

Prostředí eko-kolonií a P-kolonií

Šárka Vavrečková

Ústav informatiky, Slezská univerzita v Opavě

Bezručovo nám. 13, Opava

Email: sarka.vavreckova@fpf.slu.cz

Abstrakt

Existuje řada různých matematických modelů, ve kterých více či méně nezávislí agenti zasahují do sdíleného prostředí či s ním přímo interagují. V článku srovnáváme působení agentů na prostředí ve dvou matematických modelech – eko-koloniích a P koloniích. Eko-P kolonie jsou variací P kolonií povolující vývoj prostředí, proto byly do srovnání také zařazeny. Vzhledem k výrazně jednodušší struktuře agentů eko-kolonií byl zvolen postup napodobení činnosti agentů eko-kolonie v P kolonii, resp. v eko-P kolonii.

1 Úvod

Eko-kolonie byly představeny v Vavrečková (2005) jako rozšíření kolonií (Kelemen a Kelemenová, 1992) – gramatických systémů s velmi jednoduchými gramatikami (komponentami, agenty) spolupracujícími na sdíleném prostředí. Eko-kolonie obohacují tento koncept o možnost vývoje prostředí. V eko-koloniích je prostředí určeno 0L nebo E0L schématem, podle toho hovoříme o 0L eko-koloniích nebo E0L eko-koloniích.

Agenti dle svých pravidel zpracovávají prvky prostředí, zbývající prvky podléhají vývoji prostředí. Agent dokáže zpracovávat jen jeden konkrétní prvek (který v každém kroku vyhledává v prostředí, tento prvek nazýváme „startovacím symbolem“ agenta), jeho pravidla stanovují řetězce, na které agent přepíše nalezený výskyt svého prvku v prostředí. Důležitým omezením je, že agent nedokáže vygenerovat svůj startovací symbol (tento prvek se nesmí nacházet na pravé straně žádného pravidla tohoto agenta).

Ve zdroji Vavrečková (2005) jsou pro eko-kolonie definovány dva módy odvození – slabě paralelní *wp* a plně paralelní *ap*. V obou případech pracují agenti paralelně, ale zatímco u slabého paralelismu pracují právě ti agenti, kterým to stav prostředí dovolí (ti, kteří v prostředí najdou svůj „volný“ symbol), u plného paralelismu pracují v každém kroku všichni agenti (pokud to není možné, výpočet končí).

Slabý paralelismus byl postupně definován několika různými způsoby (třebaže v důsledku jsou tyto způsoby navzájem ekvivalentní, měnil se spíše způsob zápisu definice), zde se přidržíme definice publikované ve zdroji Kelemenová a Vavrečková (2011).

V tomto článku se z důvodu srovnatelnosti (podobnosti) budeme zabývat pouze 0L eko-koloniemi pracujícími formou slabého paralelismu.

Co se P kolonií týče, jejich základním kamenem je membránový systém (struktura vzájemně vnořených membrán s procházejícími a transformovanými množinami objektů), či lépe P systém (membránový systém s jedinou membránou) představený ve zdroji Páun (2000).

Přímo s konceptem P kolonií se poprvé setkáváme v Kelemen a Kelemenová (2005), kde jsou představeny jako formální model inspirovaný membránovými systémy a výše zmíněnými koloniemi. P systémy zde hrají roli agentů koexistujících ve sdíleném prostředí. Každý agent je vybaven množinou objektů určujících jeho stav a sadou programů. V každém programu z této sady je stanovený počet pravidel, pravidla mohou být několika různých typů. Pravidla buď mění stav agenta nebo umožňují agentovi ovlivňovat stav prostředí.

Dostačující informace o variantách a také o možnostech využití membránových systémů, P systémů, P kolonií a dalších modelů najdeme zejména na webové stránce <http://ppage.psyste.ms.eu/>.

Eko-P kolonie jsou rozšířením P kolonií, kde je umožněn vývoj prostředí (prostředí je určeno 0L schématem) a agenti používají pouze dva typy programů: generující a konzumující. Oba typy programů mění stav agenta, generující programy navíc distribuují jeden symbol ze stavu agenta do prostředí, naopak konzumující programy přenášejí jeden symbol z prostředí do stavu agenta. Informace o eko-P koloniích jsou ze zdrojů Cienciala a Ciencialová (2011) a Cienciala (2015).

Upozorňujeme, že v tomto článku není v žádném případě řešena výpočetní síla uvedených matematických modelů či její srovnání, zabýváme se pouze a jen srovnáním vlivu agentů na prostředí. Protože o tomto tématu by bylo možno psát velmi obsáhle, omezíme se pouze na dva algoritmy ukazující, jak lze podle dané eko-kolonie (v prvním případě splňující určité omezující podmínky, v druhém případě obecné) sestavit P kolonii respektive eko-P kolonii takovou, že agenti se ve výsledném systému budou ke sdílenému prostředí chovat s podobným výsledkem jako v původní eko-kolonii. Určitým handicapem je odlišné pojetí prostředí v těchto systémech, tedy o skutečné ekvivalenci mezi vstupy a výstupy uvedených algoritmů samozřejmě nelze uvažovat.

2 Prerekvizity

Budeme předpokládat, že čtenář již ovládá alespoň základy teoretické informatiky. Zde je stanoveno značení, které bude používáno v následujícím textu, dále v podkapitolách jsou definice všech srovnávaných modelů: eko-kolonií, P kolonií a eko-P kolonií.

