

**Evoluční algoritmy využívající
umělé imunitní systémy**

**Evolutionary Algorithms Using
Artificial Immune System**

Zadání diplomové práce

Student: **Bc. Lukáš Palatinus**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Evoluční algoritmy využívající umělé imunitní systémy**
Evolutionary Algorithms Using Artificial Immune System

Zásady pro vypracování:

Cílem práce je naimplementovat diferenciální evoluci (různé typy DE/rand/1/bin, DE/best/1/bin atd.), samoorganizující migrační algoritmus, rojení částic a genetický algoritmus, které budou využívat umělý imunitní systém za účelem nalezení globálního optima.

1. Student nastuduje evoluční algoritmy zmíněné výše.
2. Student nastuduje princip umělého imunitního systému.
3. Student implementuje zmíněné algoritmy a do nich zakomponuje umělý imunitní systém.
4. Student provede statistickou analýzu výsledků.

Seznam doporučené odborné literatury:

- [1] Dipankar Dasgupta: Artificial Immune Systems and Their Applications, Springer-Verlag, ISBN: 3-540-64390-7, November 1998.
- [2] De Castro, Leandro Nunes, and Jonathan Timmis: Artificial immune systems: a new computational intelligence approach, Springer, 2002.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Lenka Skanderová**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015




doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostravě 20. dubna 2015

..........

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 20. dubna 2015

..........

Rád bych na tomto místě poděkoval své vedoucí diplomové práce Ing. Lence Skanderové za vedení, její podporu, trpělivost a čas. Rovněž patří můj dík rodině a blízkým přátelům za podporu při studiu a tvorbu potřebného zázemí.

Abstrakt

Dnes jsou umělé imunitní systémy známy jako jedno z počítačových vědeckých odvětví, které se nechalo inspirovat biologickým imunitním systémem. Tato diplomová práce se nechala inspirovat řešením vědců X. He a L. Han [23]], kteří se zabývali tématem využití umělého imunitního systému v diferenciální evoluci. Cílem této práce je provést implementaci práce zmiňovaných autorů a rozšířit funkcionalitu o pozitivní selekci a další druhy diferenciální evoluce. Následně je provedena sada experimentů pomocí testovacích problémů a jejich následné vyhodnocení.

Klíčová slova: evoluční algoritmy využívající umělé imunitní systémy, umělý imunitní systémy, algoritmus pozitivní selekce, algoritmus negativní selekce, evoluční algoritmy, diferenciální algoritmy

Abstract

Nowadays artificial immune systems are known as one of the branches of computer science industry, inspired by biological immune system. This thesis was inspired by solution of scientists X. He and L. Han [23] who dealt with the theme of using artificial immune system in differential evolution. The aim is to implement the work of mentioned authors and extend the functionality of positive selection and other types of differential evolution. Subsequently, a set of experiments is executed using testing problems followed by their evaluation.

Keywords: evolutionary algorithms using artificial immune systems, artificial immune systems, algorithm positive selection, algorithm negative selection, evolutionary algorithms, differential algorithms

Seznam použitých zkratk a symbolů

- AIS – Umělý imunitní systému (z anglického Artificial immune system)
- TLR – Toll-like receptory
- CTL – Cytotoxické T buňky (z angl. Cytotoxic T lymphocytes)

Obsah

1	Úvod	7
1.1	Obsah kapitol	7
2	Biologický imunitní systém	9
2.1	Obranné mechanismy imunitního systému	9
2.2	Poruchy imunitního systému	15
3	Umělý imunitní systém	17
3.1	Prvky a principy	17
4	Cíl práce	31
4.1	Evoluční algoritmy	31
4.2	Diferenciální evoluce	35
5	Implementace a popis aplikace	43
5.1	Článek „A Novel Binary Differential Evolution Algorithm Based On Artificial Immune System“	43
5.2	Třídy aplikace	43
5.3	Popis implementace	44
6	Experimenty	51
6.1	Testovací problémy	51
6.2	Provedení experimentů	53
6.3	Výsledky experimentů	54
6.4	Závěr experimentů	60
7	Závěr	61
8	Reference	63

Seznam tabulek

1	Doporučené hodnoty	36
2	Parametry použité při experimentech	54
3	Parametry použité pro multimodální problém	54
4	Goldbergův problém pro negativní selekci	55
5	Goldbergův problém pro pozitivní selekci	55
6	Bipolarův problém pro negativní selekci	55
7	Bipolar problém pro pozitivní selekci	55
8	Mühlebeinův problém pro negativní selekci	56
9	Mühlebeinův problém pro pozitivní selekci	56
10	Clercův problém pro negativní selekci	56
11	Clercův problém pro pozitivní selekci	56
12	Multimodální problém pro negativní selekci	57
13	Multimodální problém pro pozitivní selekci	57

Seznam obrázků

1	Fagocytóza [47]	11
2	Útok NK buňky pod mikroskopem [12]	12
3	Třídy imunoglobulínů [16]	12
4	Tvar proteinů na B buňkách	14
5	Komplement s antigenem [4]	19
6	Naplnění množiny P, pozitivní selekce	22
7	Filtrace nového prvku, pozitivní selekce	23
8	Naplnění množiny P, negativní selekce	25
9	Filtrace nového prvku, negativní selekce	26
10	Klonální selekce	28
11	Reprezentace třídy Jedince v třídním diagramu	32
12	Křížení	33
13	De Jongova funkce	34
14	Schwefelova funkce	35
15	Diagram aktivit diferenciální evoluce	39
16	De Jongova a Schwefelova funkce	41
17	Výpočet binárního mutačního vektoru [23]	45
18	Diagram diferenciální evoluce využívající umělý imunitní systém	48
19	Manipulace s vektorem	49
20	Goldbergs order-3	57
21	Bipolars order-6	58
22	Mühlebeins order-5	58
23	Clerc's Zebra3	59
24	Multimodální problém	59

1 Úvod

Umělé imunitní systémy (AIS, z anglického názvu Artificial immune system) jsou dnes známy jako jedna z počítačových oblastí, jejichž potenciál se rozšiřuje do mnoha dalších počítačových odvětví. Umělé imunitní systémy se nechaly inspirovat vlastnostmi a principy biologického imunitního systému, který chrání tělo před napadením různými druhy infekcí. Jedná se o druh umělé inteligence, která reaguje a učí sama sebe jak se nejlépe zachovat v určitých situacích. Této logiky se nejvíce využívá zejména v počítačové bezpečnosti, ale ne jen zde. Jde například i o optimalizaci existujících algoritmů, které řeší komplexní problémy a snaží se díky logice umělého imunitního systému zefektivnit a urychlit jejich řešení.

Cílem této práce je skloubit dvě odlišná odvětví výzkumu – diferenciální evoluci a umělé imunitní systémy za účelem zvýšení rychlosti konvergence diferenciální evoluce ke globálnímu optimu. Z tohoto důvodu jsme se inspirovali prací X. He a L. Han [23], kteří spojili princip umělého imunitního systému, zejména pak negativní selekci, a diferenciální evoluci a otestovali toto spojení na 5 základních funkcích. Z výsledků vyplývá, že se jejich metoda osvědčila. Na základě informací, které jsme načerpali, jsme se rozhodli implementovat diferenciální evoluci, která využívá negativní selekci a diferenciální evoluci, která využívá pozitivní selekci. Pro experimenty bylo vybráno 5 variant diferenciální evoluce – DE/rand/1/exp, DE/best/1/bin, DE/rand/2/bin, DE/current/to/best/1/bin a DE/current/to/best/1/bin. Rychlost konvergence pak testujeme na 5 funkcích podle [23]. Pro porovnání výsledků obou strategií jsme využili Wilcoxon Signed Rank test. Jedná se o statistický neparametrický test, který nám poslouží pro testování hypotéz.

1.1 Obsah kapitol

Abychom byli vůbec schopni provést implementaci diferenciální evoluce využívající umělý imunitní systém, musíme v první řadě pochopit základy, vlastnosti a principy biologického imunitního systému (v kapitole 2). V třetí kapitole bude popsáno několik algoritmů, které umělý imunitní systém používají, spolu s příklady jejich využití v praxi. Následně je třeba porozumět tomu, jak funguje diferenciální evoluce a k čemu nám slouží (kapitola 4.2). V rámci praktické části diplomové práce provedeme implementaci diferenciální evoluce využívající vlastností a principů umělého imunitního systému pro svůj prospěch. Abychom se přesvědčili, že je implementovaný algoritmus efektivní, v kapitole 6 bude provedena řada experimentů.

2 Biologický imunitní systém

Lidské (a nejen lidské) tělo je velmi křehké a zranitelné. Denně jsme ohrožováni množstvím patogenů různých druhů. Tělo organismu má schopnost se proti napadení bránit. Tato schopnost se nazývá imunitní systém (imunita pochází z latinského *immunitas* – v překladu odolnost). Imunitní systém je schopný rozeznat vlastní prvky od těch cizorodých a provést případný zásah proti nim. Základní funkcionalitu a principy, které biologický imunitní systém poskytuje, budou následně popsány.

Naše tělo se setkává s následujícími druhy infekcí, které mohou napadnout náš organismus:

1. **Patogeny** -nebo také infekční prvky jako například viry, bakterie, houby a paraziti, jsou biologické prvky, které způsobují nakažení svého hostitele. Patogeny hledají obživu, úkryt, možnost rozmnožení nebo pouhý přesun k novému hostiteli či prostředí. Způsob rozmnožování patogenů a změna hostitelů jsou rozličné. Některé patogeny jsou přenášeny vzduchem, některé dotykem, tržnou ranou nebo pohlavním stykem. Mezi nejznámější patogeny, které nás neohrožují pouze na zdraví, ale také na životech, patří například vir HIV, cholera, chřipka, hepatitida A, zarděnky a mnoho dalších.
2. **Antigeny** – jsou substance přichyceny na cizích organismech, které se snaží vniknout do našeho těla. Imunitní systém se snaží tyto antigeny rozpoznat a následně aktivovat obrannou reakci, která se je pokusí zničit (podrobnější popis bude podán v další kapitole).

Účelem našeho imunitního systému je zabránit nebo alespoň omezit napadení infekcí. Imunitní systém je schopný rozlišit zdravé buňky od těch nakažených. Avšak tato detekce není nijak jednoduchá. Postupem času se patogeny vyvíjejí a mutují. Není tedy jednoduché je rozeznat ani proti nim zakročit. Látky, které dříve dokázaly patogen nalézt a neutralizovat, nemusí již být schopné detekovat tyto vyvinuté a zmutované patogeny [7][8].

2.1 Obranné mechanismy imunitního systému

V této části se zaměříme na podrobnější popis fungování imunitního systému. Obranný mechanismus imunitního systému můžeme dělit na dvě hlavní části: specifickou a nespecifickou obranu.

2.1.1 Nespecifická obrana

Nespecifické obranné mechanismy jsou taktéž nazývány jako vrozené nebo neadaptivní. Veškeré tyto obranné schopnosti získává dítě skrz placentu své matky při narození, tedy díky DNA. Dítě se rodí s imunitou vůči nákazám a chorobám, proti kterým je imunní i jeho/její matka.

Nespecifická obrana nemá „paměť“. To znamená, že brání tělo při styku s patogenem vždy stejným způsobem (na rozdíl od získané imunity – více v další kapitole).

2.1.1.1 První vrstva

První vrstvou nespecifické obrany je tzv. fyzická bariéra, která brání vniknutí patogenů do těla. Mezi takové fyzické bariéry patří například kůže a její sekrety, řasinky (pohyblivé výběžky na povrchu buněk dýchacích cest), slzy, sliny a mnoho dalších.

Může však dojít k poškození fyzické bariéry tedy protržení kožní tkáně, které pak umožní vniknutí cizorodých látek do těla organismu, který mohou vážně ohrozit. Proto existuje v nespecifické obraně ještě další vrstva [4][21].

2.1.1.2 Druhá vrstva

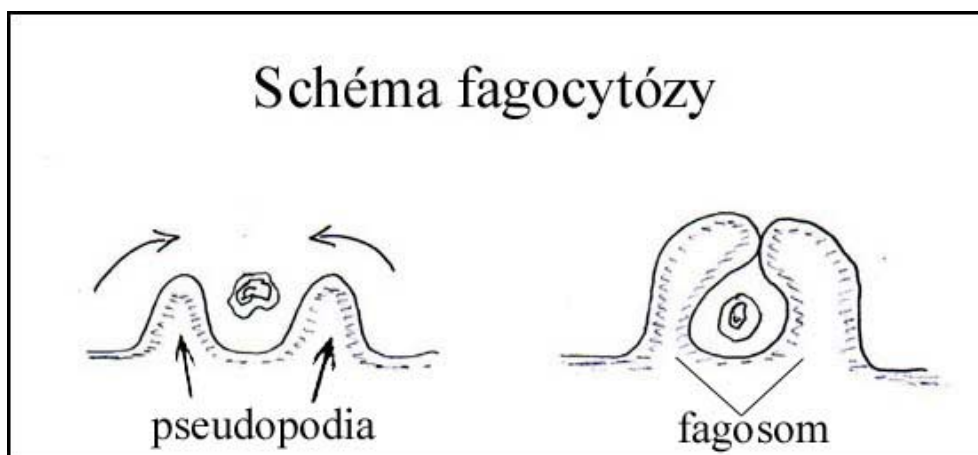
Pokud se infekci podaří projít přes první fyzickou vrstvu, existuje vrstva druhá, která zajišťuje další obrannou linii. V této části už dochází k vnitřnímu boji s infekcí. V krevním řečišti putuje velké množství buněk, které proplouvají celým naším tělem a „hlídkují“. Tyto buňky mají po celém svém povrchu velké množství senzorů (čidel), kterými dokáží rozeznat hrozbu, jež organismu hrozí. Jakmile se tak stane, vyšle buňka signál do celého těla, ten přivolá další obranné buňky a aktivuje získanou imunitu (v kapitole 2.1.2).

Mezi obranné buňky této vrstvy řadíme neutrofilny, NK buňky, imunoglobuliny, dendritické buňky a TLR.

2.1.1.2.1 Neutrofilny

Neutrofilny (nebo také neutrofilní granulocyty) jsou jedním z druhů bílých krvinek (leukocytů). Tvoří celkově 33 % všech bílých krvinek [9]. Neutrofilny hrají významnou roli při útoku infekce na organismus, kde působí zejména proti bakteriím, které nevnikají do nitra buněk. Spolu s dalšími buňkami vytvářejí zánět, který patří mezi základní obranné mechanismy těla. I přesto, že část neutrofilů putuje tělem, drtivá většina se nachází v kostní dřeni, dokud není organismus napaden infekcí. Jakmile dojde k proniknutí infekce do těla, neutrofilny okamžitě zaútočí na bakterie. Tyto buňky nečekají na pokyny imunitního systému jako je tomu například u lymfocytů, jež čekají na aktivační signál.

Obranným mechanismem neutrofilů je fagocytóza. Jedná se o schopnost buňky pohlcovat různé částice, v tomto případě bakterie. Neutrofilý obsahují ve svém těle enzymy, které po fagocytóze vylučují k usmrcení pohlcených bakterií. Pokud je ovšem cizorodých látek větší množství, než jaké by neutrofilý dokázaly zvládnout, vypustí látky zapříčínující vznik zánětu. Neutrofilý nemají schopnost se dělit ani reprodukovat své enzymy, tudíž po usmrcení určitého počtu buněk umírají a stávají se součástí zánětlivého ložiska. Odumřelé neutrofilý a bakterie tvoří hnis. Maximální životnost neutrofilů je přibližně 5 dní.



Obrázek 1: Fagocytóza [47]

2.1.1.2.2 NK buňky

NK buňky (zkratka z anglických slov „Natural killers“, v překladu „Přirození zabíjáci“) jsou dalším druhem bílých krvinek a jsou taktéž součástí vrozeného imunitního systému. Představují 5 – 10 % celkového množství lymfocytů v krevním řečišti. NK buňky dokážou pracovat nezávisle na jiných buňkách. To znamená, že nemusí čekat na pokyny od imunitního systému, aby zaútočily na cizorodé látky, které napadly náš systém. Z tohoto důvodu lékaři a vědci považují NK buňky za první a velmi důležitou obrannou linii proti rakovinným a virovým buňkám [12].

