový přírůstek spálené plochy ze všech sousedů vynásoben koeficientem $\frac{R_{max}}{10}$. Pokud tedy buňka s hořlavostí 1 bude mít jednoho přilehlého, úplně vyhořelého souseda, je doba potřebná k jejímu vyhoření 10 minut (tiků). Kvůli naakumulované chybě ve floatu však bude potřeba o 1 tik navíc, protože po 10 ticích bude spáleno 99,999999999999% plochy buňky.

Buňka v automatu popsaném článku může nabývat pouze dvou stavů: spálená a nespálená. Pro drobné zpestření jsem přidal dva stavy navíc, které umožňují rozeznat různé stupně spálenosti. Rozsahy a příslušné obarvení je uvedeno v tabulce 3. Pro úplnost je program možné přepnout na režim uvedený v článku, kdy si buňka nechá svou původní barvu až dokud nebude celá spálená. Výhodou tohoto zbarvení je možnost pozorovat jak se oheň šíří mezi buňkami, které jsou jen částečně spálené.

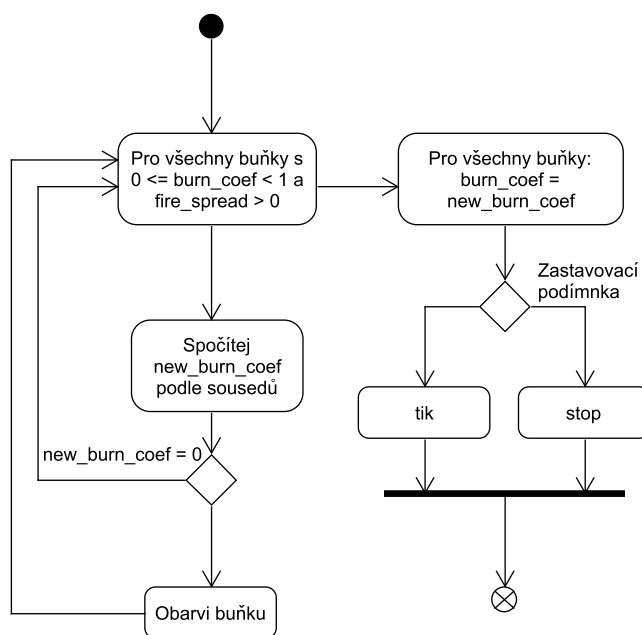
Rozsah spálené plochy buňky a	Obarvení
0	Původní barva
$0 < a \leq 0.1$	Žlutá
$0.1 < a \leq 0.6$	Oranžová
$0.6 < a < 1$	Červená
1	Černá

Tabulka 3: Tabulka obarvení buňky

3.1 Popis programu

Program obsahuje globální proměnné pro sledování celkové spálené plochy, maximální rychlosti šíření ohně a zastavovací podmínku pro testy. Jak již bylo řečeno, program neobsahuje 'želvičky' a pracuje pouze s buňkami světa (patches). Každá buňka nese informaci o velikosti spálené plochy (v čase t a $t + 1$) a rychlosti šíření ohně.

Hlavní procedura programu je *fire-spread-step*, která provede jeden krok evoluce nad buněčným automatem. Činnost procedury je vyjádřena na obrázku 3.



Obrázek 3: Činnost procedury *fire-spread-step*

3.2 Načítání mapy z obrázků

Načítání mapy z obrázků je řešeno v proceduře *setup-image*. Předpokládá se, že obrázek je umístěn ve složce `./img/` a má jméno `source.bmp`. Program rozpoznává několik druhů

