

# PROČ GENETICKÉ ALGORITMY SELHÁVAJÍ? (WHY DO GENETIC ALGORITHMS FAIL?)

Josef HYNEK

Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu Univerzity Hradec Králové,  
Nejedlého 573, 500 03 Hradec Králové, tel. +420 49 5061 233, E-mail: Josef.Hynek@uhk.cz

## SUMMARY

*Genetic algorithms are quite often presented as a panacea that is suitable to almost any problem and is going to replace traditional techniques. They are oversimplified, oversold and their real potential is very often overestimated. In order to avoid some mistakes related to genetic algorithms application and to prevent frustration associated with that it is necessary as in the other fields of human activities to learn from faults that were done before. To be able to do it these failures must be recorded, described, and analysed.*

*Unfortunately, there are just a few contributions whose authors have confessed a failure when attempting to apply genetic algorithms to a particular problem and who has analysed such a disappointment. When one gets deeper into genetic algorithms field it is easy to realise that these algorithms are definitely not a kind of tool that is applicable immediately and to whichever problem without sufficient understanding the essence of the particular problem. Having understood the nature of the particular problem we might be able to design a hybrid evolutionary algorithm that might be a very competitive one or we might reach a conclusion that it is not a good idea to employ an evolutionary inspired approach. We do believe that there are some very good reasons that can help us to indicate potential problems and that is why we are going to describe them here in a greater detail.*

**Keywords:** genetic algorithm, problems, reasons for failure

## 1. ÚVOD

Přestože genetické algoritmy jsou snadno aplikovatelné a do značné míry i univerzální, nelze automaticky očekávat, že bez detailní znalosti jejich fungování a podrobné analýzy problému, je možné aplikací evolučních technik dosáhnout uspokojivých výsledků. Klíč k opravdu efektivnímu využívání genetických algoritmů je zcela zřetelně ukryt ve využívání maxima znalostí o struktuře a charakteru konkrétního problému (viz. [12]), ve vzájemném prolínání evolučních algoritmů s tradičními technikami, které byly pro řešení těchto problémů využívány v minulosti, v procesu, který souhrnně nazýváme hybridizací.

Ve snaze předejít některým chybám spjatým s aplikací genetických algoritmů a vyhnout se možnému zklamání, je vhodné se stejně jako v jiných oborech lidské činnosti poučit z chyb, které jsou obvykle zaznamenávány právě za tímto účelem. Bohužel, existuje jen málo prací, které by přiznaly neúspěch nasazení genetického algoritmu na určité doméně nebo zkoumaly možné příčiny takových neúspěchů. Protože genetické algoritmy rozhodně nejsou nástrojem použitelným okamžitě a na kterýkoliv problém bez hlubšího porozumění jeho podstatě a následném vhodném přizpůsobení genetického algoritmu konkrétnímu problému, je nutné a nanejvýš potřebné se těmto tématům hlouběji věnovat.

Existuje celá řada důvodů pro možný neúspěch při aplikaci genetického algoritmu a je třeba si na tuto otázku odpovědět v několika různých rovinách.

Dle našeho názoru je nutné rozlišovat neúspěch způsobený neopodstatněnou snahou o využití genetického algoritmu, selhání způsobené nevhodným typem konkrétního problému a konečně pochybení při vlastní aplikaci genetického algoritmu. V dalším textu se pokusíme tento názor obhájit, blíže objasnit a zkonkretizovat.

## 2. NEOPODSTATNĚNÉ POUŽITÍ

Zřejmě nejjednodušším důvodem, proč genetické algoritmy selhávají, je nesprávná motivace pro jejich použití. Protože celá oblast evolučními procesy inspirovaných paradigmat je relativně mladá, velmi populární a zdá se, že mnoho výzkumných pracovníků i praktických specialistů příslušné algoritmy s úspěchem používá, vzniká u některých uživatelů potřeba „jít s dobou“ a také najít pro tuto technologii nějaké použití. Tento entusiasmus, který provází každý slibně se rozvíjející obor, a to nejen v oblasti informatiky, sice přitahuje nové a nové uživatele, ale je též příčinou častých rozčarování, pokud se ideálně vypadající výsledky dosažené na ilustrativních úlohách školního typu nedaří zopakovat při řešení skutečných problémů [7].