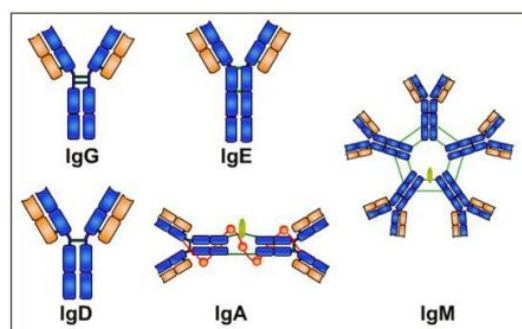
Při útoku se NK buňky navážou na cizorodou buňku a začnou proti této hrozbě vypouštět smrtelnou dávku toxinů. Tyto toxiny vytvoří otvory v membráně cizorodé buňky, ty následně zapříčiní zánik membrány a tím rozdělení DNA na mnoho částí (tzv. apoptóza – programovaná buněčná smrt). Jakmile NK buňky vypustí toxiny, dokáží se odělit a zneškodnit další buňky.



Obrázek 2: Útok NK buňky pod mikroskopem [12]

2.1.1.2.3 Imunoglobuliny

Jsou biologicky významné a aktivní biomolekuly. Imunoglobuliny jsou protilátky, které pomáhají buňkám (například neutrofilům) rozpoznat specifické cizorodé buňky v těle. Protože neutrofilové a další obranné buňky mají záporný elektrický potenciál, je poměrně obtížné navázat spojení s infikovanou buňkou v případě, že má stejný elektrický potenciál. Imunoglobuliny dokážou tento záporný elektrický potenciál infikované buňky neutralizovat a tím usnadnit práci neutrofilům.



Obrázek 3: Třídy imunoglobulinů [16]

2.1.1.2.4 Dendritické buňky

Tento druh buněk patří mezi ty nejdůležitější antigeny, které vrozená imunita obsahuje. Dendritické buňky dokážou zmobilizovat naivní T a B buňky (popsáno podrobněji v kapitole Specifická imunita) a tím i regulovat míru imunitní odpovědi organismu. Jsou rozmístěny téměř ve všech tkáních. Jakmile se objeví nějaký neznámý antigen, dendritické buňky jej pohltnou a poté s ním cestují do sekundárních lymfatických orgánů (například slezina). Zde předávají antigen T buňkám, které započnou účinný vývoj odpovědi imunitního systému.

Dendritické buňky jsou vybaveny TLR receptory (popsáno níže), díky nimž jsou schopny rozpoznat antigen.

2.1.1.2.5 TLR

Zkratka TLR (v angl. Toll-like receptory) znamená v překladu „receptory podobné genu Toll“ (geny a proteiny nalezené v octomilce – dvoukřídlý hmyz). Dnes jsou považovány za nejdůležitější receptorové molekuly vrozeného imunitního systému. Jejich prací je detekovat charakteristické mikrobiální produkty (například lipoproteiny, bakteriální DNA, virové RNA a další) a vyvolat následnou obrannou reakci systému [4][44].

Existuje celkem 10 tříd TLR receptorů: TLR-1 až TLR-10 a každá třída má svou funkci a zaměření.

2.1.2 Specifická obrana

Je nazývána také jako adaptivní nebo získaná. Specifická imunita vstupuje do boje proti patogenům, které vnikly do těla až ve chvíli, kdy obdrží signál od nespecifické imunity. Její aktivace však trvá řádově několik dní (přibližně pět dní).

Obrovskou výhodou specifické imunity je schopnost pamatovat si patogeny, kterými byl organismus v minulosti napaden. To znamená, že jakmile dojde k opětovnému útoku patogenu, specifická imunita dokáže zareagovat a vygenerovat obranné látky mnohem rychleji. Nejdůležitějšími buňkami této imunity jsou T a B lymfocyty (popsány níže). Tyto se po úspěšném zlikvidování infekce promění v paměťové buňky. V dnešní době je možné umělé vkládání těchto paměťových buněk do těla, tedy očkováním. Pomocí očkování lze dodat do těla informace o patogenech, se kterými se tělo nesešlo, a nese informace, jak se proti nové hrozbě bránit.

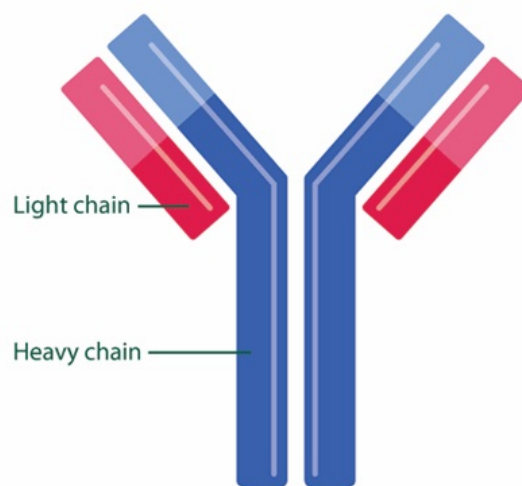
2.1.2.1 Lymfocyty B a látková imunita

Lymfocyty typu B, také nazývané jako B buňky (z lat. Bursa Fabricii – imunitní orgán ptáků, u nichž byla tato buňka poprvé objevena), jsou buňky produkující protilátky

(imunoglobuliny), které jsou základním kamenem látkové (nebo také humorální) imunity. Tyto protilátky napadají patogeny a postupně narušují jejich strukturu, což vede k jejich zničení.

Buňky B jsou tvořeny v kostní dřeni, odkud se pak stěhují do sleziny a dalších lymfatických tkání, kde dozrávají. Jsou pokryty zhruba 10 000 proteiny ve tvaru Y (obrázek 4) nazývané protilátky (antibody), ty se navazují na určité typy patogenů pomocí B buňkových receptorů (BCR – v angl. B-cell receptors).

K aktivaci B buňky dojde díky rozpoznání antigenu receptory nebo je aktivována T buňkou, případně antigenem samotným. Aktivace B buněk má za následek jejich rychlé buněčné rozmnožení a vytvoření zárodkových center, kde nové B buňky dozrají v plazmatické buňky (tvoří protilátky likvidující antigen) nebo paměťové buňky. Všechny tyto buňky jsou klonem původní buňky, tím pádem rozliší stejný typ patogenu.



Obrázek 4: Tvar proteinů na B buňkách

2.1.2.1.1 Lymfocyty T a buněčná imunita

T buňky jsou velmi důležitým druhem lymfocytů v našem těle. Jsou vytvářeny, stejně jako B lymfocyty v kostní dřeni, ale poté putují do brzlíku (v angl. Thymus – odkud název T buňka), tam procházejí složitým vývojem. Jakmile dozrají, opouštějí brzlík a putují do krve jako naivní (neznající cizí antigeny) T lymfocyty.

Na rozdíl od B lymfocytů, T lymfocyty nerozeznávají volně plovoucí antigeny v krvi. Jejich povrch je pokryt „Toll-like“ receptory, které dokáží „vidět“ fragmenty antigenů na povrchu infikovaných nebo rakovinných buněk. Existují dva druhy T buněk:

1. **Pomocník T (také Th buňky)** –koordinují odpověď imunitního systému na antigen. Některé Th buňky stimulují další buňky, aby začaly produkovat protilátky. Jiné zase, aby začaly antigen pohlcovat (fagocytóza), a v posledním případě, aby aktivovaly další T buňky.
2. **Cytotoxické T buňky (zkr. CTL – z angl. Cytotoxic T lymphocytes)** – nazývané taktéž jako „zabijácké“ T buňky, které zaútočí na buňky nesoucí cizorodé nebo abnormální molekuly na jejich povrchu. CTL jsou velmi užitečné proti virům, které se často schovávají před dalšími částmi imunitního systému. CTL dokážou rozpoznat malé fragmenty těchto virů, které vyčnívají ven z buňky, a zahájí útok proti nim.

2.2 Poruchy imunitního systému

U některých jedinců může dojít k poruše imunitního systému. Ať již při narození nebo během života. Jakékoliv poškození nebo narušení imunitního systému může vést k pomalejšímu uzdravování. V dnešní době existuje velké množství těchto poruch, například autoimunitní onemocnění, alergické onemocnění, imunodeficience a mnoho dalších (více informací i [22][26]).

3 Umělý imunitní systém

Základy umělého imunitního systému (zkr. AIS z anglického Artificial immune systém) položili J. D. Farmer, A. Perelson a N. H. Packard ve svém článku [1] v roce 1986, kde pojednávali o několika teoretických modelech imunitní sítě. Od této práce se pak odvíjel bezpočet dalších, jež dále rozšiřovaly a blíže specifikovaly fungování umělého imunitního systému [38]. Zajímavá je práce S. Forresta a A. S. Perelsona [37], ta popisuje proces rozpoznávání klonální selekce a dlouhodobý vývoj genů. Dalším příkladem je stať R. Halavatiho, S. B. Shourakiho, M. J. Heraviho, B. J. Jashmiho [39], jenž pojednává o rozšíření algoritmu ClonalG, který nazývají symbiotický umělý imunitní systém (SymbAIS). Ten podle autorů dokáže řešit i jiné problémy, které ClonalG řešit nedokáže.

V této kapitole je uveden popis a některé základní algoritmy, které se týkají umělého imunitního systému.

3.1 Prvky a principy

V umělém imunitním systému existuje řada prvků a principů, které je třeba patřičně popsat a rozlišit. Bez těchto znalostí není možné provést správnou implementaci umělého imunitního systému.

3.1.1 Prvky

3.1.1.1 Množina S

Nebo také množina vlastních (z angl. self) prvků. V biologickém systému můžeme do této množiny zařadit buňky, které do těla organismu patří. Právě pro Self prvky chceme zařídit ochranu imunitním systémem proti patogenům.

3.1.1.2 Množina P

Označujeme ji jako množinu řešení, která obsahuje obranné buňky imunitního systému, jako jsou imunoglobuliny, TLR receptory a také lymfocyty typu T a B. Tyto buňky jsou schopny v organismu rozlišit cizorodé buňky od vlastních. Může nastat situace, kdy prvky P mohou rozeznat a označit za cizí buňky i prvky z množiny S. V biologii se takový jev nazývá autoimunitní problém (například cukrovka, lupénka atd. [26]). Z tohoto důvodu existuje množina A.

3.1.1.3 Množina A

Tato množina bude vždy obsahovat pouze ty prvky z množiny P, které dokážou rozeznat pouze antigenní nebo patogenní buňky (buňky nepatřící do množiny S). V případě

použití dalších druhů selekcí (vysvětleno v další kapitole) může množina A obsahovat prvky, které naopak rozeznají pouze vlastní prvky.

3.1.1.4 Afinita

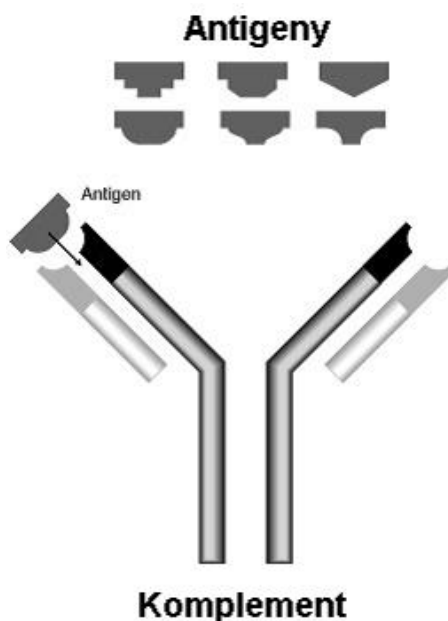
Afinita je číselná hodnota udávající nutnou míru podobnosti dvou prvků (popsáno v kapitole 3.1.2.2). Při využití jednoho z algoritmů rozpoznávání prvků, například Hammingovy vzdálenosti (3.1.2.1), získáme hodnotu rozdílnosti dvou prvků. Tato hodnota je posléze porovnána s afinitou. Je-li hodnota afinity nižší, došlo k nalezení stejného prvku (ať už dobrého či špatného) a naopak.

3.1.2 Principy

V této kapitole budou vysvětleny základní principy umělého imunitního systému, jako je rozpoznávání prvků, díky němuž jsme schopni určit vzájemnou podobnost dvou porovnávaných prvků. Další podkapitola se zabývá třemi základními druhy algoritmických selekcí AIS. Patří mezi ně pozitivní selekce, negativní selekce a klonální selekce.

3.1.2.1 Rozpoznávání prvků

V adaptivní i neadaptivní imunitě jsou buňky, které dokážou rozpoznávat buňky cizorodé, například výše popsané neutrofilny, dendritické buňky nebo lymfocyty typu B a T. Každá z těchto buněk má na svém povrchu receptory, jež dokážou rozeznat určitý vzor. Takové části buňky se říká komplement (také antibody). Těchto komplementů na povrchu buněk je nezměrné množství a každý z nich dokáže rozeznat určitý vzor (antigen). Princip rozpoznávání můžeme přirovnat k otevírání zámku klíčem (na obrázku 5).



Obrázek 5: Komplement s antigenem [4]

V umělém imunitním systému si můžeme komplement a antigen představit jako binární řetězce určité délky. Zde probíhá porovnávání na základě podobnosti binárního řetězce komplementu s řetězcem antigenu. Aby byla obrana imunitního systému účinná, je zapotřebí, aby byly oba řetězce shodné. Řetězce však mohou být velmi dlouhé a šance, že budou 2 řetězce 100 % shodné, je poměrně malá. Z tohoto důvodu se trvá pouze na většinové podobnosti. Existují metody, které dokážou tuto shodu jednoduše a účinně určit. Mezi neznámější existující metody patří Euklidova vzdálenost, Manhattanská vzdálenost a Hammingova vzdálenost [4][13].

3.1.2.1.1 Euklidova vzdálenost

Pro výpočet Euklidovské vzdálenosti se používají následující vzorec:

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

kde x a y jsou dva jedinci se stejnou délkou řetězce. Euklidovu metodu použijeme například tam, kde je zapotřebí zjistit vzdálenost dvou vektorů v určité rovině. Příklad:

$$x = (3, 5)$$

$$y = (7, 2)$$

Při použití vzorce dostaneme následující výpočet:

$$d = \sqrt{(3 - 7)^2 + (5 - 2)^2} = 5$$

Výsledek Euklidovy vzdálenosti pro vektory x a y je tedy 5.

3.1.2.1.2 Manhattanská vzdálenost

Manhattanská vzdálenost byla při svém vytvoření inspirována pravoúhlou uliční sítí na ostrově Manhattan v New Yorku [31]. Tedy pokud řidič automobilu chce dorazit z místa A do místa B, vzdálenost mezi těmito dvěma body je stejná, ať už řidič zvolí jakoukoliv trasu směřující k bodu B. Pro takový výpočet je použit následující vzorec:

$$d = \sum_{i=1}^n |x_i - y_i|^2.$$

I zde prvky x a y znamenají dva body ve stejné rovině. Pro jednoduchost použijeme stejný příklad jako u předcházející metody:

$$x = (3, 5)$$

$$y = (7, 2)$$

Po dosazení hodnot do vzorce bude výpočet vypadat následovně:

$$d = |3 - 7| + |5 - 2| = 7$$

Výsledek Manhattanské vzdálenosti pro tento konkrétní příklad je 7.

3.1.2.1.3 Hammingova vzdálenost

V rámci této diplomové práce budeme využívat Hammingovu metodu, která je nejznámější a nejvyužívanější. Hammingova metoda počítá, kolik hodnot ve dvou řetězcích stejné délky je rozdílných nebo kolik změn je třeba provést, aby oba řetězce byly

identické. Pro větší názornost si tuto metodu můžeme ukázat na následujícím příkladu s binárními řetězci:

$$\begin{aligned}x &= (1, 0, 1, 1, 1, 0, 0, 1, 0) \\y &= (1, 1, 1, 0, 1, 0, 1, 1, 0) \\d &= (0, 1, 0, 1, 0, 0, 1, 0, 0) = 3\end{aligned}$$

Výsledek Hammingovy vzdálenosti je 3, tedy přesně 3 prvky v řetězci je zapotřebí pozměnit (lhostejno ve kterém), aby byly oba identické. Porovnání těchto řetězců můžeme provést metodou výlučného logického součtu (tzv. logická funkce XOR) [42].

Hammingova vzdálenost se dnes používá převážně v tzv. samoopravných kódech [40], ty se snaží nalézt a odstranit chyby vzniklé při přenosu dat v síti. Euklidovy a Manhattanové vzdálenosti se využívá v algoritmech shlukové analýzy dat, jež se snaží vyhledávat data a shlukovat je na základě podobnosti určitých vlastností [41][45].

3.1.2.2 Algoritmy selekce

V této kapitole budou popsány základní selekční algoritmy imunitního systému. Každá ze selekcí má specifické chování na dvou rozdílných místech našeho systému. Mezi selekční algoritmy patří pozitivní selekce, negativní selekce a klonální selekce.