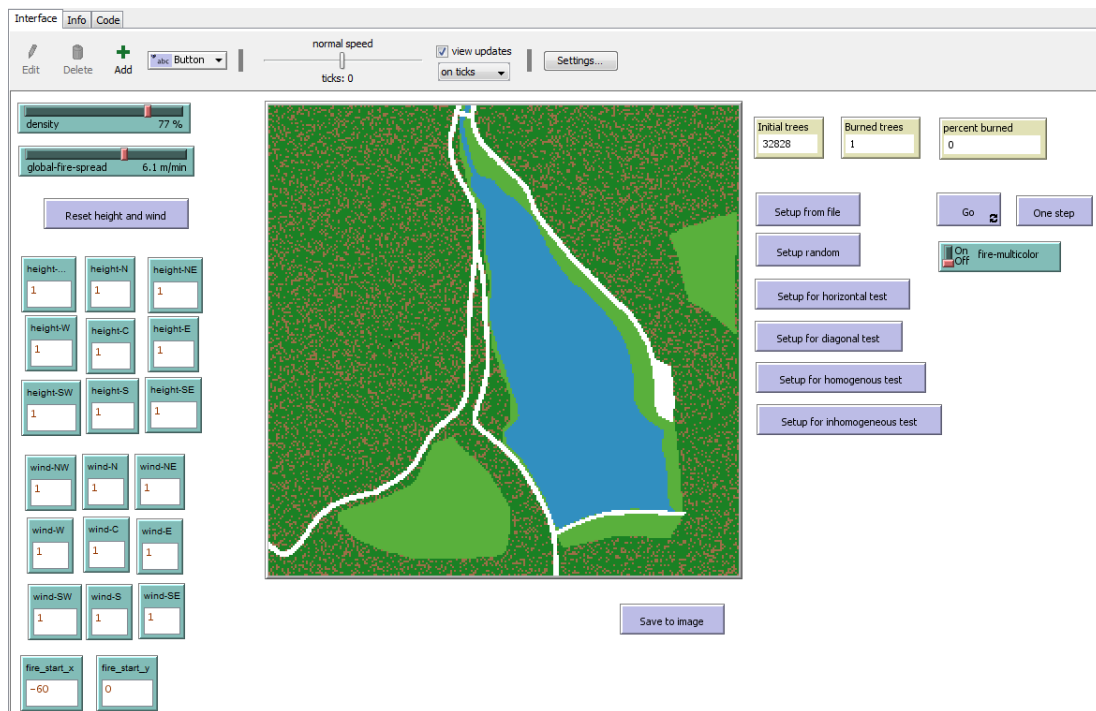
terénu podle barev jednotlivých pixelů, které musí přesně odpovídat barvám definovaným v NetLogu (pokud barva neodpovídá, je buňka nastavená jako nehořlavá). Typy terénu jsou popsány v tabulce 4, hodnotou R nastavuje uživatel.

Barva pixelu NetLogo	Barva pixelu RGB	Typ terénu	R_{teren}
white	0,0,0	Silnice, budova	0
green	89,176,60	Tráva, louka	$2R$
lime - 2	26,129,36	Les	R
sky	45,141,190	Voda	0

Tabulka 4: Vlastnosti a barvy jednotlivých typů terénů

4 Uživatelská příručka

Po otevření programu v NetLogu je zobrazeno ovládací rozhraní (obrázek 4), skrze které je možné nastavení parametrů a ovládání simulace.



Obrázek 4: Program otevřený v Netlogu

4.1 Parametry programu

Prvním z nastavitelných parametrů je hustota zalesnění (*density*), ten určuje poměr buněk představujícím zalesněnou plochu vůči buňkám představujícím plochu bez stromů. V případě načtení mapy ze souboru bude tato hustota aplikována pouze na buňky, které představují les, na jiné materiály se tento parametr nevztahuje.

Druhým parametrem je rychlost šíření ohně u stromů (*global-fire-spread*). Parametr je zadáván v $\frac{m}{min}$ a určuje, jak rychle se bude oheň šířit po zalesněné ploše.

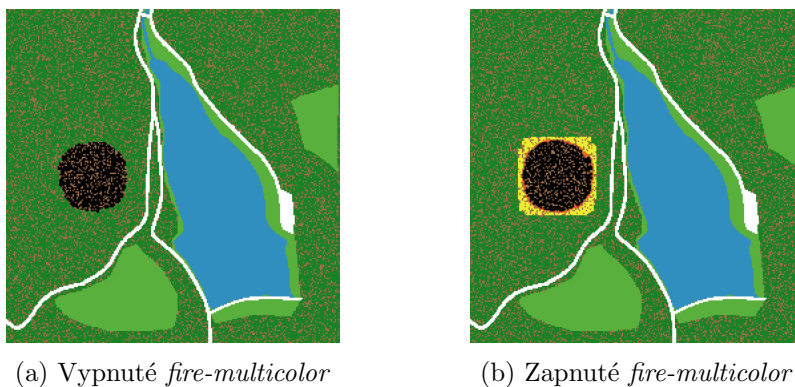
Následují parametry pro zadání matice terénu a větru. Z úsporných důvodů je možné zadat pouze matici lokálního okolí buňky a ne individuální výšku každé buňky. Matice terénu a větru lze zresetovat na hodnoty 1 (tlačítko *Reset height and wind*). Posledním parametrem, který lze zadat je počáteční bod požáru.

