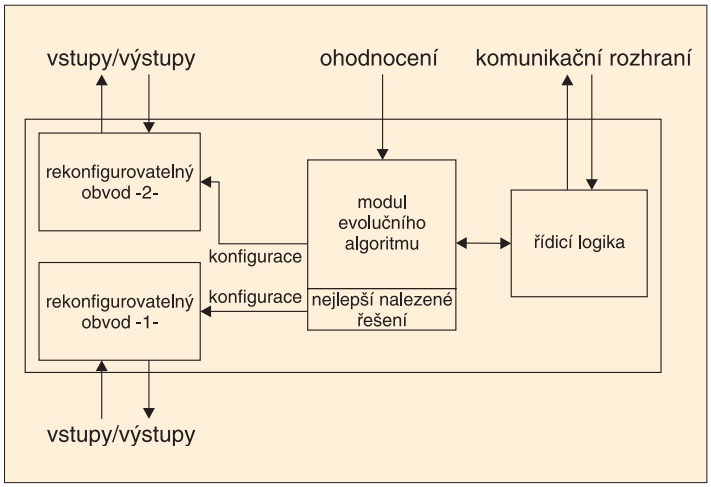
Návrh číslicových obvodů za pomoci evolučních výpočetních technik

Evoluční výpočetní techniky (EVT) jsou skupinou algoritmů, které lze velmi efektivně využít pro řešení optimalizačních problémů či úloh. Tyto techniky patří do kategorie algoritmů umělé inteligence. Jejich inspirace biologií však nevede k tomu, aby napodobovaly lidské myšlení. Evoluční výpočetní techniky napodobují principy postupné evoluce živočišných druhů. Snaží se tedy postupným vylepšováním hledat vhodná (nikoliv vždy ideální) řešení optimalizačního problému. Skupina EVT je široká, mezi základní zástupce této skupiny patří například genetický algoritmus, evoluční strategie, genetické programování atd.

Evoluční výpočetní techniky

Všechny evoluční algoritmy jsou si principiálně velmi blízké. Algoritmy této skupiny jsou založeny na existenci tzv. jedinců (někdy nazývaných chromozomy), které následně tvoří populace. Každý jedinec populace představuje právě jedno řešení daného optimalizačního problému. Populace je pak tedy skupina různých řešení optimalizace. Úkolem algoritmu je postupným vývojem kvalitu těchto řešení zlepšovat až na úroveň, kterou požadujeme. Jednou z klíčových otázek je, jak je řešení optimalizační úlohy jedincem reprezentováno. Velmi často se využívá prostá reprezentace v binárním tvaru, ale je možné využít i celočíselnou či jinou formu. Na počátku běhu algoritmu je nutné jedince inicializovat. To se většinou děje tak, že se naplní náhodnými hodnotami, popřípadě se jedinci vytvoří na základě znalostí o řešené úloze. Následuje aplikování tzv. rekombinačních operátorů na populaci. Ty jsou nejčastěji dva: křížení a mutace. Křížení představuje sloučení genetických informací ze dvou (či více) jedinců (rodičů), čímž dojde k vytvoření nových jedinců (potomků). Přesný princip křížení je závislý na použitém algoritmu a reprezentaci chromozomu. Druhý operátor – mutace – představuje menší zásah do genetické informace jedince, nejčastěji se tímto pojmem rozumí například negace jednoho či několika bitů v reprezentaci jedince. Tím, jak se modifikuje obsah jedinců, se mění i řešení optimalizační úlohy. Proto po použití rekombinačních operátorů je nutné zjistit, jak kvalitní řešení daný jedinec přináší. K tomuto účelu slouží funkce ohodnocení (*fitness*). To je jeden z nejklíčovějších elementů evolučních algoritmů. Optimalizační úlohu je třeba vhodně přetransformovat tak, aby jí bylo možné vnímat jako problematiku hledání globálního (lokálního) maxima (minima). Potom můžeme u každého jedince jednou hodnotou vyjádřit, jak kvalitní řešení úlohy přináší. Jestliže po ohodnocení jedinců populace, jejich mutantů či potomků zjistíme, že bylo nalezeno uspokojivé řešení, můžeme algoritmus ukončit a výsledkem optimalizace bude jedinec s nejlepším ohodnocením. Pokud tato podmínka nebude splněna, z jedinců vybereme další populaci, která znovu podstoupí aplikování rekombinačních operátorů. A to až do doby, než budeme s kvalitou řešení spokojeni. Algoritmus můžeme též ukončit například po vykonání určitého počtu generací. Vybírání jedinců pro novou generaci (tzv. selekce) může probíhat několika způsoby. Nejznámější typy selekce fungují na principu rulety či turnaje. Podstatné však je, že jedinec přinášející kvalitní řešení, má větší šanci postoupit do další generace než jedinec s menším ohodnocením. Tento mechanismus odpovídá inspiraci z biologie; v přírodě také přežívá nejsilnější (či nejpřizpůsobivější) jedinec. Selekce je často doplněna tzv. elitismem, tedy operací, která vždy zajistí přežití nejkvalitnějšího jedince [1 a 2].