3.1.2.2.1 Pozitivní selekce

V biologickém imunitním systému se proces pozitivní selekce používá pro odstranění vlastních poškozených a neúčinných buněk, které již neplní svou funkci. Veškeré buňky, jež projdou procesem pozitivní selekce, jsou ušetřeny (tj. zůstávají v organismu). V opačném případě jsou buňky odstraněny [4][20].

V umělém imunitním systému se tato selekce stará o odstranění prvků z množiny řešení (P), jež nedokážou rozpoznat vlastní prvky v množině S. Tento druh selekce tedy používáme v systému, kde je zapotřebí, aby množina P (tj. množina všech řešení) obsahovala pouze komplementy (detektory) rozpoznávající striktně jen vlastní prvky.

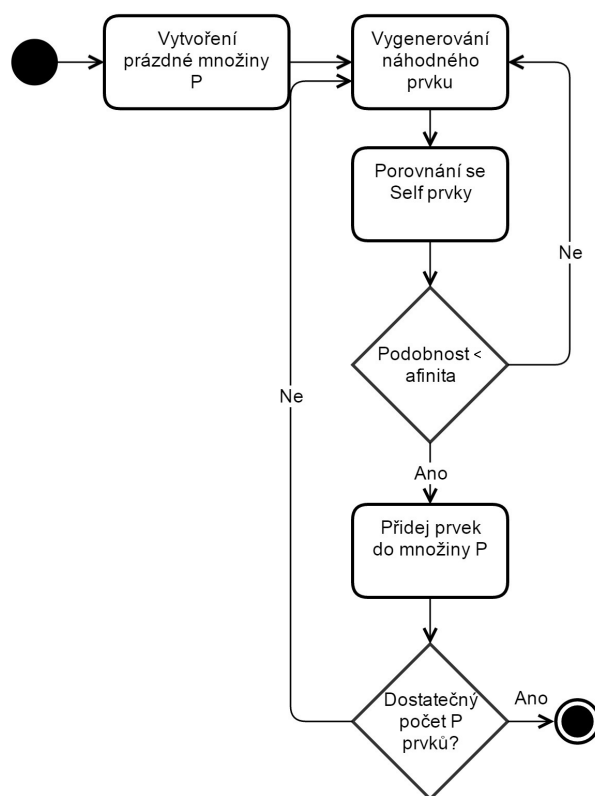
Selekce se uplatňuje na dvou místech algoritmu. Při vytvoření prvků řešení (množina P) a při filtraci nových prvků v systému.

Kroky k vytvoření množiny řešení:

1. **Vytvoření prázdné množiny řešení a nastavení hranice afinity.** V množině řešení (P) budou ty prvky, které dokážou rozeznat alespoň jeden prvek z množiny Self. Afinita stanovuje minimální hodnotu podobnosti, jíž musí prvek dosahovat, aby jej nějaký prvek z množiny P rozeznal jako vlastní (Self).

- Naplnění množiny řešení.** Zcela náhodně je vygenerován prvek řešení se všemi parametry. Tento prvek se porovná se všemi vlastními prvky (množina S) – vypočtená hodnota podobnosti (Hammingova vzdálenost) se porovná s afinitou a pokud je menší, je prvek přidán do množiny P , algoritmus pokračuje krokem 4. Jinak je náhodně vygenerovaný prvek zahozen a algoritmus se vrací ke kroku 2.
- Kontrola míry naplnění množiny P .** Zde dochází ke kontrole, zda již existuje v množině řešení dostatečný počet prvků. Pokud ne, algoritmus se opakuje od kroku 2. Je-li počet prvků dostačující, dochází k ukončení první části pozitivní selekce.

První část pozitivní selekce vystihuje níže uvedený diagram:

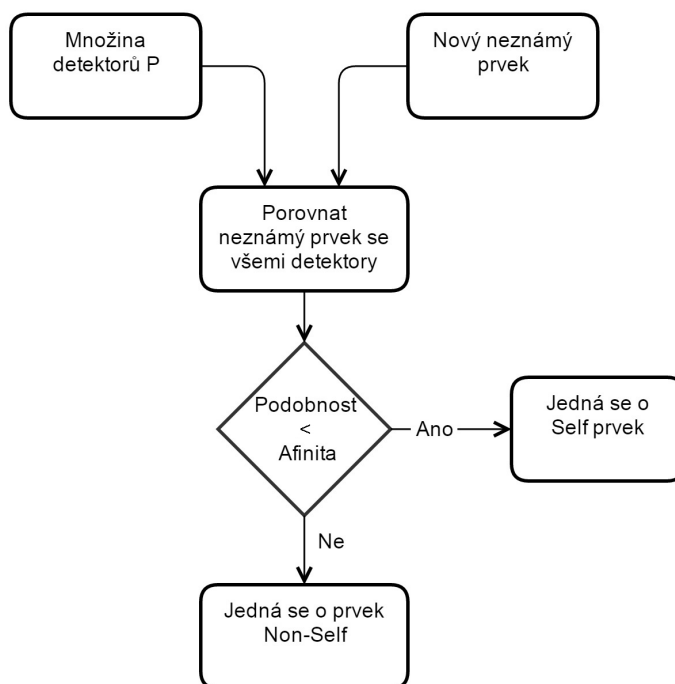


Obrázek 6: Naplnění množiny P , pozitivní selekce

Pokud za běhu programu dojde k nalezení nového neznámého prvku v systému, je třeba okamžitě zasáhnout a zjistit, zda se jedná o prvek vlastní či cizí. Níže je popsána kontrola a případná filtrace neznámého prvku:

1. **Porovnání neznámého prvku s množinou P.** Jakmile je nalezen nový prvek, je porovnán se všemi stávajícími prvky z množiny P (viz předchozí část) a vypočítá se míra podobnosti.
2. **Vyhodnocení podobnosti.** Stejně jako v předešlé části se hodnota vypočítané podobnosti porovná s afinitou, pokud je tato hodnota menší, prvek je rozeznán jako vlastní (Self), může tedy zůstat v systému. Je-li hodnota stejná nebo větší, jedná se o nevlastní (Non-self) prvek a je zapotřebí proti němu adekvátně zakročit.

Druhou část vyjadřuje tento diagram:



Obrázek 7: Filtrace nového prvku, pozitivní selekce

3.1.2.2 Negativní selekce

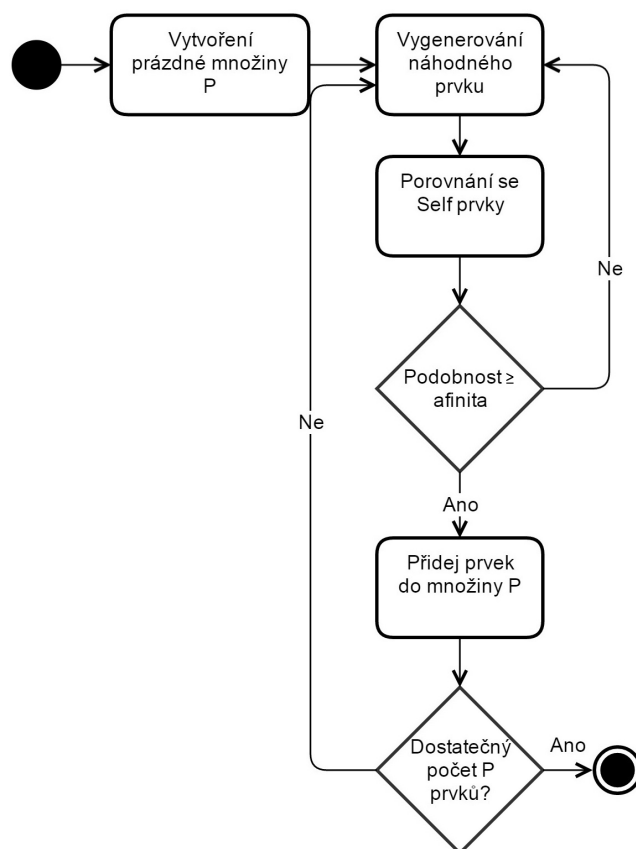
Biologický proces negativní selekce se používá k odstraňování nevhodných T lymfocytů. Některé lymfocyty se totiž mohou navázat i na vlastní buňky (podrobněji v sekci 2.1.2). Negativní selekce tedy odstraňuje ty T lymfocyty, jež rozeznávají (mimo jiné) také vlastní buňky. Tímto způsobem lze zabránit vzniku autoimunitních onemocnění.

Algoritmus negativní selekce v umělém imunitním systému tedy funguje na stejném principu jako v biologii – odstraňuje veškeré prvky z množiny řešení P , které rozeznávají vlastní buňky z množiny S . V množině P tedy zbudou jen ty buňky, jež jsou schopny se navázat pouze na buňky cizí.

První část algoritmu negativní selekce:

1. **Vytvoření prázdné množiny řešení a nastavení hranice afinity.** Je inicializována množina řešení P , ta bude obsahovat pouze ty buňky, jež nerozeznají ani jednu z buněk množiny S . Afinita se nastaví na předem zvolenou hodnotu.
2. **Naplnění množiny řešení.** Je vytvořen potenciální prvek množiny řešení a vypočítá se jeho podobnost (Hammingova vzdálenost), ta se poté použije pro srovnání s afinitou všech prvků množiny S . Je-li míra podobnosti stejná nebo větší než daná afinita (musí platit pro všechny prvky množiny S), je prvek přidán do množiny řešení. Pokud je však míra podobnosti menší, vygenerovaný prvek se zahodí a algoritmus zopakuje tento krok.
3. **Kontrola míry naplnění množiny P .** Pokud je již v množině P dostatečný počet prvků, algoritmus končí. V opačném případě se vrací ke kroku 2.

Základní diagram negativní selekce:

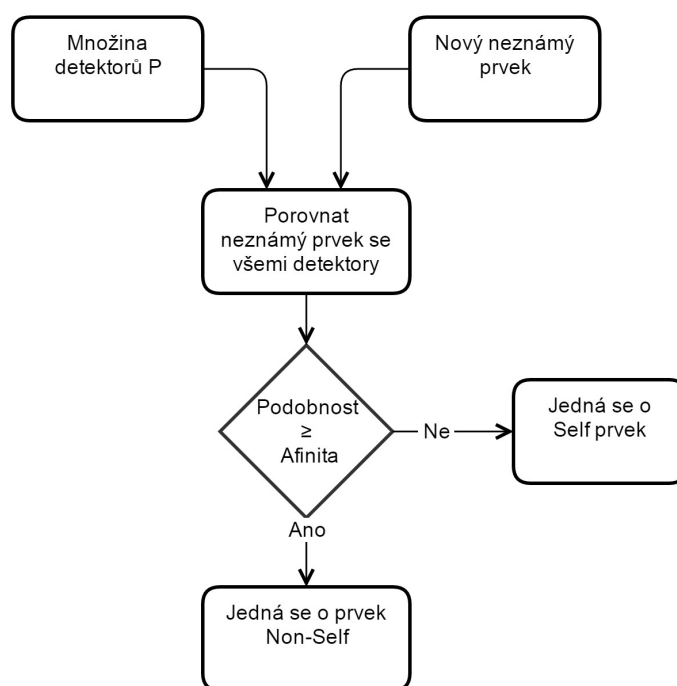


Obrázek 8: Naplnění množiny P, negativní selekce

Stejně jako pozitivní selekce je i selekce negativní složena ze dvou částí. Níže rozebraná druhá část se opět zabývá kontrolováním nových buněk v systému.

1. **Porovnání neznámého prvku s množinou P.** Dojde-li k detekci nového prvku, je tento porovnán se všemi prvky z množiny řešení P. Výsledkem je míra podobnosti.
2. **Vyhodnocení podobnosti.** Míra podobnosti vypočítaná v předchozím kroku se porovná s definovanou afinitou. Je-li větší nebo rovna hodnotě afinity, nově nalezený prvek je rozeznán jako cizí a spustí se adekvátní reakce umělého imunitního systému. V opačném případě se jedná o vlastní prvek.

Následuje diagram druhé části negativní selekce kontrolující nově nalezené prvky:



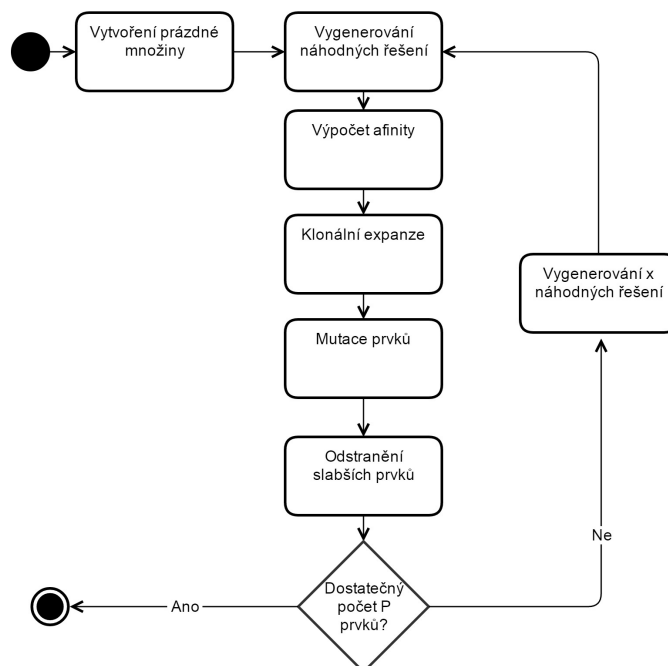
Obrázek 9: Filtrace nového prvku, negativní selekce

3.1.2.2.3 Klonální selekce

Klonální selekce se v biologických imunitních systémech objevuje v okamžiku, kdy dojde navázání T a B lymfocytů na patogen. Navázané buňky zvýší svůj metabolismus a začnou se velice rychle množit. Na jejich povrchu se vytvoří receptory (komplementy) schopné se navázat na antigen patogenu. Nově vzniklé lymfocyty se poté navážou na další patogeny, přičemž se rovněž množí. Celý proces se opakuje, dokud nejsou všechny patogeny zahubeny.

Následující kroky popisují, jak správně postupovat při vytvoření prvků pro množinu řešení P tak, aby byly úspěšné a efektivní při vypořádávání s cizími buňkami. Do množiny řešení P jsou ukládány pouze ty prvky, které dokážou rozlišit nějaký patogen/-patogeny nejlépe. Prvky, které nemají tak dobrou schopnost rozpoznat patogen jako jiné prvky z množiny, jsou z této množiny odstraněny. Díky tomu, množina řešení P obsahuje pouze ty nejlepší a nejpřesnější buňky. Pro větší přehlednost je klonální selekce popsána v bodech:

1. **Vytvoření prázdné množiny řešení P.** Základním krokem celého algoritmu je vytvoření prázdné množiny řešení P, která bude obsahovat pouze ty prvky, které dokážou eliminovat nějaký patogen.
2. **Vygenerování řešení.** V tomto kroku je vygenerován určitý počet prvků, které jsou následně uloženy do naší množiny P.
3. **Výpočet afinity.** Pro všechny prvky z množiny P je vypočítána afinita.
4. **Klonální expanze.** Podle velikosti afinity jsou prvky klonovány. Čím větší je hodnota afinity, tím více klonů bude vytvořeno.
5. **Mutace prvků.** Následuje mutace veškerých prvků dle afinity. Prvky, jež mají hodnotu afinity nižší, podstoupí větší mutaci než ty, které mají hodnotu afinity větší.
6. **Odstranění slabších prvků.** Z množiny řešení P jsou odstraněny prvky, které mají nízkou afinitu.
7. **Kontrola míry naplnění množiny P.** Nyní je kontrolováno, zdali je v množině P dostatečný počet prvků. Pokud ano, algoritmus klonální selekce končí, jinak je vygenerován určitý počet nových potenciačních řešení a následně se algoritmus opakuje od kroku 3.



Obrázek 10: Klonální selekce

Jakmile je množina P naplněna potřebnými řešeními jako u pozitivní nebo negativní selekce, je na řadě druhá část algoritmu selekce. Tato druhá část je stejná jako u negativní selekce, proto zde již nebude popsána.

3.1.3 Využití umělého imunitního systému

Umělé imunitní systémy mají obrovské pole působnosti v reálném světě. Mezi nejznámější typické oblasti, kde se umělé imunitní systémy využívají, patří počítačová bezpečnost, detekce anomálií, těžení dat (data mining), detekce vzorů, optimalizace, robotika, těžení webu a mnoho dalších. V této kapitole budou popsány některé z těchto oblastí. Informace k této kapitole byly čerpány z [4], [20], [30], [31] a [32].