Multimnožina je uspořádaná dvojice $M = (V, f)$, kde V je základní množinou multimnožiny obsahující objekty vyskytující se v M , f je zobrazení $f: V \rightarrow \mathcal{N}$, které každému prvku množiny V přiřazuje jeho četnost v multimnožině M . Množinu všech multimnožin nad množinou V označíme V° .

2.1 Eko-kolonie

Eko-kolonie jsou matematický model vycházející z ekogramatických systémů. Existují ve variantách s OL nebo EOL prostředím (vývoj prostředí je určen OL nebo EOL schématem), zde se přidržíme varianty s prostředím OL.

Agent v eko-kolonii je jednoduchý systém generující defacto konečný jazyk, interakce s prostředím je pouze jednosměrná (agent ovlivňuje prostředí). Následuje formální definice OL eko-kolonie.

Definice 1 (OL Eko-kolonie) OL eko-kolonie stupně n je uspořádaná struktura $\Sigma = (E, w_0, A_1, A_2, \dots, A_n)$, kde

- $E = (V, P)$ je OL schéma, přičemž
 - V je neprázdná konečná abeceda prostředí,
 - $P \subseteq V \times V^*$ je konečná úplná množina vývojových pravidel prostředí typu OL,
- $w_0 \in V^*$ je počáteční stav eko-kolonie, axiom,
- $A_i = (S_i, F_i)$ je i -tý agent, $1 \leq i \leq n$, kde
 - $S_i \in V$ je startovací symbol agenta,
 - $F_i \subseteq (V - \{S_i\})^*$ je neprázdná konečná množina akčních pravidel agenta.

Vlastnost	Eko-kolonie
prostředí $E = (V, P)$	dynamické, typu OL, stav je řetězec
agenti $A = (S, F)$	jsou bezstavoví, zpracovávají jeden konkrétní symbol v prostředí, přepisují na řetězec
pravidla agentů	$S \rightarrow f_1 \mid f_2 \mid \dots \mid f_k$, $f_j \in F, F \subseteq (V - S)^*$
přednost	agenti mají přednost před vývojem prostředí
činnost agentů	paralelní
výstup	řetězec – stav prostředí

Tab. 1: Souhrn vlastností eko-kolonií

V eko-koloniích můžeme použít několik různých módů odvození, v tomto článku se omezíme jen na slabý paralelismus (wp – weakly parallel), kdy pracují paralelně všichni agenti, kterým to stav prostředí umožňuje. i -tý agent v jednom kroku najde v prostředí svůj startovací symbol S_i a přepíše ho na některé ze slov z množiny F_i . Symboly prostředí, které v daném kroku nejsou zpracovány žádným agentem, podléhají vývojovým pravidlům prostředí.

Definice kroku odvození wp je následující:

Definice 2 Necht $\Sigma = (E, w_0, A_1, A_2, \dots, A_n)$ je OL eko-kolonie. Definujeme slabě paralelní krok odvození v Σ jako relaci \xrightarrow{wp} takovou, že $\alpha \xrightarrow{wp} \beta$, jestliže

- $\alpha = v_0 S_{i_1} v_1 S_{i_2} v_2 \dots v_{r-1} S_{i_r} v_r, r > 0$,
- $\beta = v'_0 f_{i_1} v'_1 f_{i_2} v'_2 \dots v'_{r-1} f_{i_r} v'_r$
pro $A_{i_k} = (S_{i_k}, F_{i_k}), f_{i_k} \in F_{i_k}, 1 \leq k \leq r$,
- $\{i_1, \dots, i_r\} \subseteq \{1, \dots, n\}, i_k \neq i_m$ pro všechna $k \neq m, 1 \leq k, m \leq r$,
- pro všechny symboly $S \in V$ označme t_S počet agentů přepisujících symbol S ; potom

$$\sum_{j=1}^r \text{Eval}(S_{i_j} = S) = \min(|\alpha|_S, t_S) \quad (1)$$

kde funkce Eval() vrací 1 právě tehdy, když platí $S_{i_j} = S$; součet na levé straně rovnice představuje počet výskytů symbolu S zpracovaných agenty v daném kroku odvození (pracují všichni agenti, kteří mohou pracovat),

- $v_k \Rightarrow v'_k, v_k \in V^*, 0 \leq k \leq r$ je krok odvození v OL schématu E .

Výpočet v eko-kolonii končí tehdy, když žádný agent nemůže pracovat, výstupem (výsledkem činnosti eko-kolonie) je stav prostředí.

2.2 P kolonie

P kolonie jsou biologicky motivovaný matematický model vycházející z membránových systémů (P systémů) používajících pouze jednu membránu s prvky gramatických systémů.

V prostředí, které je statické (bez vlastního vývoje), se nacházejí agenti – jednoduché membránové systémy s vlastním stavem a několika sadami pravidel, kterým říkáme programy. Stav prostředí není řetězec, ale multimnožina, totéž platí o stavech agentů. Stav agentů obsahují vždy přesně daný počet prvků, tento počet se během výpočtu nemění.

Interakce s prostředím je obousměrná – existují komunikační pravidla sloužící k výměně objektů mezi stavem daného agenta a stavem prostředí, nicméně aktivita je pouze na straně agenta.