Podobně jako u jiných moderních směrů výzkumu je snadné i v tomto případě najít množství publikací, jejichž cílem je vysvětlit činnost genetického algoritmu co nejširšímu okruhu potenciálních uživatelů. Ve snaze o maximální jednoduchost a názornost takového textu se běžně stává, že funkce genetického algoritmu je demonstrována na relativně jednoduchém problému.

Z hlediska pedagogického je samozřejmě vhodné volit jednoduchý příklad, ale pokud se v souvislosti s takovým výkladem současně neuvede, že daný problém je ilustrativní a že na této úloze funguje mnohem lépe algoritmus jiný (ať již z hlediska času výpočtu či kvality nalezeného řešení), vytváří se u uživatelů zcela matoucí a chybný dojem, že genetické algoritmy jsou s výhodou použitelné na příslušné třídy problémů (viz. např. [8] a [9]).

Je proto třeba jasně a zřetelně konstatovat, že genetické algoritmy jsou sice do značné míry univerzálním a všestranně použitelným nástrojem, ale pokud pro danou třídu problémů existuje specifický algoritmus, potom nelze očekávat, že jednoduchý genetický algoritmus (bez dalších úprav, které by podstatným způsobem zvýšily jeho účinnost na daném problému) by z příslušného porovnání mohl vyjít vítězně ať již z hlediska rychlosti zpracování či kvality nalezeného řešení.

Genetické algoritmy by tedy neměly být vnímány a aplikovány jako náhrada stávajících technik, pokud takové existují a umožňují nám dosahovat na dané třídě problémů přijatelných výsledků v přijatelném čase. Je-li však třeba atakovat problém, na kterém tradiční metody selhávají nebo neexistují (kromě enumerativních metod, které jsou však obvykle prakticky nepoužitelné), potom rozhodně použití genetického algoritmu stojí za úvahu. Šance genetických algoritmů tkví také v jejich vzájemné kombinaci s tradičními technikami (tzv. hybridizaci), která může vést k velmi zajímavým výsledkům.

### 3. NEVHODNÝ PROBLÉM

Vychází-li naše motivace z opravdových potřeb a nikoliv pouhého pokušení vyzkoušet si možnosti genetických algoritmů, stále ještě je možné narazit na skutečnost, že konkrétní problém není vhodný pro řešení použitím těchto algoritmů. Goldberg [6, s. 42-47] se podrobněji zabývá například takzvanými klamnými problémy (*deceptive problems*), které se pro genetický algoritmus jeví jako velmi obtížné. Jejich společným znakem je skutečnost, že optimální řešení je izolované a obklopené řešeními s podprůměrným ohodnocením. Jelikož genetický algoritmus využívá při hledání optimálního řešení informaci obsaženou v chromosomech nadprůměrných jedinců, k dalšímu postupu ve hledání dochází kombinací částí těchto chromosomů, jenž jsou v tomto případě vzdáleny od osamocené optima. Algoritmus je tak vlastně klamán zvláštní strukturou prohledávaného prostoru a hledání je směřováno do takových oblastí prohledávaného prostoru, které jsou daleko od optimálního řešení [1].

