VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

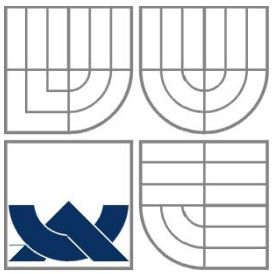
APLIKACE UMĚLÝCH IMUNITNÍCH SYSTÉMŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

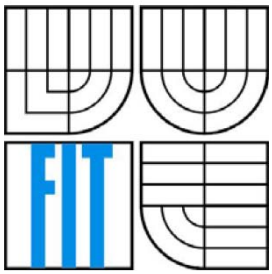
AUTOR PRÁCE
AUTHOR

Bc. Petr Dolejší

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

APLIKACE UMĚLÝCH IMUNITNÍCH SYSTÉMŮ

APPLICATION OF ARTIFICIAL IMMUNE SYSTEMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. Petr Dolejší
AUTHOR

VEDOUCÍ PRÁCE — Doc. Ing. Josef Schwarz, CSc.
SUPERVISOR

BRNO 2008

Abstrakt

Tato diplomová práce, se snaží přiblížit čtenáři principy a vlastnosti biologického imunitního systému, následně abstrahovat z těchto znalostí principy, jež je možné využívat při aplikaci na umělé imunitní systémy. Poskytnout náhled na praktické aplikace, které již tyto myšlenky využívají a rozšiřují a předvést skutečnou implementaci aplikace, která je postavena na těchto principech.

Klíčová slova

imunitní systém, umělý imunitní systém, algoritmus pozitivní selekce, algoritmus negativní selekce, klonální selekční algoritmus, optimalizace, multimodální problém, teorie imunitní sítě.

Abstract

This final year thesis introduces the principles and properties of the artificial immune systems to the reader, then abstracts the principles from this knowledge and applies the real artificial immune systems on them. It provides a view at the practical applications that use and extend given ideas.

Keywords

immune system, artificial immune system, positive selection algorithm, negative selection algorithm, clone selection algorithm, optimization, multimodal system, immune network theory.

Citace

Dolejší Petr: Aplikace umělých imunitních systémů. Brno, 2008, diplomová práce, FIT VUT v Brně.

Aplikace umělých imunitních systémů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením doc. Ing., CSc. Josefa Schwarze a tímto mu také děkuji za jeho odbornou pomoc. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……

Jméno Příjmení
Datum

Poděkování

Chtěl bych poděkovat především doc. Ing., CSc. Josefu Schwarzovi za jeho odborné vedení, pomoc a za poskytnutou literaturu, kterou jsem využil při tvorbě této diplomové práce. Dále bych chtěl poděkovat všem, kteří mi pomohli k dokončení této práce.

Obsah

1 Úvod

Umělý imunitní systém je v poslední době objektem vysokého zájmu, nejen díky svým silným výpočetním schopnostem. Jak již název napovídá je inspirován biologickým imunitním systémem, který je jedním z nejdůležitějších a také nejsložitějších systémů v lidském těle, jehož úlohou je zabezpečit obranu před cizími látkami, jako jsou bakterie, viry a jiné škodliviny z okolí. Jeho ohromnou vlastností je schopnost rozpoznat a reagovat na látky, se kterými se doposud nestřetl a díky paměti se i rychleji vypořádat s vetřelcem, kterého již zná.

Z výpočetního hlediska je imunitní systém vysoce paralelní systém, jenž využívá učení, paměť, a asociační vyhledávání k rozpoznávání a klasifikaci úloh. Tato práce se snaží popsat přirozený imunitní systém a k němu v paralele možnosti a techniky jeho využití ve zpracování informací. Tyto techniky byly úspěšně použity například v úlohách pro rozpoznávání vzorů, detekci chyb a diagnostice či v počítačové bezpečnosti jako účinná detekce a obrana proti virům.

Jsou zde představeny dvě aplikace, které jsou součástí této práce, postavené na principech imunitního systému. Jedna pro optimalizaci mutlimodálních funkcí a druhá pro prezentaci možných principů využitelných v počítačové bezpečnosti (například k detekci virů).