Definice 3 (P kolonie) *P kolonie s kapacitou k je struktura $\Pi = (A, e, f, V_E, B_1, \dots, B_n)$, kde*

- A je neprázdná konečná abeceda,
- $e \in A$ je základní (environmentální) objekt,
- $f \in A$ je finální objekt,
- $V_E \in (A - e)^o$ je počáteční stav prostředí,
- $B_i = (O_i, P_i)$ je i -tý agent, $1 \leq i \leq n$, kde
 - O_i je multimnožina nad A , představuje stav agenta, platí $|O_i| = k$,
 - $P_i = \{p_{i,1}, p_{i,2}, \dots, p_i, r_i\}$ je konečná sada programů agenta, každý program obsahuje právě k pravidel v této formě:
 - * $a \rightarrow b$ – evoluční pravidla,
 - * $c \leftrightarrow d$ – komunikační pravidla,
 - * p_1/p_2 , kde p_1, p_2 jsou evoluční nebo komunikační pravidla – kontrolní pravidla.

Počáteční konfigurace se skládá z počátečního stavu prostředí a počátečních stavů jednotlivých agentů.

Agenti v P kolonii mohou pracovat sekvenčně nebo paralelně, my se přidržíme paralelního způsobu práce (v jednom kroku pracují všichni agenti, kteří mohou pracovat, v podobném smyslu jako v módu *wp* popsaném výše pro eko-kolonii).

Vlastnost	P kolonie
prostředí V_E	statické, stav je multimnožina
agenti $B = (O, P)$	mají stav, několik typů pravidel vč. komunikace s prostředím, ovlivňují prostředí v závislosti na svém stavu
pravidla agentů	<ul style="list-style-type: none"> • $a \rightarrow b$, $a \in O$ (evoluční) • $c \leftrightarrow d$, $c \in O$, $d \in V_E$ (komunikační) • p_1/p_2, kde p_1, p_2 jsou evoluční nebo komunikační pravidla (kontrolní)
přednost	agenti mají přednost, prostředí se nevyvíjí
činnost agentů	sekvenční i paralelní
výstup	číslo – četnost prvků f v prostředí

Tab. 2: Souhrn vlastností P kolonií

Agent může ovlivňovat stav prostředí svými *komunikačními pravidly* (pravidlo ve tvaru $c \leftrightarrow d$ se aplikuje tak, že prvek c ve stavu agenta vyměníme s prvkem d ve stavu prostředí). *Evoluční pravidla* slouží pouze k vývoji samotného agenta. *Kontrolní pravidla* vnášejí do systému podmíněnost – pokud nelze uplatnit první uvedené pravidlo, je použito druhé uvedené pravidlo.

Každý agent má sadu programů (každý agent může mít jiný počet programů), v jednom kroku odvození lze použít jen jeden program. V každém programu je právě k pravidel (kde číslo k nazýváme kapacitou P kolonie, určuje i počet prvků ve stavu agenta) a program je proveden tehdy, pokud jsou všechna jeho pravidla aplikovatelná (a všechna pravidla programu jsou aplikována paralelně).

Výpočet končí tehdy, když žádný agent nemůže pracovat, výstupem je počet prvků f v prostředí.

2.3 Eko-P kolonie

Eko-P kolonie byly inspirovány P koloniemi s těmito změnami:

- prostředí se vyvíjí, je to 0L schéma,
- kapacita k je vždy 2,
- je definován jiný typ pravidel – generující (insertion) a konzumující (delation).

Počítá se pouze s paralelní činností agentů.

Definice 4 (Eko-P kolonie) *Eko-P kolonie je struktura $\Pi = (A, e, f, V_E, D_E, B_1, \dots, B_n)$, kde*

- A je neprázdná konečná abeceda,
- $e \in A$ je základní (environmentální) objekt,
- $f \in A$ je finální objekt,
- $V_E \in (A - e)^o$ je počáteční stav prostředí,
- D_E je 0L schéma (A, P_E) , kde P_E je množina bez-kontextových pravidel,
- $B_i = (O_i, P_i)$ je i -tý agent, $1 \leq i \leq n$, kde
 - O_i je multimnožina nad A , představuje stav agenta, platí $|O_i| = 2$,
 - $P_i = \{p_{i,1}, p_{i,2}, \dots, p_i, r_i\}$ je konečná sada programů agenta, každý program je jednoho z těchto typů:
 - * $\langle a \rightarrow bc, d \text{ out} \rangle$ – generující (insertion) pravidla,
 - * $\langle ab \rightarrow c, d \text{ in} \rangle$ – konzumující (delation) pravidla.

Počáteční konfigurace se skládá z počátečního stavu prostředí a počátečních stavů jednotlivých agentů.

V původním pojetí existují v eko-P kolonii dva typy agentů – agenti generující (obsahující pouze první typ programů) a agenti konzumující (obsahující pouze druhý typ programů). V tomto článku se přidržíme volnější definice, kdy agent může mít oba typy programů, nebudeme trvat na separaci agentů podle kompetencí.

Eko-P kolonie nás zde zajímají mnohem více než klasické P kolonie, a to nejen vzhledem k možnosti vývoje prostředí.