3.1.3.1 Počítačová bezpečnost

Ochrana počítačů před útoky virů a hackery (programátoři, kteří se snaží nabourat do cizích počítačů) je v dnešní době jedním z nejproblematictějších úkolů počítačových specialistů. V biologii, se patogen snaží vniknout do našeho těla, kde se chtějí množit a

škodit. Stejně tak fungují i viry, které se pokoušejí vniknout do našich sítí a počítačů. Proto se v této oblasti stále více využívají aplikace založené na umělém imunitním systému.

3.1.3.1.1 Síťová bezpečnost

Existuje řada útoků zaměřujících se proti počítačovým sítím. Ty se snaží nalézt slabiny v operačních systémech nebo v aplikacích. Jakmile dojde k infekci počítače, je zapotřebí tuto hrozbu detekovat a adekvátně zakročit. J. Kim a P. Bentley navrhli ve své práci [31] několik vlastností umělého imunitního systému tak, aby byl co neúčinnější v rozpoznávání počítačových útoků. Tyto vlastnosti byly rozděleny do tří bodů:

1. Distribuovanost detekční sítě – tedy propojení bezpečnostní sítě na více nezávislých uzlů.
2. Samoorganizovanost zabezpečená negativní selekcí spolu s evolucí množiny genů, jež jsou využívány pro tvorbu detektorů.
3. Nenáročnost na systémové prostředky.

3.1.3.1.2 Detekce virů

Každý počítač (nebo jiné „chytré“ zařízení) by měl být chráněn antivirovým programem. Ten má za úkol detekovat a následně i eliminovat hrozby. Antivirové programy využívají také logiku biologického imunitního systému. Prohledávají operační systém v zařízení a porovnávají různé nálezy se svou databází virů, kde hledají určitou podobnost. Bohužel, viry se (stejně tak i v biologii patogeny) vyvíjejí a mutují. Je tedy zapotřebí databázi aktualizovat, obměňovat a přidávat další detektory, které dokážou viry správně rozpoznat. Analogií aktualizace databáze je v biologii očkování. Autoři článku [46] J. O. Kephart, G. B. Sorkin, M. Swimmer a S.R. White popsali myšlenku, jak vytvářet komplementy pro doposud nezmapované viry. Díky této metodě jsou komplementy schopny zajistit otisk viru a zapamatovat si jej pro budoucí použití.

3.1.3.2 Robotika

N. Mitsumoto, T. Fukada a T. Idogaki přišli jako první s nápadem využití znalostí imunitního systému v oblasti robotiky [32]. Vytvořili populaci robotů, kteří se chovali samostatně a organizovaně při hledání jídla bez žádného globálního kontrolního mechanismu. Celá ideologie robotiky využívá celkem tři vlastnosti biologického imunitního systému. Roboti v tomto systému zastupují B lymfocyty. Každý robot má konkrétní strategii při hledání jídla. Druhou vlastností je imunitní síť, která umožňuje interakci mezi

roboty. Třetí a poslední využití je kalkulace B buněk – čím více B buněk (robotů) je v systému k dispozici, tím lepší strategie je použita.

3.1.3.3 Optimalizace

Cílem optimalizace je pozměnit a tím vylepšit existující systémy tak, aby byly efektivnější při řešení problémů. Těch, při jejichž řešení se ideologie imunitního systému využívá, existuje celá řada. Jmenujme například problém obchodního cestujícího, ClonalG, doporučení filmů nebo například i v evolučních algoritmech.

1. **Problém obchodního cestujícího** - představuje výpočetně velmi náročný úkol. Rychlost výpočtu závisí zejména na velikosti množiny měst, jež jsou navzájem propojena cestami. Cílem je nalezení co nejkratší trasy mezi všemi městy tak, aby obchodník prošel všemi městy právě jednou s tím, že obchodník musí skončit svou cestu v počátečním městě. V imunitním systému jsou města reprezentována antigeny a trasy mezi nimi komplementy. Program imunitního systému poté už jen vyhledá nejlepší možné řešení. Touto možností využití se poprvé zabývali a implementovali S. Endoh, N. Toma a K. Yamada [33]. Dnes se s tímto problémem setkáváme v navigacích, online mapách atd.
2. **Doporučení filmů** – S. Cayzer [34] vyvinul systém doporučení filmů uživatelům. Každý uživatel má svůj účet, kde může doplňovat názvy filmů, které již zhlédl a následně je i ohodnotit podle svého uvážení. Pro daného uživatele (antigen) chceme zobrazit filmy, které by se mu mohly líbit. Ostatní uživatelé (komplementy) rozpoznají antigen daného uživatele, pokud zhlédli stejný film a zároveň jej ohodnotili podobně. Díky pozitivní selekci se poté vyberou ty filmy, které dosud daný uživatel nezhlédl.
3. **Evoluční algoritmy** – Díky optimalizaci evolučních algoritmů imunitním systémem je algoritmus mnohem rychlejší a efektivnější. Stručný popis evolučních algoritmů je k nalezení níže.

4 Cíl práce

Cílem diplomové práce je naimplementovat optimalizační algoritmus binární diferenciální evoluce spolu s umělým imunitním systémem. Práce vychází z publikace X. He a L. Han [23], kteří se touto problematikou zabývali, a popisuje základní funkcionalitu binární diferenciální evoluce rozšířenou o umělý imunitní systém, který má zefektivnit a urychlit proces vývoje vyhledání řešení.

Dalším cílem je rozšířit tuto implementaci o pozitivní selekci, další typy diferenciální evoluce a nakonec provést experimenty, díky nimž zjistíme, které z řešení je nejefektivnější. Logiku, funkci a princip umělého imunitního systému jsme si již nastudovali a pochopili. Nyní je zapotřebí nastudovat diferenciální evoluci, jež je jedním z typů evolučních algoritmů. Proto si nejprve osvětlíme základy evolučních algoritmů.

4.1 Evoluční algoritmy

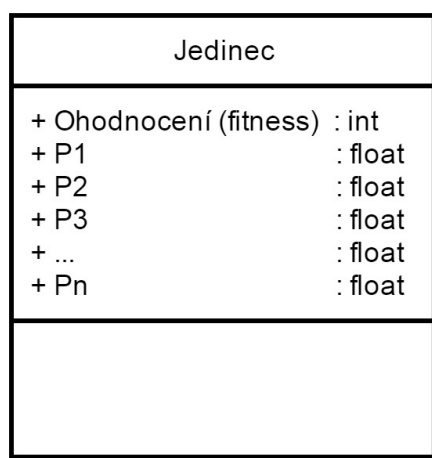
Evoluční algoritmy jsou inspirovány biologickou evolucí, za jejíhož zakladatele se považuje Charles Darwin [3]. Darwin dovedl vysvětlit a dokázat možný vývoj druhů přírodním výběrem. Ti jedinci, kteří jsou lépe přizpůsobeni svému prostředí, přežívají a mají možnost reprodukce. V rámci několika generací se populace vyvíjí a tím vylepšuje své vlastnosti.

Evoluční algoritmy jsou druhem stochastických optimalizačních metod nazývané také jako metaheuristiky, ty se snaží vyhledat optimální řešení v daném prostoru. Tento proces je časově a výpočetně velmi náročný [30].

4.1.1 Základní pojmy a princip evolučních algoritmů

Základním rysem evolučních algoritmů je vyhledávání řešení náhodnými změnami. Pokud je nově vzniklé řešení lepší a výhodnější než to staré, tak jej nahrazuje. Evoluční algoritmy pracují s populací jedinců. Každý jedinec je tvořen parametry a tzv. fitness hodnotou, jež určuje, jak je daný jedinec, představující řešení daného problému, dobrý v rámci populace.

Jakékoliv řešení určitého problému je reprezentováno jedincem (také nazývaný fenotypem) populace. Jedince si můžeme představit jako instanci třídy *Jedinec* (obrázek 11), kde každý jedinec obsahuje sadu parametrů P_1, P_2, \dots, P_n .



Obrázek 11: Reprezentace třídy Jedince v třídním diagramu

Jedinci v populaci mohou být reprezentováni následovně:

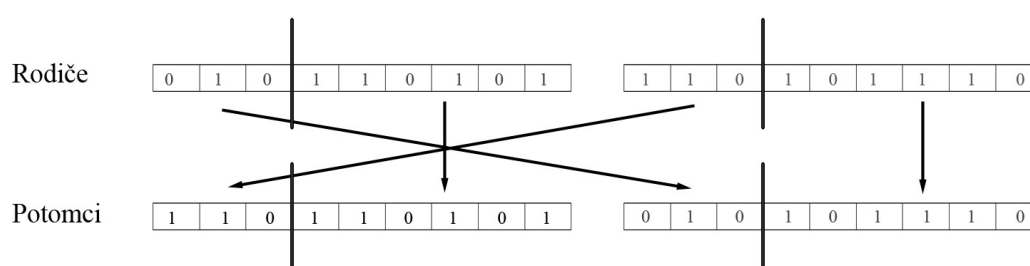
1. **Binární reprezentace** - Jedná se o historicky nejstarší a nejjednodušší reprezentaci zpopularizovanou genetickými algoritmy. Jedinci jsou řetězce, jež se skládají pouze z 0 a 1 (false a true).
2. **Celočíselná reprezentace** – Parametry jedince obsahují libovolné znaky z definované množiny. Množina může a nemusí být číselná. Mezi prvky nemusí existovat žádný vztah.
3. **Reálná reprezentace** – Tato reprezentace byla preferovaná v evolučních algoritmech, kde každý parametr jedince obsahuje reálné číslo.
4. **Pořadí** – Existuje množina objektů, které je třeba setřídít a každý prvek v této reprezentaci může být použit právě jednou. Na tomto základě jsou vytvářeny permutace jedinců.
5. **Struktury** – Pod pojmem struktury si můžeme představit matici nebo stromovou strukturu. U matice je prohledávaný prostor pevně daný. Naopak u stromu je prohledávání proměnlivé, v tomto případě se preferuje genetické programování. [24]

Jedinci se v populaci v průběhu času (generací) dále vyvíjejí a ti s lepší fitness hodnotou nahrazují ty, kteří mají hodnotu fitness horší. Po uplynutí určitého časového období

zbydou v populaci jedinci, kteří představují ta nejlepší řešení daného problému a někteří z nich mohou dosáhnout hodnot blízcích se optimálnímu řešení.

Při generování nové populace jsou důležité 2 základní kroky:

1. **Křížení** – Jedná se o reprodukci dvou (nebo více) existujících jedinců, kteří jsou vybráni z množiny populace. Podle druhu evolučního algoritmu je vytvořen jeden nový jedinec nebo více, ti nesou chromozomy obou jedinců (obrázek č. 12).



Obrázek 12: Křížení

2. **Mutace** – po křížení přichází na řadu mutace. Jedná se o změnu náhodně vybraných parametrů jedince. U binárních jedinců se mutací mění 0 na 1 a naopak. Díky tomu je možné nalézt lepšího jedince, který vyhovuje podmínkám.

Pozn.: Křížení a mutace se mohou lišit v různých typech evolučních algoritmů.

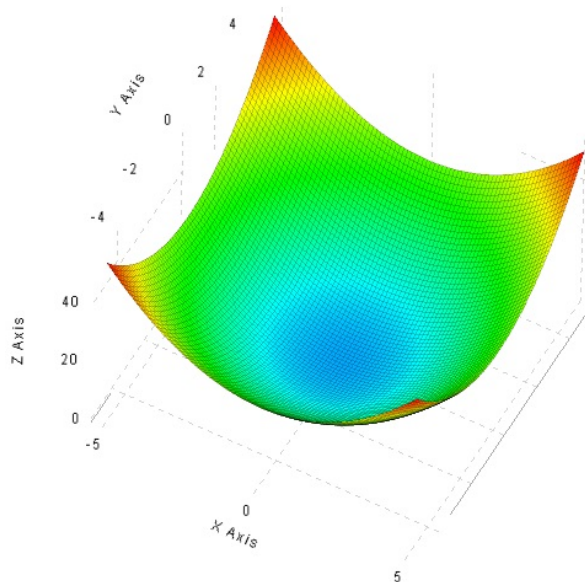
4.1.2 Účelová funkce

Účelová funkce je jedna z nejdůležitějších a nejsložitějších částí optimalizačních procesů. Její nesprávná definice může vést k výsledkům, které neodpovídají povaze problému. Obvykle pomocí účelové funkce zjišťujeme minimální nebo maximální pozici v určitém prostoru (dle velikosti rozměru plochy). Pokud jedinec populace má 5 parametrů, hledá se řešení v 5-ti rozměrném prostoru.

Doposud neexistuje návod nebo stručné instrukce, které by popisovaly, jak správně danou účelovou funkci sestavit. Existují však principy, které pomáhají správnému vytvoření účelové funkce [24][25]. V citovaných pracích můžeme nalézt podrobnější popis této problematiky spolu s příklady použití účelové funkce.

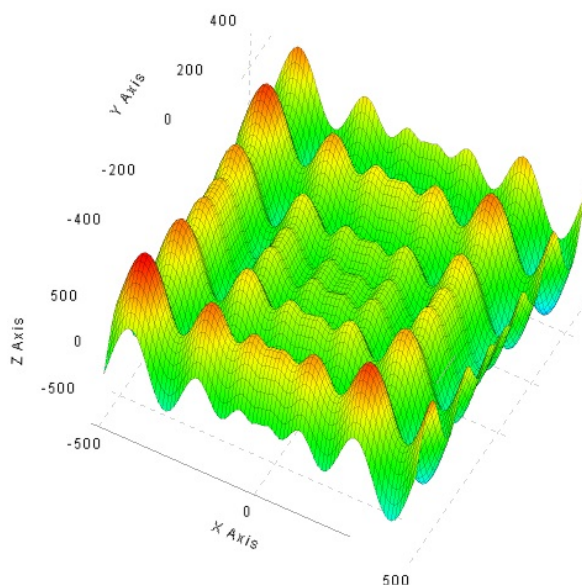
Podle počtu extrémů se účelová funkce dělí:

1. **Unimodální účelová funkce** – Jedná se funkcí, která má pouze jeden extrém (obrázek č. 13).



Obrázek 13: De Jongova funkce

2. **Multimodální účelová funkce** – Naopak multimodální účelová funkce má extrémů mnohem víc, proto je zde vyhledávání řešení mnohem komplikovanější (obrázek 14).



Obrázek 14: Schwefelova funkce

4.2 Diferenciální evoluce

Jedním z nejznámějších evolučních algoritmů, a poměrně novým, je diferenciální evoluce, jež byla vyvinuta v letech 1995 autory K. Pricem a R. Stormem [35]. Diferenciální evoluce vznikla vývojem a změnami algoritmu genetického žhání, které slouží také pro řešení složitějších problémů. Úpravami algoritmu genetického žhání vznikla tzv. diferenciální mutace, ta generuje zkušební řešení přičtením dvou náhodně vybraných jedinců ke třetímu, také náhodně vybranému. K tomuto výsledku byla poté přidána metoda selekce z genetického žhání. Posléze se ukázalo, že některé použité metody z genetického žhání jsou nadbytečné, proto byly vypuštěny.

Pokud se jedná o základní typ diferenciální evoluce DE/rand/1/bin (druh výpočtu mutačního vektoru, bude vysvětleno v kapitole 4.2.5), velmi se podobá genetickým algoritmům využívajícím podobné metody. Jednou z těchto metod je generování potomků. Diferenciální evoluce vytváří nového potomka pro budoucí populaci ze čtyř jedinců z aktuální populace, oproti tomu genetický algoritmus používá jedince pouze dva.

4.2.1 Parametry diferenciální evoluce

Existuje řada parametrů, které diferenciální evoluce využívá. Nebyli bychom schopni tento algoritmus správně naimplementovat, aniž bychom s nimi byli seznámeni.

1. **NP[10D, 100D]** – Jedná se o maximální počet jedinců v populaci, kde D značí rozměr jedince. Pro správné fungování diferenciální evoluce je zapotřebí mít vždy minimálně 4 jedince, potřebné pro generování nových jedinců. Čím větší počet jedinců bude v populaci definován, tím bude algoritmus časově náročnější.
2. **G** - Udává počet generací.
3. **CR[0,1]** - Parametr udávající práh křížení. Pokud je hodnota nastavena na 0, nebude provedena mutace na nově vytvořeném jedinci, ten bude přesnou kopií svého aktuálního rodiče a dojde tak k úplnému zastavení vývoje populace. Pokud tedy nastavíme CR na 1, bude nový jedinec pouhou kombinací tří jedinců z populace. Průběh diferenciální evoluce by tedy vypadal jako náhodné hledání řešení problému.
4. **F[0,2]** - F značí parametr mutační konstanty, poslední řídicí parametr diferenciální evoluce.