Blokové schéma vyvíjejícího se obvodu se dvěma rekonfigurovatelnými obvody

Návrh obvodů jako optimalizační úloha

Jak již bylo zmíněno, evoluční výpočetní techniky mohou řešit optimalizační problémy, které je možné transformovat na úlohy hledající globální (lokální) minimum (maximum). To platí i pro návrh obvodů, a to jak číslicových, tak analogových. Důvodů, proč využívat právě evolučních technik, může být několik. Tyto techniky mohou najít nová obvodová řešení, která jsou zcela odlišná od řešení poskytovaných konvenčními návrhovými metodami. Tato „nová řešení“ mohou být v mnoha případech výhodnější, například z hlediska množství použitých hardwarových prostředků. Často mohou tyto techniky návrhu překonat i předsudky návrhových pracovníků. Dalším důvodem pro používání tohoto způsobu návrhu může být potřeba získat obvodové řešení třeba i nižší kvality, avšak v relativně krátkém čase, za který by například konvenční techniky návrhu nenalezly řešení žádné. Neopomenutelný je také fakt, že evoluční algoritmy umožňují návrh obvodů z více hledisek než jen z hlediska funkce. Můžeme například optimalizovat obvod i z pohledu zpoždění a počtu použitých elementů. Spojení evolučních technik a návrhu číslicových obvodů můžeme rozdělit do dvou základních kategorií [4]:

- evoluční návrh obvodů;
- vyvíjející se obvody.

Evoluční návrh obvodů

Evoluční návrh obvodů přichází v úvahu za situace, kdy je řeba navrhnout obvod, který může být následně reálně implementován, ale nevyžaduje se dynamická změna jeho chování a funkce. Základem tohoto návrhu je kombinace příslušného evolučního algoritmu a rekonfigurovatelného obvodu. Musí být tedy k dispozici vhodná obvodová struktura (postačí jen její simulace), která je schopná měnit na základě konfigurační informace svou funkci. Optimalizační úloha pak určuje na základě požadované funkce vhodnou konfiguraci této rekonfigurovatelné struktury. Aby mělo použití evolučního návrhu smysl, výsledné obvodové řešení musí mít přínos oproti standardním návrhovým technikám, např. již zmíněnou úsporu hardwarových prostředků nebo lepší chování z hlediska časování.