2 Biologický imunitní systém

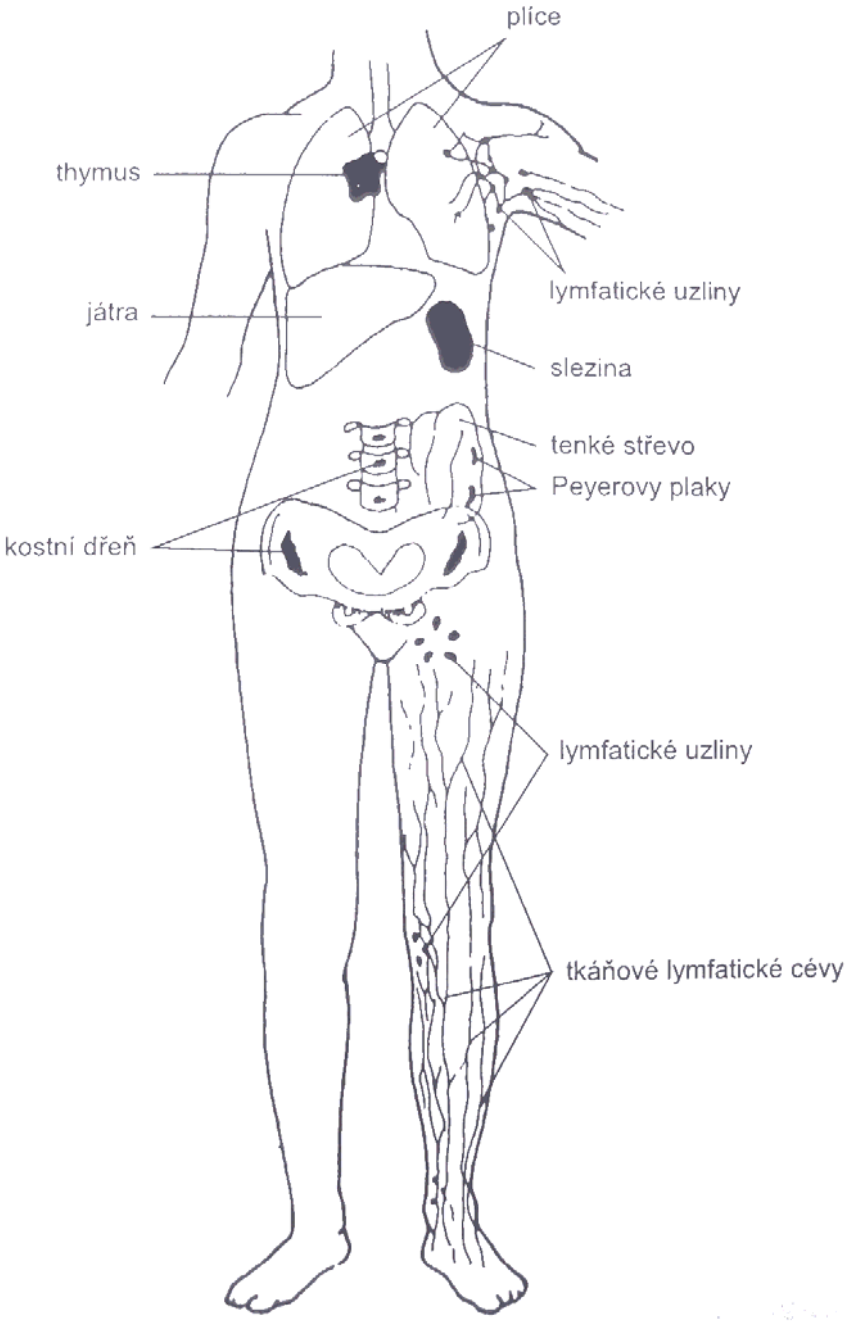
V celé této práci je pod pojmem biologická imunita myšlen přirozený imunitní systém člověka, potažmo savců všeobecně. Abychom dokázali využít imunitní systém pro řešení problémů, je potřebné mít alespoň základní znalosti o jeho struktuře a funkci.

Hlavní úlohou imunitního systému je zabezpečit obranu organismu před různými parazitickými organismy jako jsou například bakterie a viry. Dokáže rozpoznat a reagovat i na látky, se kterými se ještě nesetkal a díky paměti se dokáže rychleji vypořádat s útočníky, které už zná.

2.1 Hlavní části imunitního systému

Vzhledem k tomu, že cizím mikroorganizmem může být napadena kterákoliv část těla, musí mít imunitní systém zajištěn přístup ke všem orgánům a tkáním. Součástí imunitního systému jsou imunitní orgány a jednotlivé buňky, které se vyskytují buď volně nebo seskupené do shluků.

Celkově imunitní systém lze charakterizovat jako autonomní systém, jehož orgány, tkáně a buňky jsou rozloženy po celém organismu. Tyto orgány nazýváme lymfatické. Dělí se na primární, jež jsou zodpovědné za produkci lymfocytů a sekundární, ve kterých probíhá samotná imunitní reakce [1].



Obrázek 1 – Schéma rozmístění primárních a sekundárních lymfoidních orgánů v lidském těle. Převzato z [1].

Thymus: funguje jako jakási univerzita pro lymfocyty, kde se z nich stávají T-lymfocyty.

Kostní dřeň: je zodpovědná za produkci lymfocytů

Lymfatické uzliny: spojení lymfatických cév a místa kde nastává specifická imunitní reakce

Lymfatické cévy: slouží k transportu lymfy, což je tekutina obsahující lymfocyty a antigeny, do krve a lymfatických orgánů.