Vlastnost	Eko-P kolonie
prostředí	dynamické, stav je multimnožina, ale chová se jako 0L systém
agenti $B = (O, P)$	mají stav, dva typy pravidel, ovlivňují prostředí v závislosti na svém stavu
pravidla agentů	<ul style="list-style-type: none"> $\langle a \rightarrow bc, d \text{ out} \rangle, a, d \in O$ (generující) $\langle ab \rightarrow c, d \text{ in} \rangle, a, b \in O, d \in V_E$ (konzumující)
přednost	agenti mají přednost před vývojem prostředí
činnost agentů	paralelní
výstup	číslo – četnost prvků f v prostředí

Tab. 3: Souhrn vlastností eko-P kolonií

3 Eko-kolonie vs. P kolonie

Eko-kolonie a P kolonie pojmají prostředí každá jinak. V obou případech do prostředí zasahují agenti, ale v eko-koloniích se prostředí dynamicky vyvíjí, kdežto v P koloniích je statické. Navíc celkové pojetí prostředí je odlišné – v eko-koloniích se jedná o řetězec s přiřazeným 0L schématem určujícím jeho vývoj, v P koloniích je prostředí multimnožinou, ve které nebereme v úvahu pořadí prvků.

Z toho důvodu jsou pravidla agentů P kolonie velmi jednoduchá, zpracovávají vždy jen jeden objekt (ať už svůj nebo v prostředí), a pokud generují, pak opět jen jeden objekt. Naproti tomu pravidla agentů eko-kolonie generují obecně řetězec, jehož prvky jsou vázány svým pořadím, v prostředí je tento řetězec zařazen na konkrétní místo v prostředí, které je taktéž striktně určeno.

Do jaké míry lze srovnat, připodobnit a případně simulovat činnost agentů v prostředí eko-kolonií a P kolonií? Pokud odhlédneme od odlišnosti v chápání prostředí a výsledku výpočtu (řetězec vs. číslo), najdeme podobnost a základ pro simulaci v následujícím: vezměme eko-kolonii $\Sigma = (E, w_0, A_1, \dots, A_n)$, kde

- prostředí $E = (V, P)$ je statické, P obsahuje pouze pravidla typu $a \rightarrow a, \forall a \in V$,
- agenti $A_i = (S_i, F_i), 1 \leq i \leq n$ mají omezené vlastnosti, generují pouze slova o délce 1: $F_i = \{f_{i,1}, \dots, f_{i,r_i}\}, f_{i,j} \in (V - S_i), 1 \leq j \leq r_i$.

Sestrojíme P kolonii $\Pi(\Sigma)$:

- agenti B_i budou vytvořeni podle agentů A_i tak, aby jejich činnost v prostředí byla obdobná,
- kapacita P kolonie bude 1,
- jednomu kroku činnosti původní eko-kolonie bude odpovídat dvojice kroků činnosti P kolonie,
- každý agent B_i bude mít r_i dvojic sad programů, pro každý dvojkrok bude vybrána jedna z těchto

dvojic, jeden program z dvojice použije v „lichém“ kroku činnosti P kolonie, druhý (k němu párový) v „sudém“ kroku,

- počáteční stav prostředí přejmeme od eko-kolonie, počáteční stav agenta B_i vytvořeného podle agenta $A_i = (S_i, F_i)$ bude $S_i \in V$.

Celý postup převodu je popsán v algoritmu 1.

Algoritmus 1: Eko-kolonie \rightarrow P kolonie

Vstup:

$\Sigma = (E, w_0, A_1, \dots, A_n)$
 prostředí $E = (V, P)$, statické,
 agenti $A_i = (S_i, F_i), F_i = \{f_{i,1}, \dots, f_{i,r_i}\},$
 $f_{i,j} \in (V - S_i), 1 \leq j \leq r_i,$
 počáteční konfigurace je $w_0 \in V^*$,
 označme $|w_0| = d$

Výstup:

$\Pi(\Sigma) = (A, e, f, V_E, B_1, \dots, B_n)$
 agenti $B_i = (O_i, P_i),$
 počáteční konfigurace je
 $(V_E, O_1, \dots, O_n),$
 kapacita P kolonie je $k = 1$

Prostředí:

Abeceda: $A = V \cup \{e, f\}$
 $V_E = \bigcup_{1 \leq j \leq d} a_j,$ kde $w_0 = a_1, \dots, a_d$

end

Agenti: $B_i = (O_i, P_i), 1 \leq i \leq n$

$O_i = S_i$ (kapacita je 1)
 P_i sestrojíme podle $F_i = \{f_{i,1}, \dots, f_{i,r_i}\}:$
 $P_i = \bigcup_{1 \leq j \leq r_i} \{p_{i,ja}, p_{i,jb}\},$ kde

for j pro $1 \leq j \leq r_i$ do

$p_{i,ja} = \langle S_i \rightarrow f_{i,j} \rangle$
 $p_{i,jb} = \langle f_{i,j} \leftrightarrow S_i / f_{i,j} \rightarrow S_i \rangle$

end

end

Nastíháme si, jakým způsobem bude simulováno použití pravidla $S_i \rightarrow f_{i,j}$ agenta A_i z eko-kolonie prostřednictvím agenta B_i v P kolonii. Podle algoritmu 1 je pravidlo agenta B_i pro „lichý“ krok $\langle S_i \rightarrow f_{i,j} \rangle$, to znamená, že agent změní svůj vlastní stav z S_i na některý z objektů, které sám může generovat (z množiny F_i), čímž nedeterministicky vybere některé z pravidel původního agenta A_i z eko-kolonie.