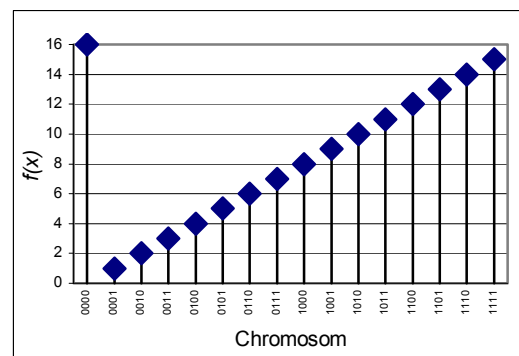
Jako jednoduchý příklad klamného problému zde může posloužit hledání maxima funkce  $f(x)$  zobrazené na obrázku číslo 1 (upraveno podle [10, s. 151]). Je zřejmé, že u takto jednoduché funkce

i při relativně malé populaci existuje velká pravděpodobnost, že se genetickému algoritmu podaří nalézt maximum již v počáteční náhodně vygenerované populaci.

Předpokládejme však nyní, že funkce  $g(x_1, \dots, x_{25})$  je definována nad pětadvaceti čtyřbitovými řetězci a platí, že

$$g(x_1, \dots, x_{25}) = f(x_1) + \dots + f(x_{25}),$$

přičemž funkce  $f(x)$  je definována dle obrázku číslo 1. Pravděpodobnost, že genetický algoritmus nyní najde již v počáteční populaci optimální řetězec (složený ze sta nul), je velmi malá a vzhledem ke způsobu fungování algoritmu a struktuře problému lze snadno experimentálně ověřit, že evoluční postup bude zaveden spíše do maxima lokálního (řetězec složený ze sta jedniček).



**Obr. 1** Příklad klamného problému  
**Fig. 1** An example of deceptive problem

Úlohu vykazující výše uvedené vlastnosti lze nejen poměrně snadno sestavit, ale kromě teoreticky vykonstruovaných úloh je možné se s větším či menším stupněm klamnosti potýkat i při řešení některých praktických problémů (viz. [6]). Přestože klamnost není vlastností konkrétního problému, ale určité reprezentace tohoto problému [4], nemusí být vůbec jednoduché, najít takovou reprezentaci, která již tuto negativní vlastnost mít nebude.

Klamné problémy bohužel nepředstavují jediné úskalí pro použití genetických algoritmů. Představíme-li si například relativně jednoduchý problém, kde bude existovat několik navzájem izolovaných řešení s nadprůměrným ohodnocením, přičemž všechny další prvky prohledávaného prostoru budou ohodnoceny podprůměrně a s pouze minimálními rozdíly či dokonce konstantně, genetický algoritmus ztratí schopnost rekombinovat a celý proces se zredukuje na hledání jehly v kupce sena, která bude odhalena jen s velmi malou pravděpodobností.

### 4. CHYBY PŘI VLASTNÍ APLIKACI

Z výčtu důvodů pro možný neúspěch aplikace genetických algoritmů, které byly uvedeny výše, je zřejmé, že i když existuje opodstatněná motivace

a problém se jeví jako vhodný pro nasazení genetického algoritmu, je možné stále ještě řadu věcí pokazit jeho nevhodnou a konkrétnímu problému nepřizpůsobenou aplikací. Protože tento poslední z výše uvedených důvodů je nejkonkrétnější a konkrétnímu nebezpečí se dá mnohem lépe čelit, nebude na škodu si jej zde rozebrat podrobněji. Jako ilustrativní problém, na němž budou možná úskalí demonstrována, byl vybrán problém obarvení grafu, na kterém genetické algoritmy dosahují relativně vysoké efektivity. Vzhledem k omezenému prostoru tohoto příspěvku se soustředíme pouze na nevhodné způsoby aplikace genetických algoritmů, přičemž efektivní způsob jejich použití na tento konkrétní problém lze nalézt například v [2].