Doporučené hodnoty pro parametry diferenciální evoluce (převzato z [24]):

Parametr	Interval	Doporučená hodnota
NP	[10D, 100D]	10
CR	[0,1]	0,8 – 0,9
F	F	0,3 – 0,9

Tabulka 1: Doporučené hodnoty

4.2.2 Mutace, křížení a selekce

Mutace a křížení jsou dva velmi podstatné kroky diferenciální evoluce (kapitola 4.1.1), avšak každý z evolučních algoritmů provádí mutaci a křížení jiným způsobem. Z tohoto důvodu je zapotřebí si je vysvětlit dříve, než bude vysvětlen princip fungování diferenciální evoluce.

4.2.2.1 Mutace

Základním krokem vývoje populace je tzv. mutace. Díky mutaci dokáže diferenciální evoluce DE/rand/1/bin vygenerovat nového jedince, který je vytvořen celkem ze čtyř jedinců z aktuální generace (genetické algoritmy potřebují jedince pouze 2). Tři z jedinců jsou vybrány z populace náhodně (r_1, r_2, r_3). Čtvrtý jedinec je aktuální jedinec cyklu (x_i),

ten je však využit až u procesu křížení. Všichni jedinci musí splňovat následující podmínku:

Mutace

$$x_i \neq r_1 \neq r_2 \neq r_3$$

Mutaci můžeme popsat následujícím vzorcem (pro základní druh DE/rand/1/bin):

$$v_j = x_{r_1,j}^G + F \cdot (x_{r_2,j}^G - x_{r_3,j}^G)$$

Ten značí, že první dva náhodně vybraní jedinci ze stejné populace ($x_{r_1,j}^G, x_{r_2,j}^G$) se od sebe odečtou a jsou následně vynásobeni mutační konstantou F . Výsledný vektor je pak přičten k třetímu náhodně vybranému jedinci ($x_{r_3,j}^G$). Výsledkem mutace je vytvořen tzv. „mutační vektor v_j “ nebo také „šumový vektor“.

4.2.2.2 Křížení

V diferenciální evoluci nastává proces křížení až po procesu mutace (například u genetických algoritmů je tomu naopak). Cílem křížení u diferenciální evoluce je z doposud nepoužitého aktuálního jedince (x_i) a vzniklého mutačního vektoru vytvořit tzv. zkušební vektor (trial vector). V tomto procesu stojí proti sobě pokaždé stejné parametry obou prvků (první parametr aktuálního jedince a první parametr mutačního vektoru, atd.). Pro každý parametr je generováno náhodné číslo z intervalu $[0,1]$ a pokud je hodnota vygenerovaného čísla menší než hodnota CR , zkopíruje se do příslušného parametru zkušebnímu vektoru parametr šumového vektoru. V případě, že je hodnota větší než CR , vybere se parametr čtvrtého jedince.

4.2.2.3 Selektce

Nově vzniklý jedinec je následně ohodnocen (spočítá se hodnota fitness). Tento jedinec pak bojuje o místo v nové populaci s aktuálním jedincem. Do nové populace se umístí jedinec s lepší fitness hodnotou (v případě hledání globálního maxima bude lepší vyšší hodnota fitness).

4.2.3 Princip

Diferenciální evoluce pracuje s populací jedinců. Cílem algoritmu je vygenerovat populaci, která bude mít co nejlepší hodnotu fitness funkce. Pro dosažení tohoto cíle je zapotřebí provést následující kroky:

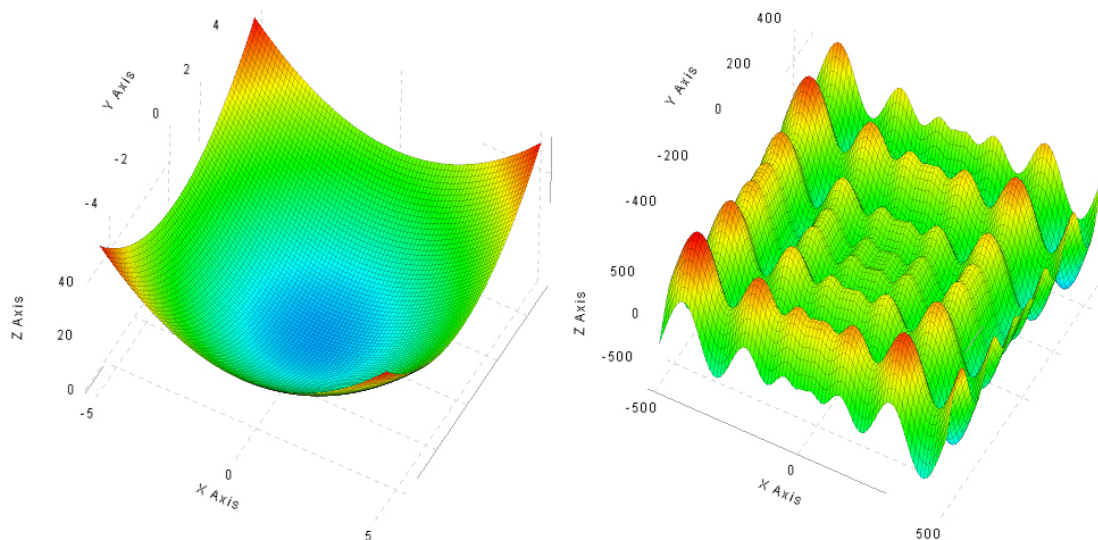
1. **Nastavení řídicích parametrů.** V prvním kroku je potřeba přesně stanovit parametry, které jsou popsány v části 4.2.1. Nastavení těchto parametrů je nezbytné pro správný chod diferenciální evoluce.
2. **Vygenerování populace.** Druhým krokem je vygenerování jedinců do populace podle parametru NP. Tito jedinci musí být vytvořeni dle předem definovaného vzoru (specimen). Každý z těchto jedinců musí mít kromě parametrů i hodnotu fitness.
3. **Cyklus generace.** Tento cyklus zajišťuje výpočet celé generace a zahrnuje i následující krok – Evoluční cyklus.
4. **Evoluční cyklus.** V evolučním cyklu se provádějí procedury podrobně rozebrané v sekci 4.2.2. Je prováděn tolikrát, kolik je v populaci jedinců. Na konci cyklu je stávající populace nahrazena novou (v každé iteraci je vytvořen jedinec pro novou populaci).
 - (a) **Mutace.**
 - (b) **Křížení.**
 - (c) **Selekce.**
5. **Ukončení cyklu generací.** Celý proces diferenciální evoluce končí tehdy, až je proveden uživatelem daný počet generačních cyklů (generací). K ukončení cyklu generací může dojít i tehdy, pokud algoritmus vyhodnotí, že není vytvořen nový unikátní jedinec.
6. **Vyhodnocení.** Výsledkem celé diferenciální evoluce je populace jedinců, kteří představují nejlepší nalezené řešení.

Celý princip fungování diferenciální evoluce může být popsán i diagramem:

4.2.4 Vlivy na průběh diferenciální evoluce

Je mnoho faktorů, které ovlivňují průběh evoluce populace jedinců. Jde zejména o parametry definované před samotným spuštěním algoritmu. Patří mezi ně:

1. **Počáteční parametry** – Nastavení parametrů z kapitoly 4.2.1 může velmi ovlivnit rychlost evoluce. Například nastavení příliš nízké hodnoty CR může mít za následek pomalou evoluci, jedinci se obměňují pomaleji a zpomaluje se tak celý algoritmus.
2. **Velikost populace** – Příliš malý počet jedinců v populaci může způsobit brzké skončení diferenciální evoluce, aniž by se mu podařilo nalézt globální extrém. Je proto zapotřebí brát v úvahu velikost populace vzhledem k výkonnosti počítače.
3. **Definice účelové funkce** – Pokud je špatně nadefinována účelová funkce, může dojít ke špatným rozhodnutím při výběru lepších jedinců, kteří postoupí do další generace. Může to mít tudíž za následek velmi pomalý průběh evoluce nebo se může evoluce zcela zastavit.
4. **Velikost prostoru** – Asi jeden z největších vlivů na algoritmus může mít prostor, ve kterém diferenciální evoluce probíhá. Například pokud evoluce bude probíhat na de Jongově funkci (obrázek 16 vlevo), bude mnohem jednodušší nalézt globální minimum. Oproti tomu na funkcích podobných Schwefelově funkci (obrázek 16 vpravo) je mnohem složitější globální extrémy nalézt. Evoluce bude stále nacházet nové a nové lepší jedince blížící se globálnímu extrému, avšak samotný globální extrém nalézt nemusí (díky množství lokálních extrémů, kde mohou jedinci „uvíznout“).



Obrázek 16: De Jongova a Schwefelova funkce

4.2.5 Druhy diferenciální evoluce

Existuje velké množství možných výpočtů diferenciální evoluce. Rozdíl mezi jednotlivými druhy je nejčastěji ve výpočtech mutačního vektoru. V předchozích kapitolách jsme si popsali jednu z možných variant výpočtu mutačních variant a to DE/rand/1/bin:

$$v_j = x_{r_1,j}^G + F \cdot (x_{r_2,j}^G - x_{r_3,j}^G)$$

Definice dalších variant výpočtu mutačního vektoru:

1. DE/best/1/exp

$$v_j = x_{best,j}^G + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G)$$

2. DE/rand-to-best/1/exp

$$v_j = x_{i,j}^G + \lambda \cdot (x_{best,j}^G - x_{i,j}^G) + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G)$$

3. DE/best/2/exp

$$v_j = x_{best,j}^G + F \cdot (x_{r_1,j}^G + x_{r_2,j}^G - x_{r_3,j}^G - x_{r_4,j}^G)$$

4. DE/rand/2/exp

$$v_j = x_{r_5,j}^G + F \cdot (x_{r_1,j}^G + x_{r_2,j}^G - x_{r_3,j}^G - x_{r_4,j}^G)$$

5. DE/best/1/bin

$$v_j = x_{best,j}^G + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G)$$

6. DE/rand/1/bin

$$v_j = x_{r_1,j}^G + F \cdot (x_{r_2,j}^G - x_{r_3,j}^G)$$

7. DE/rand-to-best/1/bin

$$v_j = x_{i,j}^G + \lambda \cdot (x_{best,j}^G - x_{i,j}^G) + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G)$$

8. DE/best/2/bin

$$v_j = x_{best,j}^G + F \cdot (x_{r_1,j}^G + x_{r_2,j}^G - x_{r_3,j}^G - x_{r_4,j}^G)$$

9. DE/rand/2/bin

$$v_j = x_{r_5,j}^G + F \cdot (x_{r_1,j}^G + x_{r_2,j}^G - x_{r_3,j}^G - x_{r_4,j}^G)$$

10. DE/current-to-best/1/bin

$$v_j = x_{i,j}^G + F \cdot (x_{best,j}^G - x_{i,j}^G) + F \cdot (x_{r_1,j}^G - x_{r_2,j}^G)$$

11. DE/current-to-rand/1/bin

$$v_j = x_{i,j}^G + F \cdot (x_{r_1,j}^G - x_{i,j}^G) + F \cdot (x_{r_2,j}^G - x_{r_3,j}^G)$$

5 Implementace a popis aplikace

V této části diplomové práce se již zaměříme na implementaci nabytých znalostí umělého imunitního systému a diferenciální evoluce. Jak již bylo uvedeno v některé z předešlých kapitol, budeme vycházet z části vědecké práce X. He, L. Hana [23].

5.1 Článek „A Novel Binary Differential Evolution Algorithm Based On Artificial Immune System“

Autoři v publikaci vysvětlují základní principy diferenciální evoluce. Dále pak ujasňují, že v binární variantě diferenciální evoluce se provádí výpočet mutačního vektoru s pomocí základních logických funkcí. Nejdůležitější částí je však kapitola zabývající se binární diferenciální evolucí, jež využívá principů negativní selekce umělého imunitního systému. Autoři však využili jen základního potenciálu negativní selekce – rozhodli se provést kontrolu mutačního vektoru jen na základě počtu jedniček v binárním řetězci. Tato práce rozšiřuje článek [23] zejména o poznatky z použití pozitivní selekce a dalších variant algoritmu diferenciální evoluce. V závěru práce jsou uvedeny výsledky experimentů, ty se soustředí ují na porovnání pozitivní a negativní selekce v již dříve popsáných variantách algoritmu diferenciální evoluce.

5.2 Třídy aplikace

Program je rozdělen do tří částí:

1. **Hlavní program a doplňky** – Zde se nachází třídy:
 - (a) **Program.cs** - Jedná se o třídu obsahující hlavní tok programu, tj. obecnou programovou funkcionalitu, která abstrahuje od implementačních detailů jednotlivých procedur diferenciální evoluce a imunitního systému.
 - (b) **Utilities.cs** - Obsahuje pouze pomocné metody pro převody různých typů.
2. **Binární diferenciální evoluce** - V této části se nachází 7 tříd:
 - (a) **Individual.cs** - Třída Individual reprezentuje jedince populace. Každý z jedinců obsahuje parametr Figure, řetězec jedniček a nul (true a false) a hodnotu Fitness, která vyjadřuje výsledek ohodnocení jedince. Čím je hodnota Fitness nižší, tím je jedinec lepší.

- (b) **AbstractDifferentialEvolution.cs** – Abstraktní třída sdružující společné vlastnosti tříd `DEBest1Bin.cs`, `DECurrentToBest1Bin.cs`, `DECurrentToRand1Bin.cs`, `DERand1Exp.cs`, `DERand2Bin.cs`, které obsahují rozdílnou metodiku pro výpočet mutačního vektoru.

3. Umělý imunitní systém - Třídy související s umělým imunitním systémem:

- (a) **Lymphocyte.cs** - Tato třída představuje prvky patřící do množiny řešení P . Díky těmto prvkům dokáže algoritmus určit, zdali je jedinec z populace vhodný pro další vývoj. Každý prvek obsahuje atribut `Antibody`, jenž stejně jako `Figure` u jedinců obsahuje řetězec jedniček a nul. Dále je zde `afinita`, která určuje minimální míru podobnosti – ta rozhodne, jestli je řešení podobné jedinci či nikoliv.
- (b) **AbstractSelection.cs** - Slouží jako abstraktní třída definující společnou funkcionalitu pro negativní (`NegativeSelection.cs`) a pozitivní (`PositiveSelection.cs`) selekci. V těchto třídách jsou implementovány samotné selekční metody.

5.3 Popis implementace

Základní principy implementace umělého imunitního systému i diferenciální evoluce již byly popsány. Nyní je zapotřebí provést implementaci diferenciální evoluce, která bude využívat principů imunitního systému tak, aby došlo k zefektivnění a urychlení hledání řešení (tj. v našem případě nalézt globální minimum v hledaném prostoru).

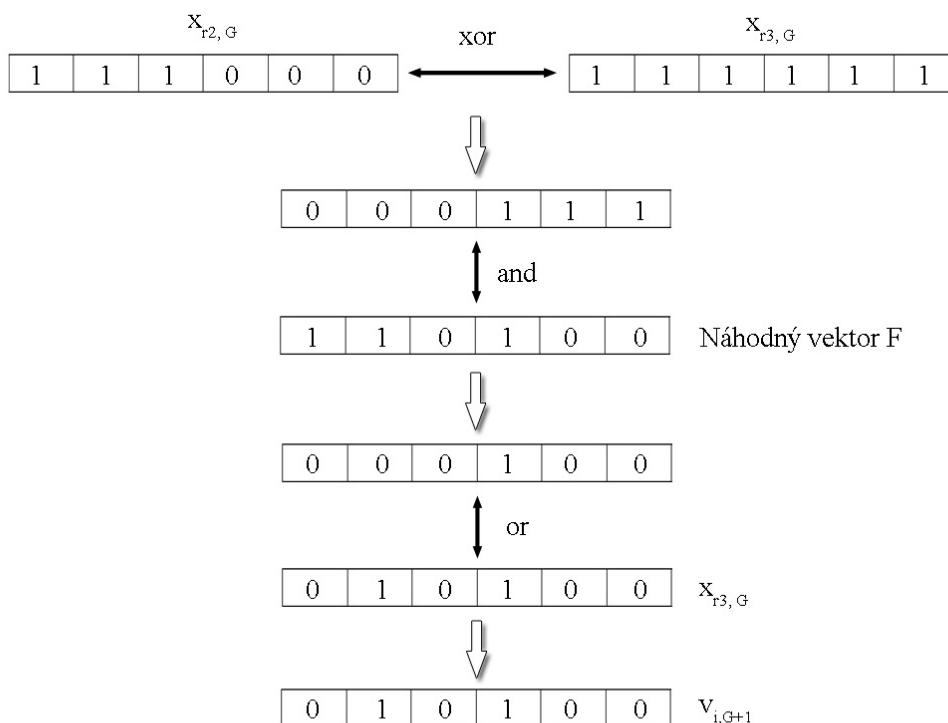
V našem kontextu můžeme označit jedince (`Individual`) první populace za prvky množiny S , tedy vlastní prvky. Na jejich základě proběhne, podle vybraného druhu selekce, vygenerování řady lymfocytů (`Lymphocytes`). Mutační vektory ($v_{i,G+1}$) budeme považovat za antigeny, které jsou vždy vypočítány dle použitého typu diferenciální evoluce. Pro výpočet binárních mutačních vektorů jsou použity logické operace, místo klasických matematických operací:

1. **Odečítání** – V případě odečítání dvou binárních vektorů (chromozomů) se používá logická operace XOR. Výsledkem XOR je další binární vektor, kde každá hodnota představuje rozdíl mezi použitými vektory. Pokud hodnota je 1, chromozomy se v této i -té hodnotě liší a naopak.
2. **Násobení** – Nástupcem násobení je operace AND. Pro AND je vždy výsledek 1, pokud oba vektory mají v i -tém prvku 1. Pokud je však jedna z hodnot 0, výsledek je 0.