V dalším kroku nastane jedna ze dvou situací – v prostředí se buď vyskytuje nebo nevyskytuje symbol S_i . Pokud se vyskytuje, použijeme komunikační pravidlo $f_{i,j} \leftrightarrow S_i$ (konkrétní pravidlo původního agenta A_i z eko-kolonie bylo zvoleno v předchozím kroku), pokud ne, použijeme evoluční pravidlo $f_{i,j} \rightarrow S_i$. V prvním případě jsme zajistili přepis symbolu S_i v prostředí na symbol $f_{i,j}$, přičemž stav agenta B_i je nyní S_i stejně jako před lichým krokem, v druhém případě jde u agenta jednoduše o návrat do původního stavu s tím, že k přepisu v prostředí nedošlo.

Kroky:	lichý	sudý
--------	-------	------

Symbol S_i je v prostředí:

prostředí	$\alpha S_i \beta \Rightarrow \alpha S_i \beta \Rightarrow \alpha' f_{i,j} \beta'$
agent B_i	$S_i \Rightarrow f_{i,j} \Rightarrow S_i$

Symbol S_i není v prostředí:

prostředí	$\alpha \Rightarrow \alpha \Rightarrow \alpha'$
agent B_i	$S_i \Rightarrow f_{i,j} \Rightarrow S_i$

Tab. 4: Stav prostředí a agentů v P kolonii

Nyní je třeba ukázat, že

1. agent B_i v P kolonii pracuje právě tehdy, kdy pracuje agent A_i v eko-kolonii,
2. v sudém kroku výpočtu volí právě takový program, který patří do páru k programu zvolenému v předchozím (lichém) kroku výpočtu.

Počáteční stav agenta B_i (pro některé $i \in \{1, \dots, n\}$) je S_i . Pouze program s pravidlem $S_i \rightarrow f_{i,j}$ pro některé $i \in \{1, \dots, r_i\}$ je možné v tomto stavu provést, tedy každý agent (bez ohledu na to, zda se symbol S_i v prostředí vyskytuje) v prvním kroku provede takové pravidlo.

Po prvním kroku je stav agenta $f_{i,j}$. V druhém kroku výpočtu je tento agent nucen zvolit pravidlo podle svého stavu, což je právě jedno z pravidel v programu $\langle f_{i,j} \leftrightarrow S_i / f_{i,j} \rightarrow S_i \rangle$. První (komunikační) má přednost, a tedy pokud se v prostředí vyskytuje symbol S_i , bude provedeno (čímž je simulována činnost agenta A_i z eko-kolonie). Druhé se provede tehdy, když první není možno aplikovat, což také odpovídá činnosti (resp. ve finále nečinnosti) agenta eko-kolonie v dané situaci při použití paralelního módu přepisování.

V druhém kroku výpočtu nemůže být zvolen jiný program než „párový“ k programu z předchozího kroku, protože zpracovává symbol $f_{i,j}$ v prostředí agenta, žádné jiné pravidlo tento symbol nezpracovává. Z toho také vyplývá, že liché a sudé kroky se nemohou míchat.

Průběh výpočtu ve vytvořené P kolonii odpovídá průběhu výpočtu v (poněkud osekané) původní eko-kolonii se statickým prostředím a s agenty generujícími řetězce o délce 1, přičemž každému kroku v eko-kolonii odpovídají dva kroky v P kolonii.

Nyní se zaměříme na ukončení výpočtu. V eko-kolonii výpočet končí tehdy, když žádný agent nemůže pracovat, v P kolonii by mělo platit totéž. Zde nastává problém – každý agent B_i ve skutečnosti může pracovat kdykoliv, a pokud v prostředí nenajde svůj symbol S_i , bude v nekonečné smyčce jednoduše používat programy $\langle S_i \rightarrow f_{i,j} \rangle$, $\langle \dots / f_{i,j} \rightarrow S_i \rangle$ pro některá $j \in \{1, \dots, r_i\}$.

V důsledku jsme tedy schopni simulovat alespoň činnost zjednodušené eko-kolonie P kolonií, což nám vzhledem k zaměření na prostředí systému stačí.

4 Eko-kolonie vs. eko-P kolonie

Už podle definice se mezi eko-koloniemi a eko-P koloniemi dal očekávat bližší vztah než v předchozím srovnání, což se potvrdilo, třebaže oba modely vznikly na sobě zcela nezávisle a jsou jinak poměrně rozdílné.

V obou případech máme dynamické prostředí, tedy prostředí není ovlivňováno pouze agenty. Nicméně odlišností je zde také dostatek, víceméně všechny odlišnosti (kromě role prostředí) můžeme přejmout z předchozí kapitoly. Pravidla agentů eko-P kolonie jsou sice jiná než u P kolonií, ovšem pravidlům eko-kolonií se neblíží.

Přesto už možnost vývoje prostředí nám pomůže k tomu, abychom mohli při přechodu z eko-kolonie k eko-P kolonii implementovat plně definované agenty (jejich množiny F_i již mohou obsahovat různé dlouhé řetězce, nikoliv pouze symboly).

