

# Využití modelů ontogeneze v evolučním návrhu číslicových obvodů

**Michal BIDLO**

Informační technologie, 1. ročník, denní studium

Školitel: Lukáš SEKANINA

Fakulta informačních technologií, Vysoké učení technické v Brně

Božetěchova 2, 612 66 Brno, Česká republika

bidlom@fit.vutbr.cz

**Abstrakt.** Cílem tohoto příspěvku je seznámit čtenáře s možnostmi využití modelů ontogeneze (tzv. development) pro evoluční návrh číslicových obvodů. Jsou diskutovány dvě zcela odlišné techniky, které se doposud ukázaly být vhodnými kandidáty pro návrh různých tříd číslicových obvodů. První z nich je založena na opakované aplikaci posloupnosti instrukcí (tzv. konstruktoru) na existující obvod a umožňuje tvorbu libovolně rozsáhlých systémů několika tříd. Druhá metoda využívá pro modelování ontogeneze celulární automat, jehož vývoj představuje postup konstrukce obvodu o předem známém počtu vstupů.

**Klíčová slova.** Evoluční algoritmus, development, celulární automat, číslicový obvod.

## 1 Úvod

Biologií inspirované algoritmy se v poslední době stále více uplatňují při řešení mnoha komplexních úloh v různých oblastech vědy. Evoluční návrh patří k relativně mladému a rozvíjejícímu se odvětví, jenž nám dosud umožnilo získat mnoho zajímavých výsledků. Jako příklad uved' me kreativní evoluční návrh nebo formy umělého života [5], číslicové obvody [6], neuronové sítě [7], analogové elektronické obvody nebo počítačové programy [8]. Přestože bylo aplikováno několik odlišných metod evolučních algoritmů, bylo možné takto získat pouze relativně jednoduchá řešení.

Problém škálovatelnosti patří pravděpodobně k nejvýznamnějším aspektům, které ztěžují proces evolučního návrhu s využitím současných technologií. Se zvyšující se složitostí cílového systému (např. nárůst počtu vstupních a výstupních veličin nebo počtu komponent potřebných k jeho implementaci) vzrůstá délka chromozomu reprezentujícího kandidátní řešení během evolučního procesu, což má za následek zvětšení vyhledávacího prostoru a je tudíž velmi náročné navrhnout efektivní prohledávací (návrhový) algoritmus.

Dosud bylo vyvinuto několik odlišných přístupů snažících se problém škálovatelnosti překonat. Patří mezi ně *evoluce na úrovni funkčních bloků* (složitějších celků tvořících stavební kameny cílového systému – viz např. [9]), *inkrementální evoluce* (viz např. Toressenova metoda *rozděl a panuj* [10]) a *development* (vývoj cílového systému z triviální instance daného problému podle určitých pravidel – inspirováno biologickými principy ontogeneze, viz např. [11, 12, 13]).

Cílem tohoto příspěvku je seznámit čtenáře s možnostmi využití modelů ontogeneze (tzv. development) pro evoluční návrh číslicových obvodů. Jsou diskutovány dvě zcela odlišné techniky, které se doposud ukázaly být vhodnými kandidáty pro návrh různých tříd číslicových obvodů. První z nich je založena na opakované aplikaci posloupnosti instrukcí (tzv. konstruktoru) na existující obvod a umožňuje

tvorbu libovolně rozsáhlých systémů několika tříd. Druhá metoda využívá pro modelování ontogeneze celulární automat, jehož vývoj představuje postup konstrukce obvodu o předem známém počtu vstupů.

Kapitola 2 obsahuje stručný popis základních principů ontogeneze. V kapitole 3 je uveden popis metody realizující development pro konstrukci libovolně velkých obvodů různých tříd. Zcela odlišná metoda (založená na vývoji celulárního automatu) je uvedena v kapitole 4. Obě tyto kapitoly také obsahují příklad výsledného obvodu získaného aplikací příslušného modelu vývoje. V kapitole 5 je uvedeno zhodnocení dosažených výsledků a přehled možností dalšího výzkumu.