3. **Sčítání** – Pokud chceme sčítat dva binární vektory, je zapotřebí použít logické operace OR. Výsledkem je vždy hodnota 1, pokud alespoň jeden z použitých vektorů obsahuje 1.

Pro větší názornost je níže ukázán výpočet mutačního vektoru (viz obrázek 17) pro základní typ diferenciální evoluce, DE/rand/1/bin, jehož vzorec je:

$$v_j = x_{r_1,j}^G + F \cdot (x_{r_2,j}^G - x_{r_3,j}^G)$$



Obrázek 17: Výpočet binárního mutačního vektoru [23]

Náhodný vektor F je náhodně vygenerovaný vektor, který nahrazuje v binární verzi diferenciální evoluce mutační konstantu F .

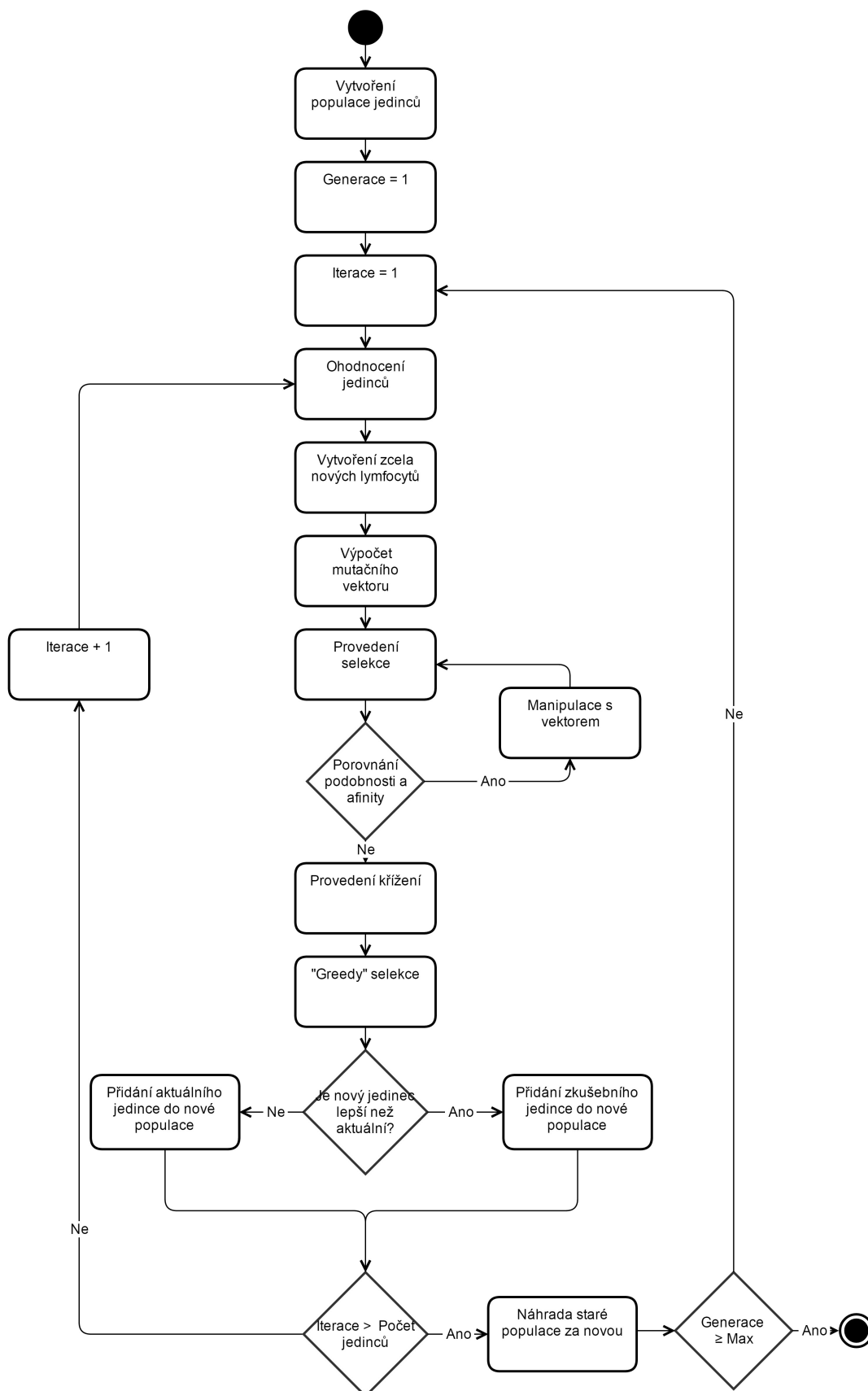
Celý algoritmus diferenciální evoluce využívající umělý imunitní systém můžeme popsat následujícími kroky:

1. **Inicializace parametrů algoritmu.** V prvním kroku je třeba vytvořit prázdné množiny populace a lymfocytů. Množina populace bude obsahovat různé jedince a množina lymfocytů bude obsahovat prvky řešení. Kromě množin je zapotřebí nastavit parametry diferenciální evoluce (viz 4.2.1) a nastavení afinity pro imunitní systém. Dále se musí vytvořit populace náhodných jedinců, se kterými se bude dále v běhu programu pracovat.
2. **Počátek generace a vytvoření nového jedince.** Zde se inicializuje výpočet nových jedinců, již se vyvíjejí z aktuální populace. Celkový počet cyklů pro výpočet nové populace závisí na velikosti množiny populace.
3. **Ohodnocení jedinců.** Zde se aktivuje algoritmus účelové funkce, která vypočítá a přiřadí všem jedincům jejich ohodnocení, tedy hodnotu, která reprezentuje řešení (čím menší hodnota, tím lepší řešení).
4. **Vytvoření zcela nových lymfocytů.** Na základě uživatelem vybrané selekce (pozitivní nebo negativní, viz 3.1.2.2) je provedena tvorba lymfocytů, ty jsou poté přiřazeny do množiny lymfocytů.
5. **Výpočet mutačního vektoru.** Výpočet mutačního vektoru je závislý na uživatelem vybraném druhu diferenciální evoluce (viz 4.2.5). Jak již název napovídá, jedná se o mutaci aktuálních jedinců (více informací 4.2.2).
6. **Provedení selekce.** Druhý krok selekce (první je načrtnut pod číslem 4 v tomto popisu), stejný princip jako v prvním kroku je použit i pro kontrolu mutačního vektoru (antigenu). Vektor je porovnán se všemi prvky množiny lymfocytů a je zjištěna míra podobnosti.
7. **Porovnání podobnosti a afinity.** Zde záleží na vybraném selekčním algoritmu, jak bude tato podobnost vyhodnocena.
 - (a) **Negativní selekce** - Pokud je míra podobnosti stejná nebo větší než afinita, bude proveden krok 8. V opačném případě algoritmus pokračuje s krokem 9.
 - (b) **Pozitivní selekce** – Zde dochází k opačnému chování než u negativní selekce. Tedy, pokud je míra podobnosti menší než afinita, algoritmus přejde na krok 8, jinak skočí na 9.
8. **Manipulace s vektorem.** Jakmile je vyvolána metoda manipulace s vektorem, je tento vektor pozměněn (viz obrázek 19). Nejprve jsou sečteny všechny 1 ve vektoru (parametr C). Pokud je jejich počet větší než zvolené maximum (V_{max}), je náhodně

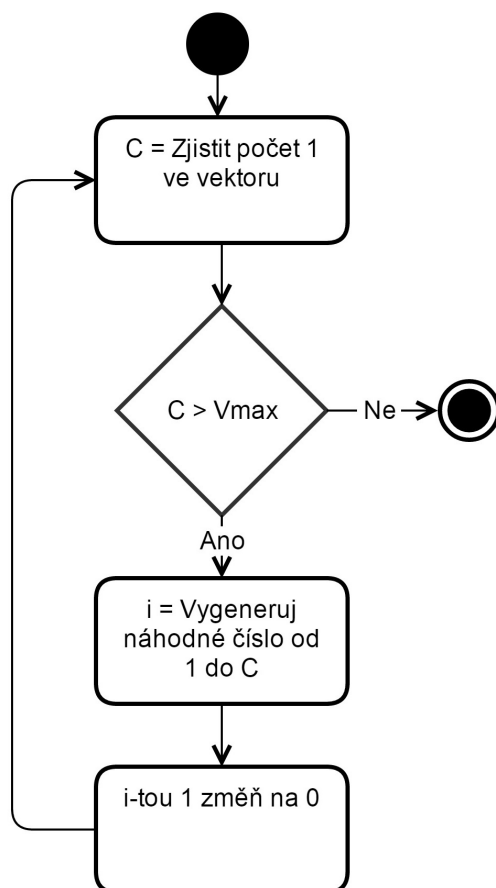
vygenerováno číslo I od 1 do C . Vygenerované číslo značí i -tou 1 vektoru, která bude změněna na 0. Celý cyklus manipulace s vektorem se opakuje do té doby, dokud není počet 1 menší než definované maximum (V_{max}).

9. **Provedení křížení.** Zde je provedeno křížení mutačního vektoru s aktuálním i -tým (i = iterace) jedincem. Více o křížení v kapitole 4.2.2.
10. **Výběr lepšího jedince a přidání do nové populace.** V tomto kroku dochází porovnání hodnot ohodnocení, tedy jestli má nově vygenerovaný jedinec lepší ohodnocení než aktuální (i -tý) jedinec. Lepší z nich postupuje do množiny nové populace, která v další generaci nahradí tu stávající. Tomuto kroku se také říká „Greedy“ selekce.
11. **Náhrada staré populace za novou.** Jakmile byl dokončen dostatečný počet iterací, dochází k odstranění staré populace a nahradí ji populace nová.
12. **Kontrola iterace.** Dochází ke kontrole počtu iterací v populaci. Pokud ještě nebylo dosaženo požadovaného počtu iterací, řídicí proměnná cyklu (i) je navýšena o jedničku a cyklus se provádí znovu (krok 5). Pokud byl již počet iterací dostačující, program přejde na krok 13.
13. **Kontrola generace.** V posledním kroku programu dochází ke kontrole, zdali byl proveden daný počet generací. Pokud ano, program končí. Jinak je k hodnotě proměnné řídicí generační cyklus přičtena jednička a program pokračuje od kroku 4.

Veškeré předchozí kroky mohou být pro lepší představu znázorněny v následujícím diagramu:



Obrázek 18: Diagram diferenciální evoluce využívající umělý imunitní systém



Obrázek 19: Manipulace s vektorem

6 Experimenty

Nyní se budeme zabývat experimenty nad implementovaným řešením. Budou provedeny experimenty jak pro oba druhy implementovaných selekcí, tak i pro pět druhů diferenciální evoluce. Pro účely testování budou využity testovací problémy [36][23].

Program optimalizované diferenciální evoluce byl naimplementován v Microsoft Visual Studio 2010 Ultimate v jazyce C# a byl použit notebook Acer s procesorem Intel Core i7-4712MQ a 8GB paměti RAM.

6.1 Testovací problémy

Jedná se o několik funkcí, přičemž každá z nich využívá jiných metod pro určení hodnoty fitness. V této kapitole budou funkce postupně popsány včetně principu jejich fungování. Následně budou použity pro testování implementované diferenciální evoluce využívající umělý imunitní systém.

6.1.1 Goldberg's order-3

Výsledkem Goldberg's order-3 problému je hodnota fitness f daného vektoru. Vektor je rozdělen na 3bitové části a ty jsou následně ohodnoceny. Zápis $|x|$ značí počet jedniček (v trojici bitů):

$$f_1(x) = \begin{cases} 0,9 & \text{if } |x| = 0 \\ 0,6 & \text{if } |x| = 1 \\ 0,3 & \text{if } |x| = 2 \\ 1,0 & \text{if } |x| = 3 \end{cases}$$

Příklad: Mějme vektor $x = 010111010$, výpočet hodnoty f je: $f(010) + f(111) + f(010) = 0,6 + 1,0 + 0,6 = 2,2$

Maximální hodnota fitness jednoho jedince je pro tento problém $D/3$ (D – značí velikost jedince, tedy počet bitů), tu jsme schopni získat, pokud je vektor jedince $x = 111 \dots 111$. Naopak pro výpočet minimální hodnoty můžeme použít výpočet $D/3 - f$. Aplikováním vzorce je minimální hodnota 0. [36]

6.1.2 Bipolar's order-6

Podobně jako Goldberg's order-3 se i zde počítá hodnota fitness f po částech s tím rozdílem, že je vektor rozdělen na 6 bitové úseky:

$$f_1(x) = \begin{cases} 1,0 & \text{if } |x| = 0 \text{ or } 6 \\ 0,0 & \text{if } |x| = 1 \text{ or } 5 \\ 0,4 & \text{if } |x| = 2 \text{ or } 4 \\ 0,8 & \text{if } |x| = 3 \end{cases}$$

Příklad: Necht' existuje vektor $y = 010111010001$. Výpočet je následující: $f(010111) + f(010001) = 0,4 + 0,4 = 0,8$. Hodnota fitness f se tedy rovná $0,8$.

Pro Bipolární problém je maximální hodnota značená $D/6$, tedy pokud vektor jedince obsahuje vždy 6×0 nebo 6×1 . Minimální se nalézají na hodnotě 0 . [36]

6.1.3 Mühlenbein's order-5

Mühlenbeinův problém se liší od předchozích dvou problémů tím, že nepracuje s počtem 1 v jednotlivých částech vektoru, ale úseky vektoru musí odpovídat předem daným sekvencím bitů. Pokud neodpovídá žádná sekvence, je dílčí fitness hodnota rovna 0 .

$$f_1(x) = \begin{cases} 4,0 & \text{if } x = 00000 \\ 3,0 & \text{if } x = 00001 \\ 2,0 & \text{if } x = 00011 \\ 1,0 & \text{if } x = 00111 \\ 3,5 & \text{if } x = 11111 \\ 0,0 & \text{otherwise} \end{cases}$$

Příklad: Existuje vektor $y = 0011011111$. Rozdělení a výpočet je následující: $f(00110) + f(11111) = 0 + 3,5 = 3,5$. Hodnota f je $3,5$.

Pro Mühlenbeinův problém jsme schopni určit maximální hodnotu podle vzorce $3,5D/5$, této hodnoty dosáhneme, pokud vektory budou obsahovat jen 0 . Minimální hodnota je zde opět 0 . [36]

6.1.4 Clerc's Zebra3

Předchozí tři problémy jsou pro testování implementace jednoduché, proto vznikla modifikace Goldbergova problému – Clercův problém. Ten je o poznání komplikovanější než již výše rozebrané problémy.

Výsledkem je hodnota fitness f daného vektoru, který je, stejně jako u Goldbergova problému, rozdělen na 3bitové části. Pokud je bitová trojice sudá v pořadí (například nultá, druhá trojice, ...), používá se pro výpočet následující funkce:

$$f_1(x) = \begin{cases} 0,9 & \text{if } |x| = 0 \\ 0,6 & \text{if } |x| = 1 \\ 0,3 & \text{if } |x| = 2 \\ 1,0 & \text{if } |x| = 3 \end{cases}$$

Pokud je bitová trojice v pořadí lichá, je dílčí fitness dána předpisem:

$$f_1(x) = \begin{cases} 1,0 & \text{if } |x| = 0 \\ 0,3 & \text{if } |x| = 1 \\ 0,6 & \text{if } |x| = 2 \\ 0,9 & \text{if } |x| = 3 \end{cases}$$

Vektor ohodnocený maximální hodnotou fitness bude vypadat takto: $y = 111000111000\dots$ maximální hodnota výsledku je $D/3$. Pro hledání minimální hodnoty fitness (0), můžeme použít vzorec $D/3 - f$.