4.2 Ovládání simulace

Nastavení před obecnou simulací je možno provést tlačítka *Setup from file*, které načte mapu ze souboru `./img/source.bmp`, a *Setup random*, které vygeneruje náhodný les. Ostatní tlačítka *Setup ** slouží k nastavení testů simulace. Přepínačem *fire-multicolor* lze přepínat mezi režimy obarvení buňky podle spálené plochy. Rozdíly jsou znázorněné na obrázku 5.

Simulace se spouští tlačítkem *Go* a je ukončena, když se oheň nemá jak šířit. Simulaci je také možné krokovat tlačítkem *One step*, které provede jednu iteraci buněčného automatu.

Výsledek simulace je možné uložit do souboru `view.png` tlačítkem *Save to image*. Během simulace je také možné sledovat počet spálených buněk a celkové procento spálené plochy.



Obrázek 5: Srovnání módů obarvení ohnivé plochy

4.3 Spouštění testů

Program obsahuje několik předpřipravených testů, které slouží k ověření implementace. Testy samy nastavují rychlost šíření ohně a parametr hustoty zalesnění ignorují. Testy se spouští stejně, jako obecná simulace tlačítky *Go* a *One step*.

Horizontal test Obsahuje dvě buňky a testuje šíření ohně z jedné buňky (souřadnice -125 0) na jejího souseda (souřadnice -124 0).

Diagonal test Obsahuje dvě buňky a testuje šíření ohně z jedné buňky (souřadnice -125 125) na jejího souseda (souřadnice -124 124).

Homogeneous test Testuje šíření ohně ze středu v homogenním prostředí (rychlost šíření R je stejná pro všechny buňky). Tento test je také použit ve zdrojovém článku.

Inhomogeneous test Testuje šíření ohně ze středu v nehomogenním prostředí. II. a I. kvadrant mají nejnižší R , IV. kvadrant nejvyšší. Tento test je také použit ve zdrojovém článku.

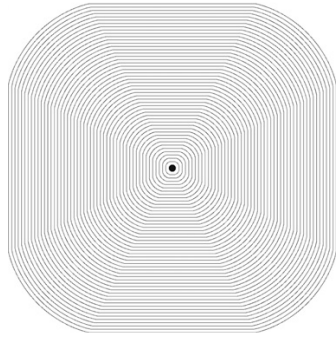
5 Testování

V článku [1] jsou popsány celkem 4 testy, které kombinují parametry větru a terénu s homogenním a nehomogenním prostředím. Pro mé testování jsem použil stejné parametry, liší se pouze velikost světa (v článku 1024x1024, má implementace 256x256) a počtem iterací (článek 500, má implementace 124). Délka buňky je během výpočtu v těchto testech ignorována, výsledná plocha tedy není dělena 10, jak bylo popsáno dříve.

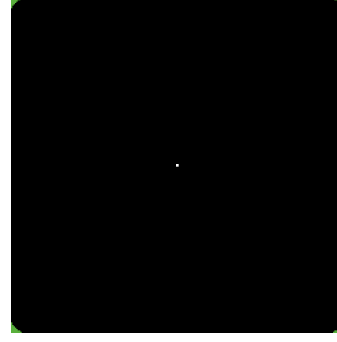
První test uvažuje homogenní prostředí (konstantní R), plochý terén a bezvětří. Matice popisující toto prostředí jsou popsány v 5.

$$\Phi_{i,j} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, W_{i,j} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (5)$$

Srovnání výsledků mého programu s výsledky publikovanými v článku je znázorněno na obrázcích 6a a 6b. Bílá tečka označuje počátek ohně. V mé implementaci je, oproti článku, zaoblení rohů téměř nepatrné.



(a) Výsledek publikovaný v článku



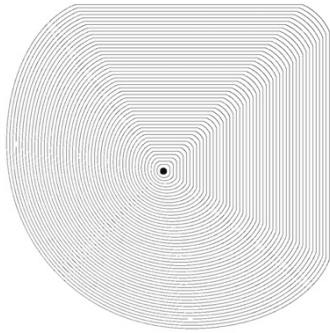
(b) Výsledek mé implementace

Obrázek 6: Výsledky prvního testu

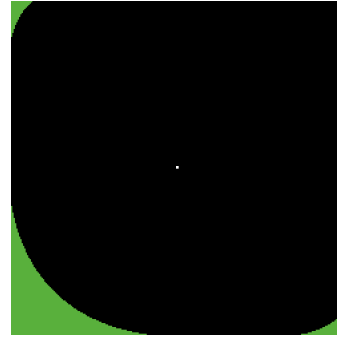
Druhý test opět předpokládá homogenní prostředí, tentokrát je však aplikován vliv větru a členitosti terénu. Matice popisující toto prostředí jsou popsány v 6.