## 2 Development

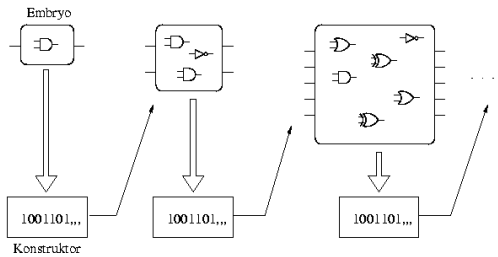
Ontogeneze (development) je biologický proces, během kterého dochází k vývoji zárodečné buňky (zygoty) ve vícebuněčný organismus. Základním principem developmentu je stavba organismu a sebeorganizace. Podstatnou částí ontogeneze je proces konstrukce, který je dán souhrou mezi geny, proteiny, buňkami a prostředím buňky obklopujícím. Výsledkem tohoto procesu je vospělý organismus. V průběhu vývoje zygoty můžeme pozorovat následující fáze [4] (poznamenejme, že tyto dílčí procesy neprobíhají zcela odděleně, ale obvykle se překrývají): (1) *zárodečné dělení* – zárodečná buňka (tzv. zygota) podstoupí v této fázi rapidní dělení, jehož výsledkem je vytvoření shluku buněk (tzv. blastula). Nedochozí zde však k žádnému nárůstu objemu jednotlivých buněk. (2) *organizace buněk* – v této fázi dochází k prostorové a časové organizaci buněk v embryu. V prvním stupni této fáze je vytvořen prostorový koncept organismu. Během druhého stupně jsou vytvořeny základní zárodečné vrstvy (vnější, střední a vnitřní). (3) *morfogeneze* – buněčné migrace, dochází k přemísťování buněk, utváří se základ vnitřností. Tento proces se nazývá gastrulace, z blastuly se vytvoří tzv. gastrula. (4) *odlišení buněk* – jednotlivé buňky získávají vlastní strukturu a funkci, vzniknou různé typy buněk, např. nervové, kožní atd. (5) *růst* – organismus nabývá na velikosti (rozmnožování buněk, růst buněčné hmoty).

V oboru počítačového inženýrství existuje mnoho technik, které jsou inspirovány právě biologickými procesy. Pravděpodobně nejznámější a nejrozšířenější jsou evoluční algoritmy. Principy ontogeneze bývají mnohdy využívány buď samostatně nebo v kombinaci s jinými biologii inspirovanými metodami. Jelikož je development ve skutečnosti velice komplikovaný proces, bývají jeho modely podstatně zjednodušeny, případně simulují pouze některé jeho části. V oblasti počítačového inženýrství je development primárně chápán jako mapovací proces mezi genotypem a fenotypem v evolučním algoritmu. Genotyp v takovém případě obsahuje *předpis* pro konstrukci cílového systému (fenotypu). Mezi nejznámější biologii inspirované modely umožňující simulaci vývoje mnohabuněčných organismů patří celulární automaty [2] a L-systémy [1].

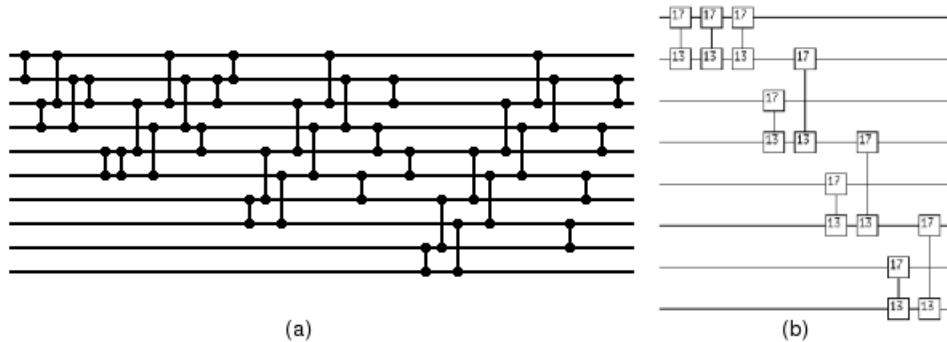
## 3 Development založený na posloupnosti instrukcí