Neorientovaným grafem rozumíme dvojici  $G = (V, E)$ , kde  $V = \{v_1, \dots, v_N\}$  je množina vrcholů grafu a  $E = \{e_1, \dots, e_M\}$  je množina hran, přičemž každá hrana  $e_k \in E$  je určena dvěma vrcholy tak, že  $e_k = (v_i, v_j)$ , kde  $v_i, v_j \in V$ , přičemž platí  $v_i \neq v_j$  (tj. jsou zakázány takzvané smyčky). Obarvením grafu potom rozumíme zobrazení  $f: V \rightarrow \mathbb{N}$ , pro které platí podmínka

$$\forall (v_i, v_j) \in E \Rightarrow f(v_i) \neq f(v_j).$$

Vyjadřeno slovy, každým dvěma vrcholům, které jsou v grafu spojené hranou, musí být zobrazením  $f$  přiřazena různá přirozená čísla. Je-li namísto přirozených čísel použita množina určitých barev, vyžaduje se, aby koncové vrcholy libovolné hrany byly obarveny různými barvami.

Tento problém se vyskytuje v mnoha různých variantách, z nichž zřejmě nejnámější jsou úlohy vyžadující nalezení minimálního počtu barev, kterými je zadaný graf možné obarvit, nebo úloha zjistit, zda daný graf je možné obarvit třemi barvami. Jelikož jsou tyto úlohy NP-úplné [5], je nepravděpodobné, že se podaří najít algoritmus, který by dokázal úlohy tohoto typu řešit v polynomiálním čase, a proto velké instance těchto problémů mohou být atakovány pouze nejrůznějšími heuristickými metodami mezi než patří i genetické algoritmy (viz. např. [3]).

#### 4.1 Použití nesprávné reprezentace

První chybou, kterou je možné při aplikaci genetického algoritmu udělat, je použití nesprávné reprezentace. Z výše uvedeného je zřejmé, že prostor, ve kterém algoritmus hledá řešení problému, je rozsáhlý. Protože kódováním je definována relace mezi potenciálními řešeními daného problému a relevantními chromosomy, nad nimiž genetický algoritmus pracuje, je třeba hledat takové kódování, které bude v ideálním případě přiřazovat jednomu řešení právě jeden chromosom. Pokud je některé řešení reprezentováno více než jedním chromosomem, vzniká zde redundance a genetický algoritmus pracuje nad množinou chromosomů, která má větší mohutnost než původně prohledávaný prostor. Protože se této redundanci v některých

případech nelze ubránit, je třeba ji alespoň minimalizovat, to jest definovat takové kódování, kde každému potenciálnímu řešení odpovídá co nejmenší počet chromosomů.

Přestože se tato myšlenka může na první pohled jevit jako zcela logická a zřejmá, je velmi snadné se takovéto chyby dopustit. Ve zvoleném ilustrativním příkladu stačí například zvolit zdánlivě přímočarou reprezentaci ve formě, kde každý gen reprezentuje jeden z vrcholů grafu a příslušnou alelou bude barva tohoto vrcholu. Jelikož při problému barvení grafu záleží spíše na počtu použitých barev a struktuře nalezeného řešení, to jest jak postupovat při obarvování vrcholů bez ohledu na skutečnost, jaké barvy jsou momentálně k dispozici, bude právě zvolená reprezentace velmi redundantní. Například chromosomy

(*žlutá, žlutá, modrá, černá, žlutá, modrá*),

(*modrá, modrá, žlutá, černá, modrá, žlutá*),

(*černá, černá, modrá, žlutá, černá, modrá*), atd.

reprezentují stále stejné řešení, které se vyznačuje tím, že používá tři barvy, přičemž první, druhý a pátý vrchol je obarven jednou barvou, třetí a šestý vrchol druhou barvou a čtvrtý vrchol musí být obarven další (dosud nepoužitou) barvou.

Z tohoto příkladu je zcela zřetelně vidět, že jednomu potenciálnímu řešení odpovídá několik různých chromosomů. Počet těchto chromosomů roste exponenciálně s počtem použitých barev a prostor, který genetický algoritmus prohledává, je mnohem větší než původní prostor, v němž se nalézají hledané řešení. Jestliže v mnoha případech je to právě velikost prostoru potenciálních řešení, kvůli níž dochází k rozhodnutí o použití genetického algoritmu, hned prvním krokem při jeho aplikaci je možné příslušnou úlohu ještě více zkomplikovat.