Vezměme eko-kolonii $\Sigma = (E, w_0, A_1, \dots, A_n)$ s agenty, kde

- prostředí $E = (V, P)$ je dynamické, jedná se o OL schéma,
- agenti $A_i = (S_i, F_i)$, $1 \leq i \leq n$ generují slova svých jazyků pravidly $F_i = \{f_{i,1}, \dots, f_{i,r_i}\}$, $f_{i,j} \in (V - S_i)^*$, $1 \leq j \leq r_i$.

Sestrojíme eko-P kolonii $\Pi(\Sigma)$:

- agenti B_i budou vytvořeni podle agentů A_i tak, aby jejich činnost v prostředí byla obdobná,
- jednomu kroku činnosti původní eko-kolonie bude odpovídat trojice kroků činnosti eko-P kolonie,
- každý agent B_i bude mít r_i trojic sad programů, pro každý trojkrok bude vybrána jedna z těchto trojic (ve skutečnosti třetí program v každé sadě bude vždy stejný, jeho úkolem bude vrátit agenta do „startovacího stavu“, takže celkový počet programů agenta B_i je $2 * r_i + 1$),
- činnost agentů A_i eko-kolonie bude částečně suplovat prostředí – agent pouze pošle prostředí informaci, které pravidlo původního agenta A_i (resp. jeho provedení) má být simulováno, a to symbolem $[i, j]$ (i -tý agent, jeho j -té pravidlo), prostředí zajistí přepis tohoto symbolu na řetězec $f_{i,j}$ (tím obejdeme omezení, dle kterého agent může generovat vždy jen jeden symbol),
- počáteční stav prostředí přejmeme od eko-kolonie, počáteční stav agenta B_i vytvořeného podle agenta $A_i = (S_i, F_i)$ bude $S_i X$.

Celý postup převodu je popsán v algoritmu 2.

Krok výpočtu eko-kolonie je v eko-P kolonii simulován posloupností tří kroků. Ukážeme, že

- v prvním z této trojice kroků pracují právě ti agenti B_i , jejichž symbol S_i se nachází v prostředí, přičemž tento symbol již nebyl zabrán jiným agentem,

Kroky:	první	druhý	třetí
<i>Symbol S_i je v prostředí:</i>			
prostředí	$\alpha S_i \beta \Rightarrow$	$\alpha' S_i \beta' \Rightarrow$	$\alpha'' [i, j] \beta'' \Rightarrow \gamma f_{i,j} \delta$
agent B_i	$S_i X \Rightarrow$	$S_i [i, j] \Rightarrow$	$S_i f \Rightarrow S_i X$
<i>Symbol S_i není v prostředí:</i>			
prostředí	$\alpha \beta \Rightarrow$	$\alpha' \beta' \Rightarrow$	$\alpha'' \beta'' \Rightarrow \gamma \delta$
agent B_i	$S_i X \Rightarrow$	$S_i X \Rightarrow$	$S_i X \Rightarrow S_i X$

Tab. 5: Stav prostředí a agentů v eko-P kolonii

Algoritmus 2: Eko-kolonie \rightarrow eko-P kolonie

Vstup:

$\Sigma = (E, w_0, A_1, \dots, A_n)$
 prostředí $E = (V, P)$ je OL schéma,
 agenti $A_i = (S_i, F_i)$, $F_i = \{f_{i,1}, \dots, f_{i,r_i}\}$,
 $f_{i,j} \in (V - S_i)^*$, $1 \leq j \leq r_i$,
 počáteční konfigurace je $w_0 \in V^*$,
 označme $|w_0| = d$

Výstup:

$\Pi(\Sigma) = (A, e, f, V_E, D_E, B_1, \dots, B_n)$
 agenti $B_i = (O_i, P_i)$,
 počáteční konfigurace je
 (V_E, O_1, \dots, O_n)

Prostředí:

$A = V \cup \{e, f, X\} \cup \{a', a''; a \in V\}$
 $\cup \{[i, j]; 1 \leq i \leq n, 1 \leq j \leq r_i\}$
 $V_E = \bigcup_{1 \leq j \leq d} a_j$, kde $w_0 = a_1, \dots, a_d$,
 $D_E = (A - \{X\}, P_E)$, kde
 $P_E = \{a \rightarrow a', a' \rightarrow a'', a'' \rightarrow \beta;$
 kde $(a \rightarrow \beta) \in P_E\}$
 $\cup \{f \rightarrow f\} \cup \{e \rightarrow e\}$
 $\cup \{[i, j] \rightarrow f_{i,j}; 1 \leq i \leq n, f_{i,j} \in F_i\}$

end

Agenti: $B_i = (O_i, P_i)$, $1 \leq i \leq n$

$O_i = S_i X$
 P_i sestrojíme podle $F_i = \{f_{i,1}, \dots, f_{i,r_i}\}$:
 $P_i = \{p_{i,c}\} \cup \left(\bigcup_{1 \leq j \leq r_i} \{p_{i,ja}, p_{i,jb}\} \right)$, kde
for j *pro* $1 \leq j \leq r_i$ **do**
 $p_{i,ja} = \langle S_i X \rightarrow [i, j], S_i \text{ in} \rangle$
 $p_{i,jb} = \langle S_i \rightarrow S_i f, [i, j] \text{ out} \rangle$
end
 $p_{i,c} = \langle S_i \rightarrow S_i X, f \text{ out} \rangle$

end

- v druhém a třetím kroku pracují právě ti agenti, kteří pracovali v prvním kroku,
- jeden krok vývoje prostředí eko-kolonie odpovídá třem krokům vývoje prostředí v eko-P kolonii, přičemž je zachován koncept přednosti agentů před vývojem prostředí.