### 3.1 Původ metody a předešlé výsledky

Základní princip této techniky je převzat z [16]. Uvedená publikace se zabývá evolučním návrhem libovolně velkých řadicích a mediánových sítí. Prezentovaná metoda však nepřinesla žádné podstatné vylepšení uvažovaných obvodů, nejlepší výsledky nalezené v [16] jsou shodné s konvenčním řešením. Metoda byla dále rozpracována a zdokonalena v diplomové práci [17], kde bylo také dosaženo výrazně lepších výsledků v oblasti řadicích, případně mediánových sítí. Další rozšíření této metody umožnilo evoluční návrh libovolně velkých obvodů i jiných tříd (např. sčítačky, paritní obvody, polymorfnní obvody [15, 18]).



Obrázek 1: Princip růstu obvodu pomocí konstruktora



Obrázek 2: Ukázky obvodů vytvořených konstruktory: (a) řadicí síť, (b) polymorfní paritní obvod (kód 13 je klasický XOR, kód 17 označuje polymorfní hradlo pracující v jednom režimu jako funkce identity, v režimu druhém jako NOT a zajišťuje tak změnu funkce obvodu liché/sudé parity mezi dvěma prostředími).

### 3.2 Princip metody

Tento vývojový model umožňuje konstrukci různých tříd číslicových obvodů o libovolném počtu vstupů. Evoluční algoritmus je použit k nalezení vhodné posloupnosti instrukcí konstruktora (= program, předpis pro stavbu složitějšího obvodu ze známého jednoduššího obvodu). Opakovaná aplikace konstruktora na obvod vytvořený předchozí aplikací konstruktora (na počátku na embryo – triviální řešení problému) umožňuje teoreticky nekonečný růst obvodu, jenž je po každém takovémto vývojovém kroku plně funkční a obsahuje všechny prvky předchozí instance. Konstruktor s těmito vlastnostmi nazveme *obecným konstruktorem*. Princip této metody znázorňuje obrázek 1. Kritickou částí celého modelu je volba vhodného instrukčního souboru. V případě číslicových obvodů se ukázaly být vhodnými instrukcemi operace typu *copy*, *modify* a *copy-and-modify*, které provádí duplikaci stavebních bloků obvodu, případně mění indexy jejich vstupů nebo funkce samotných hradel v těchto prvcích obsažených.

### 3.3 Dosažené výsledky

S využitím tohoto modelu bylo nalezeno několik konstruktů, které umožňují tvorbu libovolně velkých obvodů různých tříd. Největší úspěch byl zaznamenán v oblasti řadicích sítí, kde bylo s využitím genetického algoritmu nalezeno několik obecných konstruktů, které produkují řadicí sítě s podstatně lepšími vlastnostmi (počet porovnávacích operací a zpoždění obvodu), než které byly sestrojeny konvenční technikou (např. bubble-sort). Dalšími oblastmi jsou sčítačky, mediánové a paritní obvody, které však nepřinášejí žádnou inovaci a jsou shodné s konvenčním řešením. Zajímavé výsledky však poskytuje evoluční návrh v oblasti polymorfních obvodů [14], v níž byly s využitím tohoto vývojového modelu nalezeny řadicí sítě pro vzestupné a sestupné řazení v různých dvou režimech (prostředích) a dále polymorfní paritní obvody [15]. Obrázek 2 ukazuje (a) příklad efektivní řadicí sítě (v porovnání s konvenční metodou konstrukce) a (b) polymorfní obvod pro výpočet sudé a liché parity v různých dvou režimech.