#### 4.2 Použití nevhodných operátorů

Dalším problémem, který může snadno vzniknout, je použití nevhodných genetických operátorů. Smyslem křížení je kombinace vhodných stavebních bloků, jejichž použitím by měla vzniknout nová individua s lepšími vlastnostmi, než které měli jejich předchůdci. Použijeme-li pro jednoduchost i nadále výše diskutovanou reprezentaci společně s operátorem dvoubodového křížení, je velmi snadné se přesvědčit o tom, že tomu tak být ani zdaleka nemusí.

Uvažujme následující dvě individua, která jsou řešením problému barvení grafu z obrázku číslo 2, a nechť druhý a čtvrtý gen jsou náhodně vybrané body křížení:

(*žlutá, žlutá, | modrá, černá, | žlutá, modrá*),

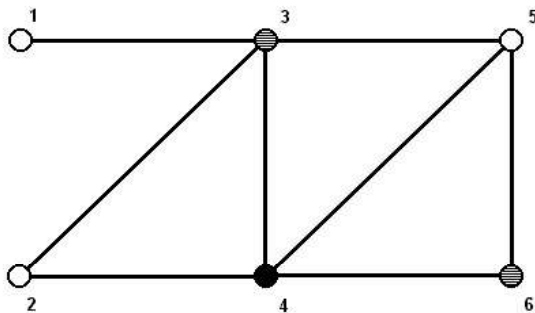
(*zelená, zelená, | žlutá, bílá, | zelená, žlutá*).

Výměnou příslušného podřetězce potom vzniknou individua

(*žlutá, žlutá, | žlutá, bílá, | žlutá, modrá*),

(*zelená, zelená, | modrá, černá, | zelená, žlutá*).

Odtud je vidět, že zatímco rodičovské chromosomy byly v obou případech řešením daného problému, ani jeden z potomků není přípustným řešením. Druhý z potomků dokonce místo původních a plně postačujících tří barev již využívá barvy čtyři. Namísto dědění dobrých vlastností po rodičích zde došlo k úplnému rozbití a zničení informace obsažené v původních chromosomech. Protože tento výsledek není dán pouze nepřilíš šťastnou volbou bodů křížení, ale zcela zřetelně je možné podobného efektu docílit i v naprosté většině dalších možností volby těchto bodů, je evidentní, že použitý operátor křížení v souvislosti se zde použitým kódováním rozhodně nefunguje ve smyslu kombinace vhodných stavebních bloků.



**Obr. 2** Příklad barvení grafu  
**Fig. 2** An example of graph colouring

Negativním způsobem může v tomto případě působit i klasická verze operátoru mutace, která náhodně mění hodnotu vybraného genu. Uvažujme například individuum reprezentované chromosomem

(*modrá, žlutá, modrá, černá, žlutá, modrá*),

kde k nalezení přípustného řešení schází již pouze změnit hodnotu prvního genu, přičemž jak lze usoudit z obrázku číslo 2, tento gen odpovídá vrcholu, který může nabývat libovolné barvy kromě modré (barva vrcholu číslo 3). V ideálním případě operátor mutace změní právě tuto alelu na hodnotu *žlutá* nebo *černá* a výsledný chromosom reprezentuje dokonce optimální řešení, neboť příslušný graf nelze obarvit méně než třemi barvami. Pokud tato mutace změní hodnotu prvního genu tak, že bude použita nová (další) barva, chromosom bude reprezentovat přípustné řešení daného problému, ale z hlediska počtu použitých barev již nikoliv řešení optimální.