Při evolučním návrhu představuje jedinec populace právě jednu konfigurační informaci pro rekonfigurovatelný obvod. Změnou jedince měníme i konfiguraci této struktury, a tak měníme její funkci. Podle použité struktury můžeme pak volit různé rekombinační operátory. Co se týče ohodnocení jedinců, zde se skrývá velké úskalí tohoto návrhu. A to především proto, že otestování

funkce rekonfigurovatelného obvodu pro danou konfigurační informaci může být velmi složité. Nejsnadnější je situace v případě malých kombinačních obvodů. Ty je možné definovat pravdivostní tabulkou. Funkce ohodnocení pak zjišťuje, v kolika řádcích pravdivostní tabulky se shoduje výstup zkoumaného obvodu s požadovanou hodnotou. Mějme například kombinační obvod se čtyřmi vstupy a dvěma výstupy. Obvod se testuje na všechny možné vstupy, tzn. 16 hodnot („0000“ až „1111“). Při každé kombinaci vstupních hodnot může nastat shoda v jednom, ve dvou či v žádném výstupu. Odtud plyne maximální hodnota ohodnocení, která je 32 (16 × 2). Tato hodnota značí, že nalezený kombinační obvod se chová přesně podle požadovaných vlastností. Samozřejmě takové ohodnocení udává kvalitu nalezeného řešení jen z pohledu funkce. Existuje-li požadavek na optimalizaci i dalších parametrů, je třeba složitější funkce ohodnocení. Je zřejmé, že v případě velkých kombinačních obvodů, či dokonce v případě sekvenčních obvodů, může být ohodnocení jedinců velmi zdlouhavé. Nemluvě o problematice návrhu analogových obvodů, kde je třeba počítat s fyzikálními modely součástek atd. Je nutné připomenout, že v případě evolučního návrhu obvodu se kvalita obvodu testuje nejčastěji pouze softwarově (simulátorem). Časová náročnost hledání obvodového řešení však v případě evolučního návrhu obvodu nemusí být zásadní. Je však nutné nalézt obvodové řešení, které je nějakým způsobem inovativní, jinak by návrh touto metodou pozbýval smysl. Evoluční návrh lze použít nejen pro hledání obvodového řešení, ale můžeme takto optimalizovat například hodnoty součástek analogového filtru [3 a 4].

Samostatně se vyvíjející obvody

Vyvíjející se obvody používají stejných principů jako evoluční návrh. Avšak tento pojem by měl být používán jen za přesně definovaných podmínek, tzn. pouze pro obvody, které jsou schopné za pomoci evolučních algoritmů měnit dynamicky v čase svou funkci, chování či parametry. Základem je opět obvod schopný dynamické rekonfigurace či změny parametrů, který musí být ale v tomto případě reálně implementovaný. Na základně evoluční optimalizace je pak měněna jeho konfigurační informace a tím i funkce, která se ale díky přímé hardwarové implementaci projevuje ihned ve fungování obvodu. Aby se mohl obvod samostatně vyvíjet, musí být přítomná jednotka, která zajistí realizaci příslušné evoluční optimalizační techniky. Ta by měla být přímo součástí vyvíjejícího se obvodu. Jedině tak můžeme zajistit, že je obvod autonomně schopný svého vývoje. To znamená, že takový obvod by měl být založen na dvou základních částech, na rekonfigurovatelném obvodu a na modulu zajišťujícím jeho evoluci. S dnešními obvody FPGA není problém obě části implementovat v rámci jedné součástky.

Jednotka realizující evoluční algoritmus samozřejmě nemusí být nutně přímo součástí obvodu schopného samovyvíjení. Může být realizována externě, například připojeným procesorem. Pro účely výzkumu a lepšího ladění je možné algoritmus realizovat i na PC a v reálném hardwaru ověřovat jen ohodnocení jedinců. Avšak je zřejmé, že pokud je algoritmus implementován přímo do logiky obvodu FPGA, budou časy potřebné pro evoluci kratší.