## 4 Development založený na celulárním automatu

### 4.1 Původ metody a předešlé výsledky

Narozdíl od předcházející metody, tato technika je zaměřena na evoluční návrh obvodů různých tříd o předem známém počtu vstupů. Vývojový model založený na celulárním automatu byl od základu navržen v rámci samostatně řešeného projektu do kurzu *Evoluční výpočetní techniky*. Jedná se o původní experimentální model, který zatím nebyl v oblasti evolučního návrhu číslicových obvodů diskutován.

### 4.2 Princip metody

Celulární automaty (CA) jsou dynamické systémy tvořené diskretní soustavou buněk, z nichž každá se může nacházet v jednom stavu z konečné množiny stavů. Stav buněk jsou aktualizovány synchronně v diskretních časových krocích v závislosti na *lokálním přechodovém pravidle* [3]. Toto pravidlo udává následující stav dané buňky pro libovolnou kombinaci stavů buněk v jejím *sousedství* (tzv. přechodová funkce).

V našem případě bude vývojovým modelem jednorozměrný uniformní CA (lineárním uspořádání buněk, všechny buňky obsahují identické přechodové pravidlo), v němž sousedství každé buňky obsahuje jednoho bezprostředního levého a pravého souseda a buňku samotnou. Levým, resp. pravým sousedem nejlevější, resp. nejpravější buňky CA bude pomyslná buňka nacházející se ve stavu 0 (tzv. *nulové okrajové podmínky*). Pro potřeby návrhu číslicových obvodů rozšíříme tento CA o generativní schopnost buněk po každém vývojovém kroku. Každá buňka vygeneruje po přechodu do dalšího stavu právě jeden *symbol* tvaru  $[a, b, f]$  interpretovaný jako logické hradlo, kde  $a, b$  označuje indexy vstupů a  $f$  jeho funkci. Konkrétní podoba generovaného symbolu je dána kombinací stavů buněk v sousedství uplatněné pro určení nového stavu buňky. V každém kroku CA je tedy vygenerován jeden stupeň obvodu. Vstupy hradel dalšího stupně tvoří výstupy hradel stupně předchozího, zpětná vazba je nepřijatelná.

Genetický algoritmus byl v tomto případě použit k nalezení počátečního stavu celulárního automatu a přechodového pravidla buněk. Počet buněk CA byl v každém experimentu shodný s počtem vstupů tvořeného obvodu. Počet vývojových kroků CA potřebný ke konstrukci daného obvodu a počet stavů byly zjištěny experimentálně.