Až dosud jsme předpokládali, že se mutace bude týkat právě a pouze prvního genu, ale jelikož všechny geny mají stejnou pravděpodobnost být subjektem této operace, je nutné pečlivě zvážit i tuto alternativu. Na tomto příkladu je jasné vidět, že při změně hodnoty libovolného ze zbývajících genů (reprezentujících vrcholy číslo 2 až 6) použitím některé ze stávajících barev {*modrá, žlutá, černá*} vzhledem k vlastnostem grafu zřejmě dojde k vytvoření nového konfliktu, jelikož při malé pravděpodobnosti, s níž je operátor mutace

aplikován, je současná změna hodnot několika genů velmi nepravděpodobná. Při použití barvy další, v chromosomu dosud nezastoupené, sice žádný nový konflikt v obarvování nevznikne, ale počet použitých barev se zvyšuje a tímto způsobem se algoritmus vzdaluje od nalezení optimálního řešení. Analogicky jako při zkoumání vlastností operátoru křížení výše je tedy možné konstatovat, že klasický operátor mutace působí destruktivně a spíše komplikuje cestu genetického algoritmu k nalezení optimálního řešení.

Zcela samostatnou kapitolu v přehledu možných chyb při aplikaci genetických algoritmů by bylo možné napsat v souvislosti s problémem volby pravděpodobností, s jakými jsou jednotlivé genetické operátory aplikovány. Protože pro nastavení těchto parametrů neexistují obecně platná východiska a navíc je třeba brát v úvahu i jejich interakci s dalšími parametry genetického algoritmu, mezi než zcela jistě patří například velikost populace či použité reprodukční schéma pro přechod od jedné generace individuí ke generaci následující, je možné si snadno představit negativní důsledky nevhodné volby těchto parametrů na vlastní průběh a konvergenci algoritmu. Výsledky celé řady experimentálních studií na téma jakým způsobem nastavovat parametry genetických algoritmů lze nalézt například v [11, s. 175-177].

### 4.3 Použití nevhodné ohodnocující funkce

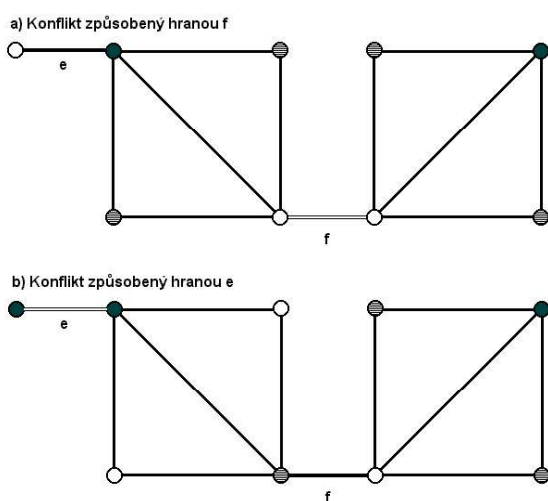
Další důležité rozhodnutí, které je plně v kompetenci uživatele a může významně ovlivnit výkonnost genetického algoritmu, je volba vhodné ohodnocující funkce. Přestože se na první pohled může zdát jako samozřejmost odvíjet ohodnocení od počtu správně (resp. nesprávně) obarvených vrcholů, i zde existují možnosti navrhnout alternativní ohodnocení, která vzhledem k charakteru tohoto problému budou lépe rozlišovat mezi jednotlivými individui v populaci a zefektivní tak vlastní evoluční proces.

K ilustraci je možné v tomto případě použít obrázek č. 3, kde je dvakrát nakreslen tentýž graf, ale pokaždé s jiným částečným obarvením. Přestože v obou případech je v grafu zcela zřetelně vidět jediný konflikt (způsobený zvýrazněnou hranou *e*, respektive hranou *f*) a dle nejjednoduššího způsobu výpočtu ohodnocující funkce, který závisí pouze na počtu konfliktů při daném částečném obarvení, by tedy oba případy byly ohodnoceny stejně, je velmi snadné si povšimnout, že mezi nimi je podstatný kvalitativní rozdíl.