Stejně tak existuje několik způsobů, jak vytvořit rekonfigurovatelný obvod (strukturu). Velký počet experimentů probíhal tím způsobem, že byla pro tento účel vyhrazena část obvodu FPGA a optimalizací byl hledán vhodný konfigurační bitstream daného obvodu. Aby bylo možné takové metody použít, musí být splněno několik podmínek. Zaprvé je nutné, aby bylo možné neomezeně a v relativně krátkém čase měnit konfigurační bitstream FPGA. Dále je nutné znát konfigurační protokol daného obvodu. A také je důležité zaručit, že žádná náhodná konfigurace nemůže obvod trvale poškodit. Za zmínku jistě stojí i fakt, že podobný postup by byl použitelný i pro samovyvíjející se analogovové obvody, kde by se jako rekonfigurovatelná struktura dal využít obvod typu FPAA. Tento přístup ovšem přináší několik úskalí, jako je například relativně velká délka konfiguračního bitstreamu části obvodu FPGA. To by se odrazilo i ve značné délce jedince v optimalizačním algoritmu. A zde narážíme na velký problém, na tzv. škálovatelnost. Ta se projevuje tak, že při nárůstu délky jedince se razantně prodlužuje potřebná doba pro úspěšnou evoluci obvodu. Další problémem je otázka přenositelnosti – tím, že necháme konfiguraci části obvodu FPGA plně v rukou evolučního algoritmu, nemáme žádnou kontrolu nad tím, která obvodová zapojení budou realizována. Může se stát, že vznikne velmi exotické zapojení, ve kterém se budou využívat hazardy, specifická zpoždění obvodu, nepředpokládané zpětné vazby atd. To může vést k tomu, že nalezená konfigurace nemusí být použitelná pro jiný obvod FPGA (i když se bude jednat o typově stejný obvod) [4].

Z výše uvedených důvodů je proto výhodnější zavádět nové struktury, pomocí nichž se těmto problémům vyhneme. Jde o různé struktury, které jsou složeny z funkčních buněk, propojených skupinami multiplexerů. Funkční buňky jsou schopné měnit na základě konfigurace svou funkci, multiplexery pak zajišťují řízení toku signálů mezi buňkami, vstupy a výstupy. Tyto struktury se označují jako „virtuální rekonfigurovatelné zařízení“ (*Virtual Reconfigurable Device*) [4]. Jsou navrženy v některém z jazyků pro popis číslicových obvodů (VHDL, Verilog), a pak implementovány do FPGA. To znamená, že problém s přenositelností je u nich z funkčního hlediska eliminován. Pokud jsou tyto struktury dobře navrženy, je možné snadno a velmi rychle měnit jejich konfiguraci. Co se týče problémů s délkou konfigurační informace, závisí na úrovni funkčních buněk. Ty mohou být například na úrovni hradel, tzn. že mohou plnit funkce AND, OR, NOR, XOR, NOT. Lze však využívat i funkce vyšší úrovně, jako jsou například sčítačky, násobičky, výběr minima (maxima) atd. Tak je možné zkrátit délku konfiguračního řetězce, a tím i délku jedince v evolučním algoritmu. Nutno však připomenout, že čím nižší úrovně funkčních bloků zvolíme, tím pravděpodobněji najde evoluce zajímavé a inovativní řešení obvodu [4].

U vyvíjejících se obvodů je klíčová volba hodnotícího kritéria. Volí se nejčastěji jen ohodnocení z hlediska funkce obvodu. Příliš nás nezajímá, jak bude výsledný obvod konkrétně vypadat. Záleží především na uspokojivé funkci obvodu a také na čase, během kterého je evoluce schopna tento obvod vytvořit. Funkce ohodnocení může být statická či dynamická v závislosti na konkrétní úloze. Případ statické funkce nastává například v situaci, kdy je třeba čas od času vytvořit kombinační logický obvod. Ten může být definován pravdivostní tabulkou a ohodnocení jedince je pak řešeno tak, jak bylo uvedeno výše. Pak je tedy zřejmé, co je cíl evoluce, k jaké funkci obvodu má optimalizace konvergovat. Po zadání očekávaných výstupů obvodu je spuštěna evoluce. Když je nalezen přijatelný výsledek, evoluce může být zastavena. Nejlepší jedinec pak představuje konfiguraci pro výsledný obvod. Evoluce se dále spouští jen v okamžiku, kdy je třeba nalézt obvod realizující jinou funkci.