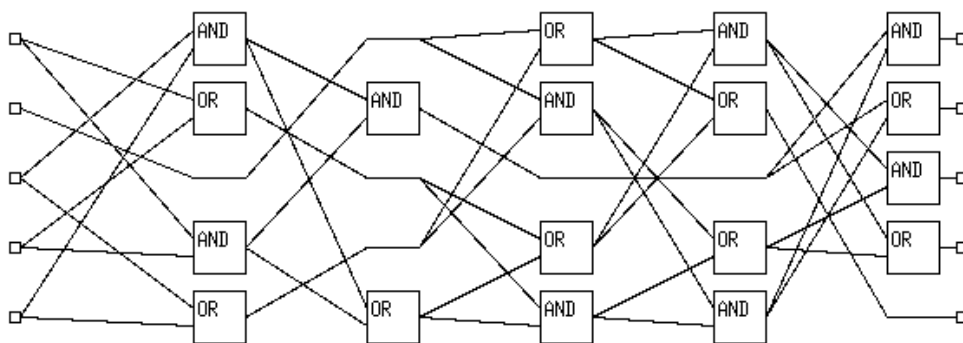
### 4.3 Dosažené výsledky

Narozdíl od předcházející metody, vývojový model založený na celulárním automatu slouží pouze k návrhu obvodů o předem stanoveném počtu vstupů. Přesto lze s využitím tohoto přístupu navrhovat širokou škálu kombinačních obvodů. Úspěšnost návrhu konkrétní struktury závisí především na typu cílového obvodu. Pro obvody s vysokým stupněm různorodosti logické struktury, případně pro obvody s vyšším počtem významných výstupních bitů, byl návrh náročnější. K obvodům různých tříd, které byly takto vytvořeny patří sčítačky, násobičky, mediánové a řadicí sítě, paritní obvody a obvody booleovské symetrie. Obrázek 3 ukazuje příklad řadicí sítě, kterou našel genetický algoritmus s využitím uvedeného vývojového modelu.

## 5 Zhodnocení a závěr

V příspěvku byly představeny dvě zcela odlišné metody pro realizaci tzv. developmentu pro evoluční návrh číslicových obvodů. Obě tyto techniky ukázaly schopnost návrhu několika různých tříd obvodů (např. sčítačky, řadicí a mediánové sítě, paritní obvody).

V případě vývojového modelu založeného na posloupnosti instrukcí bylo nalezeno několik obecných konstruktorů, které jsou schopny vytvořit řadicí/mediánové sítě o libovolném počtu vstupů, jejichž parametry (počet porovnávacích operací a zpoždění obvodu) jsou podstatně lepší než u obvodů získaných



Obrázek 3: Pětivstupová řadič síť vytvořená celulárním automatem s generativními schopnostmi.

konvenční metodou. Bylo tedy nalezeno inovativní řešení nejen pro jednu instanci problému, ale pro všechny instance uvedené třídy obvodů. Ačkoliv vytvořené obvody nedosahují takových kvalit jako nejlepší známá řešení, objevený princip může být uplatněn na některé nejlepší známé řešení o  $n$  vstupech k vytvoření  $(n + 2)$ -vstupové řadič sítě, která bude mít v takovém případě lepší parametry než při použití konvenčního přístupu.

Vývojový model založený na celulárním automatu umožňuje (z důvodu zcela odlišného přístupu ke konstrukci obvodů) návrh kombinačních obvodů pouze s předem stanoveným počtem vstupů. Velikost obvodů, které se s využitím tohoto modelu podařilo navrhnout, závisí ve velké míře na různorodosti jeho logické struktury a na počtu významných bitů výsledku. Například, paritní obvody byly úspěšně navrženy až do počtu 14 vstupů (zde se již velmi výrazně začíná projevovat časová náročnost testování obvodu, tedy problém škálovatelnosti výpočtu fitness). Naproti tomu, sčítačky nebo násobičky byly nalezeny pouze pro maximální počet 5 vstupů. Toto omezení může být zapříčiněno použitím uniformního celulárního automatu, který je při použití konečné konfigurace výpočetně slabší než automat neuniformní. Zvýšení počtu stavů přináší do jisté míry usnadnění návrhu, má však za následek enormní nárůst velikosti vyhledávacího prostoru, což vede v současné době ke znemožnění nalezení platného řešení (problém škálovatelnosti).

Jak je patrné z předcházejících odstavců, jsou obě metody použitelné jako modely vývoje v evolučním návrhu číslicových obvodů, jenž je v kombinaci s nimi schopen nalézt kvalitní řešení. Biologický development je však v porovnání s uvedenými modely velice komplikovaný proces, v němž dochází také k interakci s prostředím obklopujícím vyvíjející se organismus, které nebylo pro jednoduchost v těchto případech uvažováno. Předmětem dalšího výzkumu proto bude studium vlivu okolních signálů (jejich reprezentace, interpretace apod.) na vývojový proces, který však už nebude zaměřen pouze na návrh obvodů, ale bude přizpůsoben potřebám nového vývojového modelu v souvislosti se studiem konkrétních jeho vlastností (např. vliv zakódování znalostí do prostředí oproti jejich umístění v chromozomu evolučního algoritmu). V oblasti celulárních automatů bude naše pozornost zaměřena na studium emergentního chování (tj. vytvoření určité struktury, konfigurace CA, po určitém počtu vývojových kroků z teoreticky libovolné počáteční konfigurace).