6.1.5 Multimodální problém

U tohoto typu problému se kromě počáteční populace jedinců vytváří ještě tzv. "Landscape", jde o soubor jedinců, kteří představují optima na prohledávaném prostoru. Každý jedinec z populace je poté srovnáván s jednotlivými prvky Landscape a jeho fitness hodnota je závislá na jejich podobnosti. Jedinci pak přísluší nejnižší fitness ze srovnání se všemi prvky Landscape. Detailnější popis i samotný algoritmus lze nalézt v [36].

6.2 Provedení experimentů

V následující tabulkách 2 a 3 jsou vypsány parametry použité v experimentech. Každý problém je testován pro tři bitové délky jedinců: 30, 60 a 90 (100 pro multimodální problém). Byla testována řada hodnot CR a Vmax, protože implementované řešení je velmi citlivé na vstupní parametry. Pouze ty nejvýhodnější hodnoty byly zvoleny pro následovné testování. Vzhledem k počtu variant a délce trvání jednotlivých experimentů bylo zvoleno 30 opakování, což by však nemělo mít zásadní dopad na výsledky.

Počet bitů Testovací problém	30		60		90	
	CR	vMax	CR	vMax	CR	vMax
Goldberg's order-3	0,7	30	0,7	60	0,7	90
Bipolar's order-6	0,7	2	0,7	4	0,7	6
Mühlenbein's order-5	0,7	2	0,7	2	0,7	2
Clerc's zebra3	0,4	15	0,4	20	0,4	20

Tabulka 2: Parametry použité při experimentech

Počet bitů Testovací problém	30			60			100		
	CR	vMax	Peaks	CR	vMax	Peaks	CR	vMax	Peaks
Multimodální problém	0,7	15	10	0,7	30	20	0,7	55	30

Tabulka 3: Parametry použité pro multimodální problém

6.3 Výsledky experimentů

V tabulkách 4 – 13 můžeme nalézt výsledky experimentů pro 5 problémů, které byly provedeny nad implementovaným řešením. Každá z následujících tabulek obsahuje tři hodnoty:

1. **Minimální fitness** – Hodnota představující výsledný součet fitness hodnoty celé populace.
2. **Průměr generací k nalezení minima** – Jedná se o průměr generací, při kterých bylo splněno nalezení minimální hodnoty fitness, tedy 0.
3. Třetí sloupec naznačuje, zda-li je negativní selekce (znaménko „-“) rychlejší, nebo pozitivní selekce (znaménko „+“) anebo jsou si obě selekce rovny (znaménko „=“). K tomuto vyhodnocení byl použit Wilcoxonův test.

Pro další testování byla vygenerována sada 30 jedinců, kteří byli použiti pro všechny testovací problémy, algoritmy selekce i druhy diferenciální evoluce. Grafy znázorňují rychlost vývoje celé populace jedinců v průběhu generací (obrázky 20 – 24).

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
DE/rand/1/exp	0	9,23	=	0	12,93	=	0	15,40	=
DE/best/1/bin	0	241,80	+	0	339,53	+	0	402,60	=
DE/rand/2/bin	0	7,76	=	0	10,60	=	0	11,80	=
DE/current-to-best/1/bin	0	5,86	=	0	7,86	=	0	8,50	=
DE/current-to-rand/1/bin	0	6,20	=	0	7,63	=	0	9,00	=

Tabulka 4: Goldbergův problém pro negativní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
DE/rand/1/exp	0	9,56	=	0	13,20	=	0	14,93	=
DE/best/1/bin	0	241,70	+	0	316,50	+	0	401,30	=
DE/rand/2/bin	0	7,70	=	0	10,26	=	0	11,20	=
DE/current-to-best/1/bin	0	5,36	=	0	7,70	=	0	8,36	=
DE/current-to-rand/1/bin	0	5,83	=	0	7,76	=	0	8,86	=

Tabulka 5: Goldbergův problém pro pozitivní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
DE/rand/1/exp	0	8,43	=	0	32,63	=	0	117,10	=
DE/best/1/bin	0	6,33	=	0	26,66	=	0	113,86	-
DE/rand/2/bin	0	5,13	=	0	33,60	=	0	129,43	=
DE/current-to-best/1/bin	0	4,93	=	0	22,93	-	0	278,30	+
DE/current-to-rand/1/bin	0	8,10	=	0	56,70	=	0	376,03	-

Tabulka 6: Bipolarův problém pro negativní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
DE/rand/1/exp	0	6,96	=	0	33,03	=	0	112,00	=
DE/best/1/bin	0	7,30	=	0	16,73	=	0	166,50	-
DE/rand/2/bin	0	4,60	=	0	33,20	=	0	153,20	=
DE/current-to-best/1/bin	0	8,30	=	0	36,43	-	0	255,03	+
DE/current-to-rand/1/bin	0	5,10	=	0	48,03	=	0	359,50	-

Tabulka 7: Bipolar problém pro pozitivní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
Druh Dif. Evoluce/Hodnoty									
DE/rand/1/exp	0	3,76	=	0	7,46	=	0	15,43	-
DE/best/1/bin	0	3,13	=	0	5,96	=	0	5,56	=
DE/rand/2/bin	0	40,00	=	0	8,40	=	0	17,06	-
DE/current-to-best/1/bin	0	12,06	=	0	26,13	+	0	341,5	+
DE/current-to-rand/1/bin	0	6,40	=	0	13,86	=	0	296,26	-

Tabulka 8: Mühlebeinův problém pro negativní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
Druh Dif. Evoluce/Hodnoty									
DE/rand/1/exp	0	3,60	=	0	8,33	=	0	21,46	-
DE/best/1/bin	0	2,86	=	0	5,46	=	0	5,80	=
DE/rand/2/bin	0	3,50	=	0	7,60	=	0	31,80	-
DE/current-to-best/1/bin	0	11,36	=	0	17,86	+	0	306,20	+
DE/current-to-rand/1/bin	0	10,46	=	0	18,23	=	0	274,06	-

Tabulka 9: Mühlebeinův problém pro pozitivní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
Druh Dif. Evoluce/Hodnoty									
DE/rand/1/exp	0	1040,16	=	1	3000+	=	1,21	5000+	=
DE/best/1/bin	0	1416,76	=	0,9	3000+	=	1,4	5000+	=
DE/rand/2/bin	0	1248,03	=	1	3000+	=	1,32	5000+	=
DE/current-to-best/1/bin	0,4	1500+	=	0,7	3000+	=	1,06	5000+	=
DE/current-to-rand/1/bin	0,3	1500+	=	0,9	3000+	=	1,18	5000+	=

Tabulka 10: Clercův problém pro negativní selekci

Počet bitů	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
Druh Dif. Evoluce/Hodnoty									
DE/rand/1/exp	0	1108,07	=	1	3000+	=	1,21	5000+	=
DE/best/1/bin	0	1464,73	=	0,9	3000+	=	1,4	5000+	=
DE/rand/2/bin	0	1328,10	=	1	3000+	=	1,32	5000+	=
DE/current-to-best/1/bin	0,3	1500+	=	0,7	3000+	=	1,06	5000+	=
DE/current-to-rand/1/bin	0,3	1500+	=	0,9	3000+	=	1,18	5000+	=

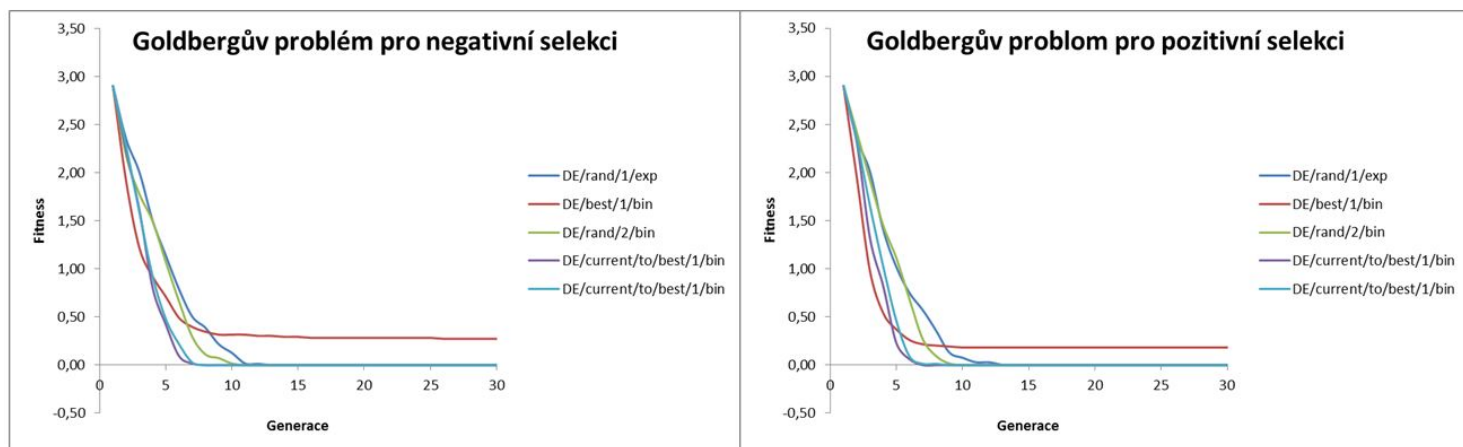
Tabulka 11: Clercův problém pro pozitivní selekci

Počet bitů Druh Dif. Evoluce/Hodnoty	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
DE/rand/1/exp	0	66,26	=	0	291,56	-	0,11	1200+	=
DE/best/1/bin	0	252,80	-	0,11	600+	=	0,16	1200+	=
DE/rand/2/bin	0	110,00	=	0	335,56	=	0,15	1200+	=
DE/current-to-best/1/bin	0,07	300,62	=	0,15	600+	=	0,15	1200+	=
DE/current-to-rand/1/bin	0,13	298,63	=	0,18	600+	=	0,16	1200+	=

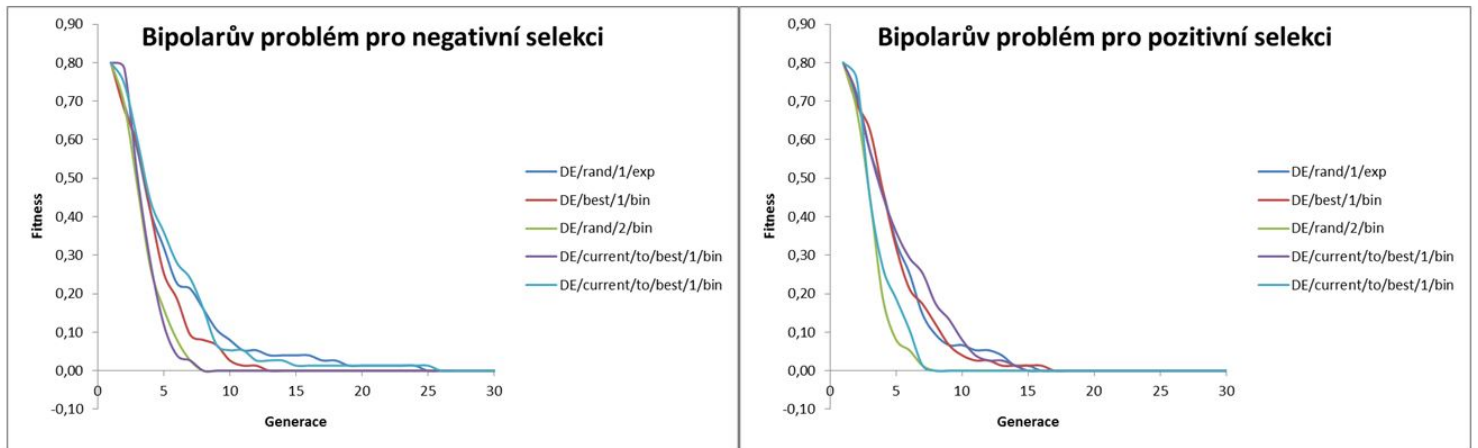
Tabulka 12: Multimodální problém pro negativní selekci

Počet bitů Druh Dif. Evoluce/Hodnoty	30			60			90		
	Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací		Min. Fitness	Průměr. generací	
DE/rand/1/exp	0	70,56	=	0	324,23	-	0,09	1200+	=
DE/best/1/bin	0	281,53	-	0,13	600+	=	0,18	1200+	=
DE/rand/2/bin	0	102,83	=	0	333,00	=	0,12	1200+	=
DE/current-to-best/1/bin	0	288,36	=	0,17	600+	=	0,19	1200+	=
DE/current-to-rand/1/bin	0,03	295,83	=	0,18	600+	=	0,2	1200+	=

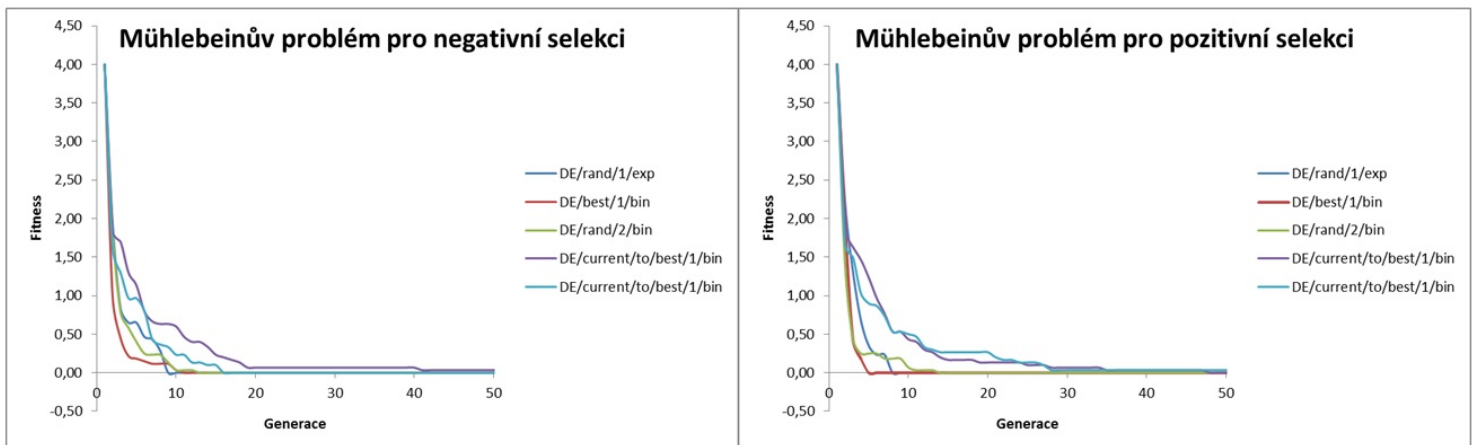
Tabulka 13: Multimodální problém pro pozitivní selekci