Odlišná situace nastává nastává v případě adaptivního filtru. Při této úloze nemusíme hledat konfiguraci rekonfigurovatelného obvodu, ale například koeficienty číslicového filtru. Před zahájením evoluce ale předem nevíme, jak má funkce ohodnocení vypadat. Adaptivní filtr musí reagovat na svůj aktuální vstup a musí se tomuto vstupu přizpůsobovat. Hovoříme tedy o tom, že aplikace pracuje v dynamickém prostředí, a proto i funkce ohodnocení se mění v čase. Cíl optimalizace je tedy dynamicky proměnný. Evoluční algoritmus pak běží *de facto* kontinuálně. Filtr se musí neustále přizpůsobovat proměnnému prostředí (vstupu). V aplikacích, kde je žádaný určitý typ adaptace, se vyskytují obvykle minimálně dva stejné rekonfigurovatelné obvody (například dva filtry). Jeden je využíván pro evoluční algoritmus, konkrétně pro funkci ohodnocení. Druhý obvod zpracovává konkrétní vstupní signál [4]. Tento obvod je konfigurován podle nejlepšího jedince, který byl zatím evolucí nalezen (*obrázek*). Pro zkrácení doby

ohodnocení jedinců populace je vhodné používat větší počet rekonfigurovatelných obvodů i v jiných případech. Vhodně navržená paralelizace může evoluci velmi urychlit.

Zajímavě se mohou zachovat tyto obvody v případě poruchy. Dojde-li například k poruše v rekonfigurovatelném obvodu, nemusí to vždy mít fatální důsledky. Během testování bylo možné pozorovat, že evoluční algoritmus je schopen se s jistými poruchami vypořádat. Zvolí například jiné zapojení, v němž chybovou část obvodu nepoužije, či dokonce této poruchy může využít jako funkčního prvku (např. spojené vstupy hradla NAND jako NOT).

Závěr

Využití evolučních výpočetních technik může být velmi užitečný nástroj pro návrh obvodů. Uplatní se například při hledání nových inovativních řešení, která zatím nejsou využívána. Stejně tak je možné využívat tyto metody pro návrh aplikací s adaptivním charakterem, pro obvody schopné samostatného vývoje v závislosti na požadované funkci. Širší využívání těchto technik zatím čelí několika překážkám. Je nutné hledat vhodné rekonfigurovatelné struktury, vhodně modifikovat evoluční algoritmy. V neposlední řadě lze také hledat způsoby vhodné implementace do obvodů programovatelné logiky. Návrh vyvíjejících se obvodů je velká výzva pro vývojové a vědecké pracovníky.

Ing. Petr Burian
KAE, ZČU v Plzni

LITERATURA

[1] LAŽANSKÝ, J., MAŘÍK, V., ŠTĚPÁNKOVÁ, O., *Umělá inteligence 3*. Praha : Academia, 2003. ISBN 80-200-0472-6

[2] LAŽANSKÝ, J., MAŘÍK, V., ŠTĚPÁNKOVÁ, O., *Umělá inteligence 4*. Praha : Academia, 2003. ISBN 80-200-1044-0

[3] SEKANINA, L., DRÁBEK, V., Soft Hardware. *Vesmír*, 81 (7), 2002, s. 393–395.

[4] SEKANINA, L.: *Evolvable components.* Springer, Natural Computing series, 2004. ISBN 3540403779

[5] GORDON, T. G. W., BENTLEY, P. J., *On Evolvable Hardware*. In Ovaska, S. and Sztandera, L. (Ed.) Soft Computing in Industrial Electronics. Germany : Physica-Verlag, Heidelberg, 2002, pp. 279–323.