## Poděkování

Výzkum byl proveden s podporou Grantové agentury České republiky v rámci projektu č. 102/04/0737 Moderní metody syntézy číslicových systémů a č. 102/05/H050 Integrovaný přístup k výchově studentů DSP v oblasti paralelních a distribuovaných systémů.

## Reference

- [1] Lindenmayer, A.: Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, 18:280–315, 1968
- [2] von Neumann, J.: *The Theory of Self-Reproducing Automata* (edited by A. W. Burks). University of Illinois Press, 1966
- [3] Sipper, M.: *Evolution of Parallel Cellular Machines*. Springer–Verlag Berlin Heidelberg, LNCS vol. 1194, p. 203, 1997
- [4] Kumar, S., Bentley, P. J. (eds.): *On Growth, Form and Computers*. Elsevier Academic Press, 2003
- [5] Bentley, P. J. (ed.): *Evolutionary Design by Computers*. Morgan Kaufmann Publisher, San Francisco, 1999
- [6] Miller, J. F., Job, D.: Principles in the evolutionary design of digital circuits – part I. *Genetic Programming and Evolvable Machines*, 1(1), pp 8–35, 2000
- [7] Yao, X.: Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9), pp 1423–1447, 1999
- [8] Koza, J. R. et al.: *Genetic Programming III: Darwinian Invention and Problem Solving*. Morgan Kaufmann Publisher, San Francisco, 1999
- [9] Murakawa, M. et al.: Evolvable Hardware at Function Level. In: *Proc. of the Parallel Problem Solving from Nature IV*, LNCS vol. 1141, pp 62–71, Springer–Verlag Berlin Heidelberg New York, 1996
- [10] Toressen, J.: A scalable approach to evolvable hardware. *Genetic Programming and Evolvable Machines*, 3(3), pp 259–282, 2002
- [11] Gordon, T. G. W., Bentley, P. J.: Towards development in evolvable hardware. In: *Proc. of the 2002 NASA/DoD Conference on Evolvable Hardware*, pp 241–250, IEEE Computer Society Press, 2002
- [12] Haddow, P., Tufté, G.: Bridging the genotype–phenotype mapping for digital FPGAs. In: *Proc. of the 3rd NASA/DoD Workshop on Evolvable Hardware*, pp 109–115, IEEE Computer Society Press, 2001
- [13] Sekanina, L., Bidlo, M.: Evolutionary Design of Arbitrarily Large Sorting Networks Using Development. *Genetic Programming and Evolvable Machines*, vol. 6, 2005 (in print)
- [14] Stoica, A., Zebulum, R., Keymeulen, D.: Polymorphic Electronics. In *Proc. of International Conference on Evolvable Systems: From Biology to Hardware*, LNCS 2210, Springer Verlag, 2001, pp. 291–302
- [15] Bidlo, M., Sekanina, L.: Providing Information from the Environment for Growing Electronic Circuits Through Polymorphic Gates. In: *Proc. of the Genetic and Evolutionary Computation Conference – Workshops 2005*, New York, US, ACM, 2005, pp. 242–248
- [16] Sekanina, L.: Evolving Constructors for Infinitely Growing Sorting Networks and Medians. In: *Proc. of the Conference on Current Trends in Theory and Practice of Computer Science SOFSEM 2004*, LNCS 2932, Springer–Verlag, 2004, pp. 314–323
- [17] Bidlo, M.: *Evoluční návrh řídicího algoritmu [diplomová práce]*. Fakulta informačních technologií, Brno, CZ, 2004
- [18] Bidlo, M.: *Evolutionary Design of Arbitrarily Large Sorting Networks and Adders*. Brno, CZ, 2005