Zatímco v prvním případě je zřejmě okamžik nalezení přípustného obarvení (předpokládáme pro tuto chvíli, že máme k dispozici pouze tři barvy) ještě velmi vzdálen a bude nutné měnit barvy několika vrcholů, aby se podařilo konflikt na hraně *f* odstranit, v druhém případě, kdy konflikt je způsoben hranou *e*, stačí již změnit pouze obarvení

jediného vrcholu, který navíc nemá žádné další sousedy a tudíž tato změna nevyvolá žádné další vynucené změny.

Samozřejmě, že „jednoduchost“ a „komplikovanost“ obou situací byla v předchozích řádcích posuzována z jakéhosi makro-pohledu, který nám umožňuje tuto rozdílnost zcela zřetelně vidět. Je ovšem otázkou, jestli genetický algoritmus kromě schopnosti tento rozdíl postřehnout, čehož lze docílit právě definicí rafinovanější ohodnocující funkce, bude též vybaven účinnými mechanismy, které mu umožní zdánlivě jednoduché operace (z našeho makro-pohledu) nad příslušnými chromosomy efektivně provádět. Ale to neznamená nic jiného než návrh vhodných operátorů a potažmo i vhodné reprezentace problému, čímž se kruh uzavírá.



**Obr. 3** Různé druhy konfliktů

**Fig. 3** Different types of conflicts

## 5. KDY GENETICKÉ ALGORITMY POUŽÍT?

Přestože lze v relevantních publikacích nalézt množství úspěšných aplikací, zcela zřetelně existuje i mnoho takových, kde genetické algoritmy nefungují dobře. Před přistoupením k vlastní aplikaci genetického algoritmu na konkrétní problém by bylo jistě vhodné znát a mít k dispozici alespoň nějaké indicie, které by umožnily poznat, zda genetický algoritmus je či není v daném případě vhodnou metodou. Ani zde samozřejmě nelze očekávat rigorózní odpověď na takto formulovanou otázku, ale přesto existují jakási spíše intuitivní vodítka, která jsou poměrně široce sdílána v komunitě věnující se problematice genetických algoritmů.

Předně lze s poměrně velkou jistotou konstatovat, že mezi příznaky signalizující, že použití genetického algoritmu bude opodstatněné a existuje dobrá šance konkurovat jiným univerzálním metodám, které nevyžívají žádné specifické znalosti o konkrétním problému, lze dle [11] zařadit následující indicie:

- prohledávaný prostor je velký
- prohledávaný prostor není vyhlazený a unimodální, nebo
- struktura prohledávaného prostoru je komplikovaná
- k řešení úlohy není třeba nutně najít globální optimum, ale přijatelné řešení v přijatelném čase

Z výše uvedeného je zřejmé, že klíčovou otázkou při rozhodování, zda použít genetický algoritmus při řešení konkrétního problému, je otázka, jak vypadá prostor, který má být tímto způsobem prohledáván. Jestliže je tento prostor vzhledem k jeho velikosti možné projít tradičními exhaustivními postupy v přijatelném čase, je jisté, že tímto způsobem nalezneme globální extrém, zatímco genetický algoritmus může snadno nalézt spíše řešení sub-optimální.

Je-li prohledávaný prostor hladký nebo unimodální, potom různé gradientové metody budou při řešení příslušné úlohy zcela přirozeně mnohem efektivnější než genetický algoritmus. Stejný závěr platí v případě, kdy prohledávaný prostor svojí strukturou a vlastnostmi umožňuje navrhnout dobře fungující specifické heuristiky, které efektivně využívají příslušných znalostí o tomto prostoru.

Jestliže však prostor, ve kterém se hledá řešení, kvůli své velikosti či struktuře nic podobného nepřipouští a pokud lze navrhnout a použít efektivní reprezentaci tohoto prostoru, potom genetické algoritmy představují alternativu coby mocný heuristický nástroj k atakování příslušného problému.