Aktuálně k analýze konkurence s ohledem na krizi

Jednotlivé fáze národohospodářského vývoje přinášejí firmám nová úskalí tím, že se musí, mnohé zcela nově zabývat problematikou stagnujících či smršťujících se, případně recesivních trhů. Zatímco důvodem pro stagnaci, případně smrštění tržních objemů je nasycení trhu, existence nákladově (a tudíž cenově) příznivějších substitučních produktů, případně demografické a společenské změny, recesivní trhy jsou jednoznačně charakterizovány poklesem tuzemské i zahraniční poptávky. To znamená, že domácí poptávka se vyznačuje sníženou osobní spotřebou, přičemž je dále ovlivňována stagnující společenskou spotřebou, poklesem investic a poklesem počtu výdělečně činných osob. Pokud jde o zahraniční poptávku je důsledkem snížení konjunktury na straně obchodních partnerů, zhodnocování vlastní měny, případně rostoucími mzdovými náklady a politikou protekcionismu zahraničních partnerů. Důsledkem změn na stagnujících až recesivních trzích jsou znatelné změny v chování tržních subjektů. Je to větší vnímání ceny, vyšší opatrnost při nákupním rozhodování, častější odstoupení od nákupu, menší ochota obchodu kooperovat s výrobci, nevyužití výrobních i obchodních kapacit, pokles zisku.

Nové tržní situace a změny v chování tržních subjektů

Systém podnikových cílů by se neměl v těchto situacích zaměřovat na intenzifikaci dosavadních záměrů a směrů, ale na hledání nových cest na základě vlastních tvůrčích schopností. Přitom je třeba rozlišovat různé postižení firem podle růstu cenové konkurence, vlastního stavu zakázek a všeobecného růstu konkurenčního tlaku. Různou intenzitu postižení znázorňuje *obr. 1*.

I z tohoto zjednodušeného výčtu různých možností působení ekonomické krize na tržní subjekty je zřejmé, že jednotlivé strategie nepředstavují absolutní změny, ale především změnu v jejich konkrétní orientaci.

Reálná opatření je možné např. shrnout takto:

- snížení nákladů;
- zlepšení zákaznických služeb;
- zlepšení kvality;
- zesílení inovační činnosti;
- aktivity k zapojení zákazníků;
- zeštíhlení organizace, štíhlá výroba;
- posílení operativního marketingu;
- očištění sortimentu;
- posílení přímé komunikace;
- budování nových odbytových cest;
- vytvoření pobídkových systémů;
- outsourcing;
- vytváření aliancí s konkurencí.

Prof. Meffert doporučuje, aby se rozhodnutí pro konkrétní strategie soustředilo především na relevantní situační proměnné, a to na skupinu:

- tržních podmínek;
- silných stránek konkurence.

Na základě toho pak uvádí normové strategie podle *obr. 2*.

Z uvedené myšlenky volby strategií lze podtrhnout skutečnost, že analýza konkurence představuje jeden ze základních kamenů komplexně pojatého strategického, resp. marketingového výzkumu a všech souvisejících situačních analýz.

Analýza konkurence – význam a podstata

Úspěšná analýza konkurence jako taková, vyžaduje rovněž velmi komplexní přístup. Nemůže se omezit jen na dílčí aspekty, ale musí představovat komplexní informaci, která vznikne tak, že postupně:

- identifikujeme konkurenci;
- určíme strukturu konkurence;
- provedeme vlastní analýzu konkurence;
- určíme konkurenční strategii.

Zásadním pro určení struktury konkurence je identifikace v rámci základních úrovní konkurence. Východiskem zde může být teorie amerického ekonoma Michaela Portera, který vysvětluje konkurenční chování prostřednictvím tržních okolností. Strukturální výnosnost sektoru je výsledkem hry síly mezi pěti konkurenčními silami, mezi které počítá:

- konkurenci uvnitř oboru (odvětví);
- hrozbu vstupu nových konkurentů;
- hrozbu substitučních výrobků;
- vyjednávací sílu kupujících;
- vyjednávací sílu dodavatelů.

Za bližší zmínku stojí hrozba vstupu nových konkurentů. Jelikož obecně tuto konkurenci v podstatě neznáme a uvažujeme, že konkurent se může objevit kdekoliv a kdykoliv, analyzujeme hrozbu této konkurence nepřímo, tj. jako možnost vyplývající z případné nedokonalosti vstupních bariér, které by této konkurenční síle umožnily, resp. neumožnily vstup na trh. Tato konkurence totiž vzniká tehdy, kdy na rostoucím trhu