$$\Phi_{i,j} = \begin{pmatrix} 1.5 & 1 & 0.5 \\ 1.5 & 1 & 0.5 \\ 1.5 & 1 & 0.5 \end{pmatrix}, W_{i,j} = \begin{pmatrix} 0.5 & 0.5 & 0.5 \\ 1 & 1 & 1 \\ 1.5 & 1.5 & 1.5 \end{pmatrix} \quad (6)$$

Srovnání výsledků mého programu s výsledky publikovanými v článku je znázorněno na obrázcích 7a a 7b. Bílá tečka označuje počátek ohně. Opět je zřejmé, že zaoblení v mé implementaci není tak intenzivní jako v případě článku. Zároveň je kruhová část otočená na opačnou stranu, což může plynout z obrácené aplikace matice s výškou terénu.



(a) Výsledek publikovaný v článku

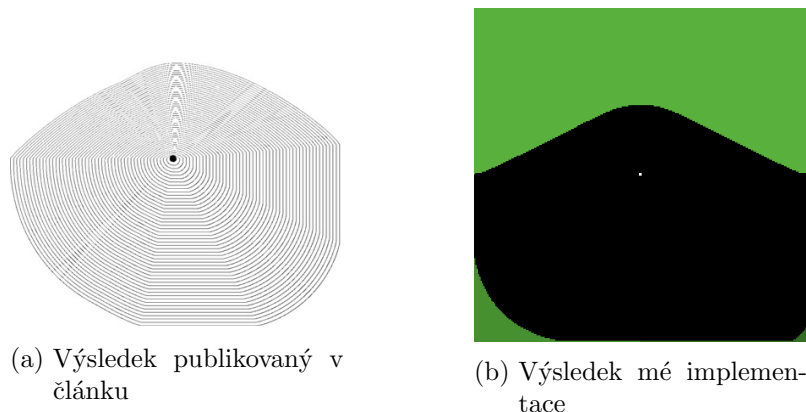


(b) Výsledek mé implementace

Obrázek 7: Výsledky druhého testu

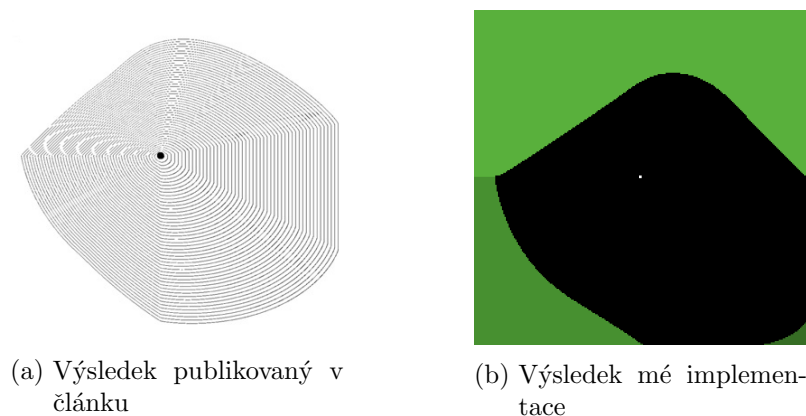
Třetí test předpokládá následující nehomogenní prostředí: I. a II. kvadrant mají $R = 1$, III. kvadrant má $R = 3$ a IV. kvadrant má $R = 4.5$. Vítr ani členitost terénu zatím nejsou aplikovány (jsou tedy použity matice z 5). Srovnání výsledků mého programu s výsledky

publikovanými v článku je znázorněno na obrázcích 8a a 8b. V tomto případě se výsledky víceméně shodují.



Obrázek 8: Výsledky třetího testu

Poslední test předpokládá nehomogenní prostředí, stejné jako v předchozím případě, a navíc jsou aplikovány i vlivy větru a členitosti terénu (matice 6). Srovnání výsledků mého programu s výsledky publikovanými v článku je znázorněno na obrázcích 9a a 9b. Z obrázků je patrné, že se výsledky liší pouze v zaoblení.



Obrázek 9: Výsledky čtvrtého testu

6 Závěr

Model se mi podařilo úspěšně na-implementovat a testy mé implementace dopadly obdobně jako testy uvedené v článku.

Reference

- [1] ENCIAS, Hernández a WHITE, Hoya a del RAY, Martín a SANCHÉZ, Rodríguez . 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>
- [2] KARAFYLLIDIS, Ioannis a THANAILAKIS, Adonios. A model for predicting forest fire spreading using cellular automata. *Ecol Model* 1997, 99, 87–97.