Tyto indicie jsou samozřejmě spíše intuitivní a na jejich základě nelze předem přesně rozhodnout, zda na dané konkrétní úloze bude genetický algoritmus schopen konkurovat jiným metodám. Jak bylo však ukázáno výše, chování genetického algoritmu bude vždy do značné míry záviset na vhodné volbě kódování, použitých operátorech a nastavení jednotlivých parametrů, které je třeba volit tak, aby genetický algoritmus byl co nejlépe přizpůsoben charakteru řešeného problému.

## 6. ZÁVĚR

Naším cílem bylo přinést realistický pohled na oblast genetických algoritmů ve smyslu nalezení obecných důvodů, které mohou vést k selhání genetického algoritmu při řešení konkrétního problému.

Genetické algoritmy jsou bezesporu velice obecným, univerzálním, robustním a snadno použitelným nástrojem pro řešení široké škály praktických problémů. Na druhé straně je zřejmé, že chceme-li tento nástroj používat rozumně a efektivně, musíme znát jeho slabiny a omezení. Kromě toho je třeba genetické algoritmy chápat nikoliv jako konkurenční nástroj či náhradu klasických optimalizačních technik, ale vyhledávat

možnosti výhodné synergie, jenž může být dosažena skrze proces hybridizace kdy genetický algoritmus je vzhledem ke své obecnosti poměrně snadné obohatit o význačné prvky algoritmů tradičních.

Podobně jako při zacházení s jakýmkoli jiným nástrojem, pokud je používán v nesprávné situaci nebo na nesprávný problém, velmi pravděpodobný neúspěch není chybou genetických algoritmů, ale toho, kdo je nevhodně aplikuje.

## LITERATURA

- [1] Deb, K. & Goldberg, D. E.: *Analyzing Deception in Trap Functions*. University of Illinois at Urbana-Champaign, IlliGAL Report No. 91009, 1991.
- [2] Eiben, A.E., Raué, P.E., Ruttkay, Z.: *GA-easy and GA-hard Constraint Satisfaction Problems*. In Proceedings of the ECAI-94 Workshop on Constraint Processing, no. 923 in LNCS, Springer-Verlag, 1995, s. 267-284.
- [3] Eiben, A.E., van der Hauw, J.K., van Hemert, J.I.: Graph Coloring with Adaptive Evolutionary Algorithms. *Journal of Heuristics*, 4(1), s.25-46, 1998.
- [4] Forrest, S. Mitchell, M.: What Makes a Problem Hard for a Genetic Algorithm? Some Anomalous Results and Their Explanation. *Machine Learning*, vol. 13, no. 2/3, 1993.
- [5] Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freedman and Co., 1979.
- [6] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [7] Goldberg, D.E.: *Genetic and Evolutionary Algorithms in the Real World*. University of Illinois at Urbana-Champaign, IlliGAL Report No. 99013, 1999.
- [8] Gordon, V., Böhm, A., Whitley, D.: *A Note on the Performance of Genetic Algorithms on Zero-One Knapsack Problems*. In Proceedings of the 1994 ACM symposium on Applied computing, 1994, s. 194 - 195.
- [9] Hynek, J.: *The Zero-One Knapsack Problem and Genetic Algorithms*. In Proceedings of the 19th International Conference on Mathematical Methods in Economics 2001, Hradec Králové 2001, s. 73-78.
- [10] Lažanský, J.: *Evoluční výpočetní techniky*. In Mařík V., Štěpánková O., Lažanský J. a kol.: *Umělá inteligence 3*. Academia, Praha 2001, s. 117-160.
- [11] Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 1996.
- [12] Wolpert, D.H., Macready, W.G.: *No Free Lunch Theorems for Search*. Technical Report SFI-TR-95-02-010, The Santa Fe Institute, 1995.

## BIOGRAPHY

**Josef Hynek** got his Ph.D. in theoretical informatics from The School of Mathematics and Physics, Charles University in Prague in 1998. He has been working for The Faculty of Management and Information Technology, University of Hradec Králové since 1991. Currently he has a position of a vice-dean and is responsible for science, development and international relations. His scientific focus is on evolutionary algorithms and their applications.