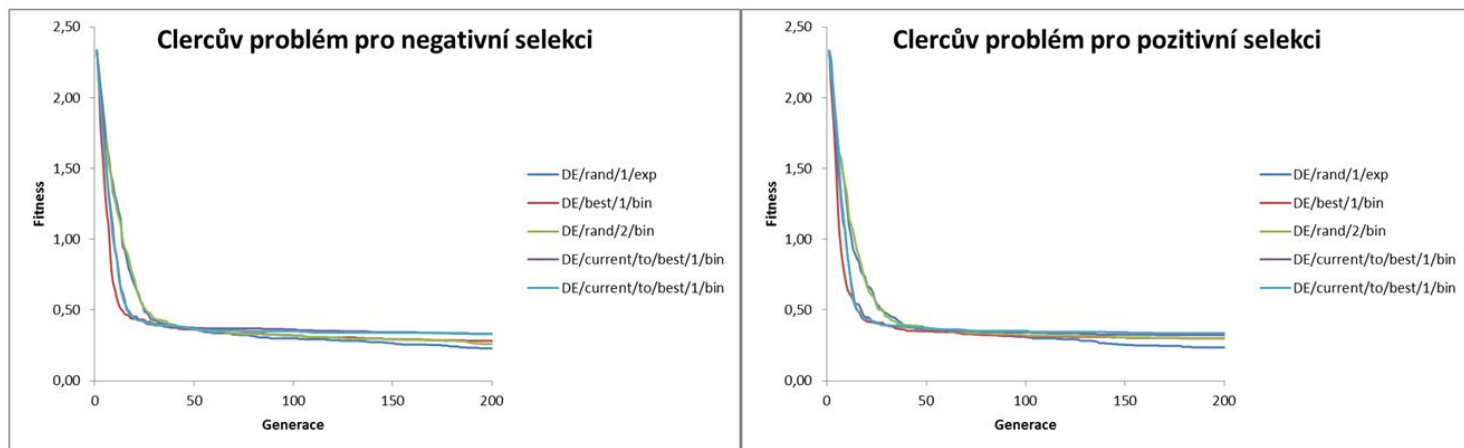
Obrázek 20: Goldbergs order-3



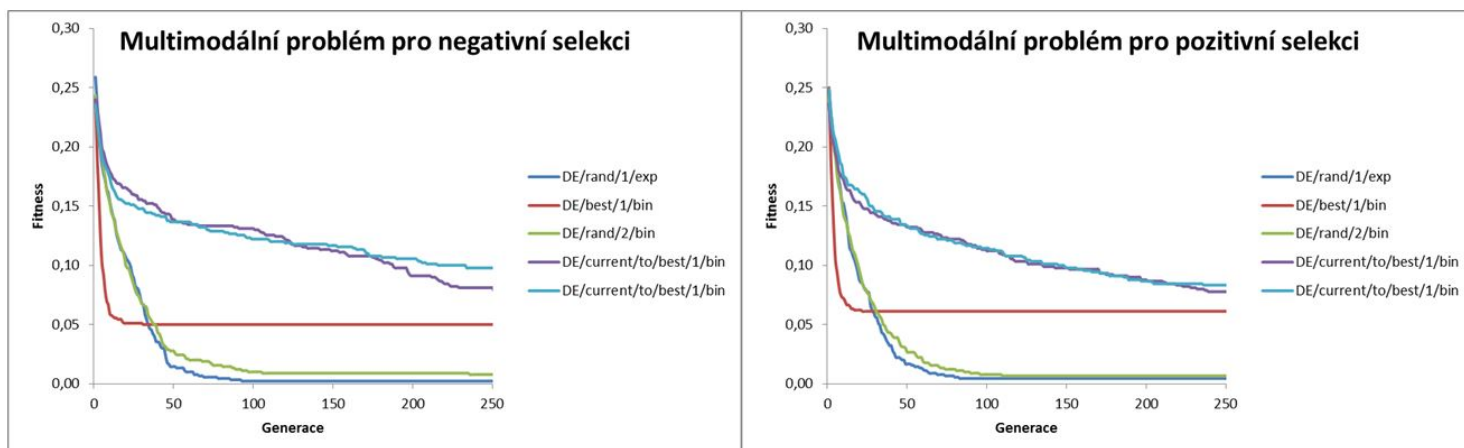
Obrázek 21: Bipolars order-6



Obrázek 22: Muehlebeins order-5



Obrázek 23: Clerc's Zebra3



Obrázek 24: Multimodální problém

6.4 Závěr experimentů

Při bližším pohledu na tabulku výsledků experimentů můžeme říci, že v přesně 62 případech (82,66 %) porovnávání negativní a pozitivní selekce jsou výsledky podle Wilcoxonova Signed-Rank testu statisticky shodné. V 8 případech (10,67 %) je negativní selekce efektivnější než pozitivní selekce. Z tabulek můžeme také zjistit, že v právě 5 případech (6,67 %) je pozitivní selekce účinnější než negativní selekce.

Dále je z tabulek patrné, že pro Goldbergův problém je nejvýhodnější použít variantu diferenciální evoluce DE/current/to/best/1/bin, jak pro pozitivní, tak i pro negativní selekci. Pro Bipolární problém nelze jednoznačně určit, který z druhů diferenciální evoluce je výhodnější použít. Z výsledků Mühlebeinova problému je, stejně jako u Goldbergova problému, efektivnější DE/best/1/bin pro oba druhy selekce. Clercův problém je pro implementovaný algoritmus komplikovaný natolik, že dokázal vyhledat optimální řešení pouze pro 30 bitové jedince, kde se osvědčil DE/rand/1/exp. I přesto však bylo pro vyhledání optima potřeba mnoho generací (tabulka 10 a 11). Stejný druh diferenciální evoluce je výhodné použít při řešení posledního, Multimodálního, problému. Pokud zahrneme do výsledků všeobecný pohled pro druhy diferenciální evoluce při řešení všech problémů, nejlepším řešením je DE/rand/1/exp. Na základě těchto poznatků můžeme říci, že zvolený druh diferenciální evoluce a nastavení počátečních parametrů hraje významnou roli za běhu implementovaného algoritmu a má zásadní vliv na výsledek.

Algoritmus se ukázal být velmi neefektivním při vyhledávání optima Clercova problému a v případě Multimodálního problému pro 90 bitové jedince.

7 Závěr

V rámci této diplomové práce jsme se seznámili s principy a vlastnostmi biologického i umělého imunitního systému. Byly objasněny některé selekční algoritmy a princip navázání detektoru na antigen, tyto poznatky byly následně využity při implementaci algoritmu diferenciální evoluce využívající umělý imunitní systém. Také jsme si vysvětlili principy fungování evolučních algoritmů, zejména diferenciální evoluce, jež je stěžejní součástí této práce.

Druhá polovina diplomové práce byla zasvěcena implementaci. Cílem bylo naimplementovat program diferenciální evoluce, který využívá některých vlastností umělého imunitního systému (a tím jej zefektivnit). Abychom si ověřili správné fungování a efektivitu implementovaného řešení, byla provedena řada experimentů pomocí testovacích problémů, které se v dnešní době využívají pro testování a porovnávání optimalizačních programů.

Po provedení experimentů jsme si ověřili, že ve větší polovině případů nezáleží, jaký druh selekce bude v implementaci použit.

Také jsme zjistili, že univerzální volbou (= vhodnou ve většině případů) druhu diferenciální evoluce je DE/rand/1/exp.

Věřím však, že existuje i další cesta integrace umělého imunitního systému do diferenciální evoluce, která může vést k efektivnějšímu fungování diferenciální evoluce.

8 Reference

- [1] FARMER, J.Doyne, Norman H PACKARD a Alan S PERELSON. *The immune system, adaptation, and machine learning* [online]. [cit. 2015-04-15]. ISBN 10.1016/0167-2789(86)90240-x
- [2] *Imunitní systém*. Biomach [online]. 2009 [cit. 2015-04-15]. Dostupné z: <http://www.biomach.cz/biologie-cloveka/imunitni-system>
- [3] Fakulta aplikovaných věd: Západočeská univerzita: Katedra informatiky a výpočetní techniky. *Biologická podstata evolučních algoritmů* [online]. 2000 [cit. 2015-04-15]. Dostupné z: http://www.kiv.zcu.cz/studies/predmety/uir/gen_alg2/E_alg.htm
- [4] NEUWIRTH, David. *Umělé imunitní výpočetní systémy*. Brno, 2007. Bakalářská práce. Vysoké učení technické v Brně.
- [5] LYTKOVÁ, Milada. *Novodobé epidemie infekčních onemocnění ve světě*. Brno, 2008. Bakalářská práce. Masarykova univerzita.
- [6] *World Health Organization* [online]. 2015 [cit. 2015-04-15]. Dostupné z: <http://www.who.int/gho/hiv/en/>
- [7] *Types of pathogens*. DairyCo [online]. 2015 [cit. 2015-04-15]. Dostupné z: <http://www.dairyco.org.uk/technical-information/animal-health-welfare/mastitis/pathogens-the-cause-of-mastitis/types-of-pathogens/#.VS5LW5Ov8qN>
- [8] *Difference Between Antigen and Pathogen*. Difference Between [online]. 2010 [cit. 2015-04-15]. Dostupné z: <http://www.differencebetween.net/science/health/difference-between-antigen-and-pathogen/>
- [9] *Neutrophils*. MedFriendly [online]. 2011 [cit. 2015-04-15]. Dostupné z: <http://www.medfriendly.com/neutrophil-neutropenia.html>
- [10] *Definition of Absolute neutrophil count*. MedicineNet [online]. 2012 [cit. 2015-04-15]. Dostupné z: <http://www.medicinenet.com/script/main/art.asp?articlekey=20030>
- [11] *Antimikrobiální peptidy: staré molekuly, nové ideje*. Akademický Bulletin [online]. 2012 [cit. 2015-04-15]. Dostupné z: <http://abicko.avcr.cz/2012/09/05/>
- [12] *Úloha NK buněk při léčbě rakoviny*. Přírodní lékárna [online]. 2013 [cit. 2015-04-15]. Dostupné z: <http://www.prirodnilekarna.cz/clanky-uloha-nk-bunek-pri-lecbe-rakoviny.html>

-
- [13] *Hammingova vzdálenost*. Algoritmy [online]. 2015 [cit. 2015-04-15]. Dostupné z: <http://www.algoritmy.net/article/1615/Hammingova-vzdalenost>
- [14] MURRAY, Patrick R.; ROSENTHAL, Ken S.; PFALLER, Michael A.. *Medical Microbiology, Fifth edition*. [s.l.] : Elsevier, 2005.
- [15] BARTUŇKOVÁ, J., ŠEDIVÁ, A., HÖLZELOVÁ, E. *Primární imunodeficiencie : Příručka pro pacienty a jejich rodiny*. FN Motol, Praha, 2. LF UK : Ústav imunologie. 1999. Dostupný z WWW: <<http://www.tigis.cz/Knihy/imuno/>>.
- [16] *Different types of Immunoglobulins*. Biology Exams 4 U [online]. 2014 [cit. 2015-04-15]. Dostupné z: <http://www.biologyexams4u.com/2012/11/different-types-of-immunoglobulins-igg.html#.VS5ToJOv8qN>
- [17] *Different types of Immunoglobulins*. Nature Publishing Group [online]. 2014 [cit. 2015-04-15]. Dostupné z: 17) <http://www.nature.com/nri/journal/v4/n7/full/nri1391.html>
- [18] DAŇKOVÁ, Jana. *Dendritické buňky*. Brno, 2007. Bakalářská práce. Masarykova univerzita.
- [19] BARRI, Dalibor. *Návrh a optimalizace filtru OTA-C s využitím evolučních algoritmů*. Praha, 2007. Diplomová práce. České vysoké učení technické v Praze.
- [20] *Artificial Immune System* [online]. 2013 [cit. 2015-04-15]. Dostupné z: <http://www.artificial-immune-systems.org/>
- [21] FERENČÍK, Miroslav. *Imunitní systém: informace pro každého*. [online]. 1. vyd. Překlad Kristýna Pokorná. Praha: Grada Publishing, c2005, 236, [4] s. [cit. 2015-04-15]. ISBN 80-247-1196-6.
- [22] *Geneticky podmíněné poruchy imunitního systému*. Genetika [online]. 2007, č.1 [cit. 2015-04-15]. Dostupné z: http://genetika.wz.cz/geneticke_poruchy_imunitniho_systemu.pdf
- [23] Xingshi He, Lin Han. *A novel binary differential evolution algorithm based on artificial immune systém.*, IEEE, 2007
- [24] prof. Ing. Ivan Zelinka, Ph.D. *Biologicky inspirované výpočty aneb vybrané statě z evolučních algoritmů*. BEN 2008
- [25] MOLNÁR, Zbyněk. *Testování výkonnosti evolučních algoritmů při aplikaci na deterministický chaos*. Zlín, 2010. Diplomová práce. Univerzita Tomáše Bati ve Zlíně.

-
- [26] *Autoimunitní onemocnění*. Mgr. Margit Slimáková, Ph.D. [online]. 2013 [cit. 2015-04-15]. Dostupné z: <http://www.margit.cz/autoimunitni-onemocneni/>
- [27] *Základní logické funkce*. Server reálných aplikací "352LAB.vsb.cz": Vysoká škola báňská [online]. 2010 [cit. 2015-04-26]. Dostupné z: http://352lab.vsb.cz/MinServer/PraceStud/LPaS/Srek/Soubory_LogickePrvky/LogickeFunkce.htm
- [28] *Shluková analýza*. KUČERA, Jiří. Masarykova univerzita [online]. 2010 [cit. 2015-04-15]. Dostupné z: http://is.muni.cz/th/172767/fi_b/5739129/web/web/main.html
- [29] J. Timmis, T. Knight, L. N. de Castro, E. Hard. *An Overview of Artificial Immune Systems*. University of Kent. 2003
- [30] VOLNÁ, Eva. *Uměla inteligence*. [online]. 2013 [cit. 2015-04-15]. Dostupné z: http://www1.osu.cz/volna/Umela_inteligence_skripta.pdf
- [31] KIM, Jungwon a Peter BENTLEY. *The human immune system and network intrusion detection*. CiteSeerX [online]. 1999 [cit. 2015-04-15]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.367&rep=rep1&type=pdf>
- [32] MITSUMOTO, N., T. FUKUDA, F. ARAI, H. TADASHI, T. IDOGAKI, Nikolaus CORRELL a Alcherio MARTINOLI. *Self-organizing multiple robotic system (a population control through biologically inspired immune network architecture)*. [online]. [cit. 2015-04-15]. ISBN 10.1007/4-431-35881-1_4.
- [33] ENDOH, S., TOMA, N., YAMADA, K. *Immune Algorithm for n-TSP*. IEEE International Conference on 1998 : Systems, Man and Cybernetics. 1998, vol. 4, s. 3844-3849.
- [34] CAYZER, S. *Artificial Immune System*. HP Labs Bristol. 2003.
- [35] STORN, Rainer a Kenneth PRICE. *Journal of Global Optimization*. [online]. s. 341-359 [cit. 2015-04-16]. ISSN 09255001. DOI: 10.1023/A:1008202821328. Dostupné z: <http://link.springer.com/10.1023/A:1008202821328>
- [36] M. Clerc. *Binary Particle Swarm Optimisers: toolbox, derivations, and mathematical insights*. 2007
- [37] FORREST, Stephanie a Alan S. PERELSON *Computation and the immune system*. ACM SIGBIO Newsletter, v.12 n.2, p.52-57, June 1992
- [38] ROY, Rajkumar a Raju S. BAPI. *Book review: Artificial Intelligence in Perspective by Daniel G. Bobrow*. (Ed.) (The MIT Press 1993), ACM SIGART Bulletin, v.7 n.1, p.10-13, Jan. 1996

-
- [39] HALAVATI, Ramin, Saeed Bagheri SHOURAKI, Mojdeh Jalali HERAVI a Bahareh Jafari JASHMI *An artificial immune system with partially specified antibodies*. Proceedings of the 9th annual conference on Genetic and evolutionary computation, July 07-11, 2007, London, England
- [40] TŮMA, Jiří. *Samoopravné kódy*. Matematicko-fyzikální fakulta Univerzita Karlova v Praze [online]. 2012, č. 8 [cit. 2015-04-16]. Dostupné z: <http://www.karlin.mff.cuni.cz/tuma/2002/NLinalg8.pdf>
- [41] *Shluková analýza*. Masarykova universita [online]. 2012 [cit. 2015-04-16]. Dostupné z: http://www.fsp.s.muni.cz/sebera/vicerozmerna_statistika/shlukova.html
- [42] *Základní logické funkce*. Server reálných aplikací "352LAB.vsb.cz"[online]. 2011 [cit. 2015-04-16]. Dostupné z: <http://352lab.vsb.cz/ServerFinalVer/UcebniceLPaS/text/zaklofce/zaklofce.htm>
- [43] KENNETH V. PRICE, Kenneth V. Rainer M. *Differential evolution a practical approach to global optimization*. [online]. Berlin [etc.]: Springer, 2005 [cit. 2015-04-16]. ISBN 978-354-0313-069.
- [44] *Toll-Like Receptors Review*. InvivoGen [online]. 2012 [cit. 2015-04-26]. Dostupné z: <http://www.invivogen.com/review-tlr>
- [45] *Shluková analýza*. České vysoké učení technické v Praze: Fakulta dopravní [online]. 2008, č. 1 [cit. 2015-04-26]. Dostupné z: <http://www.fd.cvut.cz/personal/nagyivan/Projekty/Classification/ShlukovaAnalyza.pdf>
- [46] KEPHART, J. O., SORKIN, G. B., SWIMMER, M., WHITE, S. R. *Blueprint for a Computer Immune System*. 1999.
- [47] *Výměna látek*. Mendelova univerzita v Brně [online]. 2010 [cit. 2015-05-03]. Dostupné z: <http://user.mendelu.cz/sladek/cytologie/vymena.html>