Popíšeme, jak se chovají jednotliví agenti, a jak reaguje prostředí. Vezměme některého agenta B_i , $1 \leq i \leq n$:

Krok 1. Mohou pracovat jen ti agenti, kteří v prostředí najdou „volný“ ten prvek, který dokážou zpracovat, tedy S_i . Agent B_i nedeterministicky vybere některý z prvků původní množiny F_i a zároveň si z prostředí vyzvedne jeden symbol S_i , a to použitím programu $\langle S_i X \rightarrow [i, j], S_i \text{ in} \rangle$. Důsledkem je změna stavu agenta z $S_i X$ na $S_i [i, j]$.

Agent, který v prostředí nenajde symbol S_i , v tomto kroku nepracuje, protože

- programy typu $p_{i,ja}$ (podle algoritmu 2) nejsou aplikovatelné – v prostředí není žádný výskyt symbolu S_i ,
- programy typu $p_{i,jb}$ nejsou aplikovatelné – ve stavu agenta není symbol $[i, j]$, který by měl být takovým programem distribuován do prostředí,
- program $p_{i,c}$ není aplikovatelný – ve stavu agenta není symbol f , který by měl být distribuován do prostředí.

Prostředí samotné se v tomto kroku vyvíjí podle pravidel typu $a \rightarrow a'$, poněvadž se v jeho stavu nacházejí jen symboly původní abecedy $a \in V$. Po provedení tohoto kroku najdeme v prostředí pouze symboly a' , $a \in V$.

Krok 2. Agent B_i , který pracoval v předchozím kroku, je nyní ve stavu $S_i [i, j]$. Jediný aplikovatelný program je $\langle S_i \rightarrow S_i f, [i, j] \text{ out} \rangle$ pro konkrétní hodnotu indexu $j \in \{1, \dots, r_i\}$ ($p_{i,ja}$ nejsou aplikovatelné, protože v prostředí není S_i , $p_{i,c}$ není aplikovatelný, protože ve stavu agenta není f). Stav agenta se změní na $S_i f$, do prostředí je distribuován symbol $[i, j]$.

Pokud agent B_i v předchozím kroku nepracoval, nebude pracovat ani v tomto kroku, protože jeho stav zůstává $S_i X$ a zároveň (pro jednotlivé typy programů):

- $p_{i,ja}$ nejsou aplikovatelné – v prostředí není žádný výskyt symbolu S_i (jsou tam pouze symboly a' pro $a \in V$, případně symboly f a e),
- $p_{i,jb}$ nejsou aplikovatelné – ve stavu agenta není symbol $[i, j]$ pro žádné $j \in \{1, \dots, r_i\}$, přičemž pravidla $p_{i,jb}$ takový symbol transportují do prostředí,
- $p_{i,c}$ není aplikovatelný – ve stavu agenta není symbol f , který toto pravidlo transportuje do prostředí.

V prostředí jsou nyní pouze „čárkované“ symboly (a případně symboly e a f), tedy provádí přepisy podle pravidel $a' \rightarrow a''$. Po provedení tohoto kroku jsou v prostředí symboly a'' , $a \in V$ a dále symboly typu $[i, j]$ distribuované do prostředí aktivními agenty.

Krok 3. Agent B_i , který pracoval v předchozím kroku, je nyní ve stavu $S_i f$. Jediný aplikovatelný program je $\langle S_i \rightarrow S_i X, f \text{ out} \rangle$ (protože v prostředí není S_i , ve stavu agenta není $[i, j]$). Stav agenta se změnil na $S_i X$, do prostředí je distribuován symbol f .

Pokud agent B_i v předchozím kroku nepracoval, nebude pracovat ani v tomto kroku, protože jeho stav zůstává $S_i X$ a zároveň (pro jednotlivé typy programů):

- $p_{i,ja}$ nejsou aplikovatelné – v prostředí není žádný výskyt symbolu S_i ,
- $p_{i,jb}$ nejsou aplikovatelné – ve stavu agenta není žádný symbol $[i, j]$,
- $p_{i,c}$ není aplikovatelný – ve stavu agenta není symbol f .

Prostředí reaguje takto:

- symboly a'' jsou zpracovány pravidly $a'' \rightarrow \beta$ (simulují činnost prostředí eko-kolonie),
- symboly typu $[i, j]$ jsou zpracovány příslušnými pravidly $[i, j] \rightarrow f_{i,j}$, čímž prostředí spolupracuje na simulaci činnosti agentů A_i z eko-kolonie.

Prostředí je teď ve stavu odpovídajícím stavu prostředí eko-kolonie po provedení jednoho kroku, až na pořadí prvků. Navíc se zde za každého pracujícího agenta objevil symbol f .

Je zřejmé, že průběh výpočtu ve vytvořené eko-P kolonii odpovídá průběhu výpočtu v původní eko-kolonii, s trojnásobným počtem kroků, čemuž je přizpůsoben i vývoj prostředí. Dále se zaměříme na ukončení činnosti eko-P kolonie.

Sledujme nyní výpočet v původní eko-kolonii. Pokud žádný agent A_i nemůže pracovat (v prostředí neexistují symboly, které mají zpracovat), pak v ekvivalentním výpočtu v eko-P kolonii se po daném počtu trojic kroků dostáváme do stejné situace – žádný agent B_i nemůže pracovat, protože v prostředí nejsou žádné symboly S_i (programy $p_{i,ja}$), ve stavech agentů nejsou symboly $[i, j]$

(programy $p_{i,jb}$) a ve stavech agentů není ani symbol f (program $p_{i,c}$). Proto se také činnost eko-P kolonie zastaví.

Výsledkem výpočtu v eko-P kolonii je počet symbolů f v prostředí po provedení posledního kroku výpočtu. V našem případě každý pracující agent transportoval jeden symbol f do prostředí vždy v posledním z trojice kroků simulujících krok činnosti agenta eko-kolonie. V každé trojici kroků do prostředí přibylo tolik symbolů f , kolik agentů v této trojici pracovalo. Po posledním kroku je tedy v prostředí maximálně tolik symbolů f , kolik bylo provedeno trojic kroků vynásobeno počtem agentů eko-P kolonie (v algoritmu číslo n).

5 Závěr

Jak bylo zdůrazněno v Úvodu, cílem tohoto příspěvku není srovnávat výpočetní sílu diskutovaných modelů, zabývali jsme se pouze srovnáním činnosti agentů v jejich prostředí.

Zatímco agenti v eko-kolonii jsou (záměrně) velmi jednoduše konstruováni, nemají žádný vnitřní stav, pouze ovlivňují prostředí (s tím, že nedokážou vygenerovat svůj vlastní startovací symbol) a působení je tedy pouze jednosměrné, v P koloniích a eko-P koloniích jsou agenti poněkud „sofistikovanější“ – mají svůj vnitřní stav a s prostředím se vzájemně ovlivňují působením pravidel několika typů. Eko-kolonie mají kromě toho s eko-P koloniemi společnou další vlastnost – prostředí s vlastním vývojem.

V článku jsou uvedeny dva algoritmy – jeden nastiňující rozdíl ve vztahu agentů a prostředí u eko-kolonií a P kolonií, druhý pro totéž u eko-kolonií a eko-P kolonií.

První uvedený algoritmus produkuje P kolonii, jejíž výpočet nikdy nekončí (protože v každém kroku pracují vždy všichni agenti – po ukončení simulace činnosti agentů původní eko-kolonie agenti provádějí pouze „udržovací“ kroky, nicméně je provádějí a výpočet tudíž nemůže být ukončen. Navíc je na vstupu algoritmu pouze silně zjednodušená eko-kolonie se statickým prostředím a agenty, kteří generují pouze jednoznakové řetězce. Tento algoritmus lze proto brát pouze jako studii srovnání vlivu agentů na prostředí.

Druhý algoritmus je již funkční i ve smyslu ukončení výpočtu. Na vstupu algoritmu může být jakákoliv eko-kolonie, výstupem je eko-P kolonie simulující činnost agentů v prostředí podle eko-kolonie na vstupu. Délka výpočtu je trojnásobná oproti původní eko-kolonii, aktivní (pracující) agenti navíc v prostředí generují symboly f . Prostředí výsledné eko-P kolonie se vyvíjí stejně jako v eko-kolonii, navíc vyrovnává určitý handicap agentů – agenti eko-P kolonie pracují pouze s jednotlivými symboly, nikoliv s řetězci, tedy oni sami do prostředí netransportují řetězce, pouze prostředí „sdělí“ pomocí symbolů typu $[i, j]$, jaký řetězec má vytvořit.

Z tohoto pohledu můžeme říci, že eko-P kolonie jsou (co se týče činnosti agentů v prostředí a samotného pro-

středí) eko-koloniím blíže, a to nejen díky možnosti vývoje prostředí.

Za úvahu by stálo, jak by srovnání dopadlo, pokud bychom v P kolonii přidali prostředí možnost vlastního vývoje a zároveň bychom nechali původní typy pravidel (evoluční, komunikační, kontrolní) a neomezovali kapacitu na hodnotu 2.

Poděkování

Tento příspěvek vznikl ve spojitosti s projektem IT4Innovations Centre of Excellence, reg. no. CZ.1.05/1.1.00/02.0070.

Literatura

Cienciala, L. (2015). *Membránové agenty: O variantách P kolonií*. Slezská univerzita v Opavě.

Cienciala, L. a Ciencialová, L. (2011). P colonies and their extensions. V *Computation, Cooperation, and Life – Essays Dedicated to Gheorghe Paun on the Occasion of His 60th Birthday*, str. 158–169, Berlin Heidelberg. Springer-Verlag.

Kelemen, J. a Kelemenová, A. (1992). A grammar-theoretic treatment of multiagent systems. *Cybern. Syst.*, 23(6):621–633.

Kelemen, J. a Kelemenová, A. (2005). On P colonies, a biochemically inspired model of computation. *Proc. of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*, str. 40–56.

Kelemenová, A. a Vavrečková, S. (2011). Generative power of eco-colonies. V *Computation, Cooperation, and Life – Essays Dedicated to Gheorghe Paun on the Occasion of His 60th Birthday*, str. 107–121, Berlin Heidelberg. Springer-Verlag.

Păun, G. (2000). Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143.

Vavrečková, Š. (2005). Eko-kolonie. *Kognice a umělý život – sborník konference*, str. 601–612.