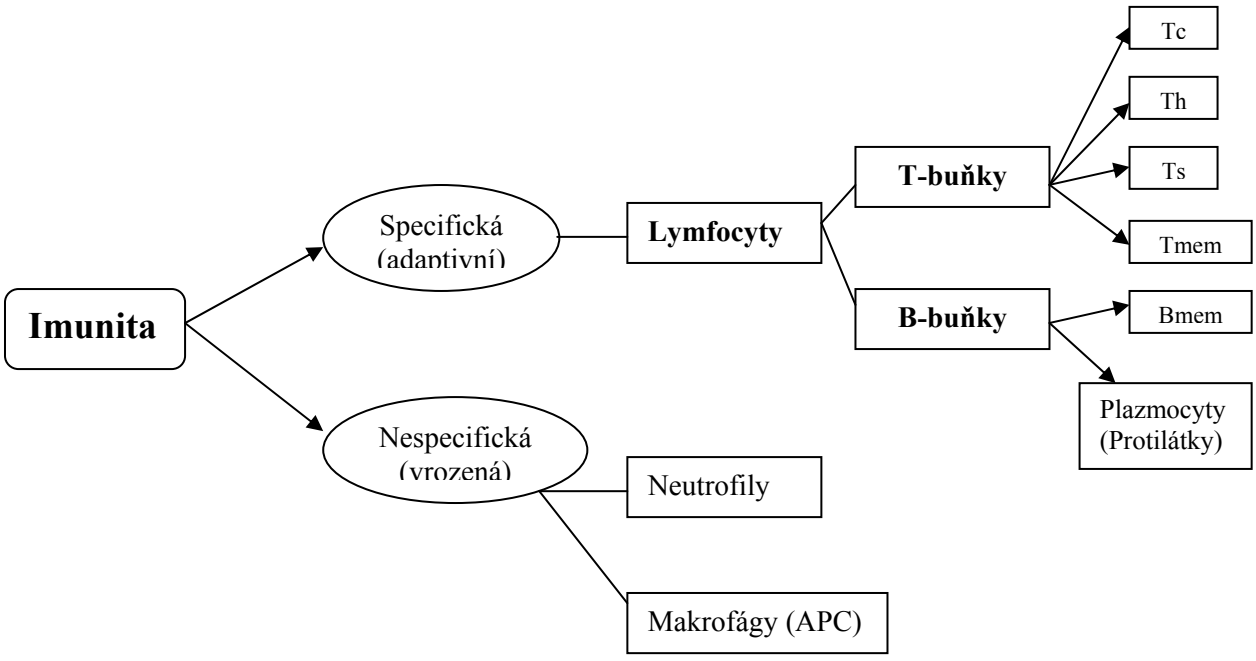
Slezina: kontroluje přítomnost antigenů přímo v krvi.

Slepé střevo a Peyerovy plaky: speciální lymfatické uzliny chránící trávicí systém.

2.2 Funkce imunitního systému

Imunitní systém je velmi komplexní systém s mnoha funkčními komponentami. Poskytuje víceúrovňovou obranu proti útočníkům skrze: nespecifickou imunitu (vrozenou) a specifickou imunitu (adaptivní). Hlavní funkcí imunitního systému je rozpoznání všech buněk (popřípadě molekul) uvnitř těla a roztřídit je na buňky vlastní a nevlastní. Nevlastní buňky jsou později kategorizovány ke stimulování odpovídajícího typu obranného mechanismu. Imunitní systém se evolucí učí rozpoznávat mezi tělu vlastními buňkami a molekulami a cizími antigeny (viry, bakterie apod.).

Hlavním výkonným prvkem z pohledu funkčnosti systému je tedy buňka. Počet imunitních buněk, které kolují po celém těle v rámci krevního a lymfatického oběhu je asi 10^{12} [2]. Jednotlivé buňky spolu komunikují prostřednictvím různých enzymů a bílkovin. Lymfocyt je hlavním typem imunitní buňky, jejíž vlastnosti jsou: specifičnost, diverzita, paměť a adaptivita. Ostatní buňky nazýváme fagocyty a patří mezi podpůrné buňky imunitního systému. Lymfocyty dělíme na 2 základní typy buněk: T-lymfocyty a B-lymfocyty. V primárních lymfatických orgánech lymfocyty dozrávají a dostávají antigenovou výbavu. T-lymfocyty se vyvíjejí v kostní dřeni a dozrávají v thymusu, zatímco B-lymfocyty se vyvíjejí a dozrávají uvnitř kostní dřeně. Existuje velké množství různých druhů imunitních buněk, proto si popíšeme pouze ty nejdůležitější.



Obrázek 2 - Dělení typu buněk podle typu imunity

2.2.1 Funkce jednotlivých typů buněk

2.2.1.1 Buňky nespecifické imunity

Do této skupiny buněk patří nejpočetnější skupina lymfocytů a to takzvané fagocytické buňky, jež jsou v podstatě bílé krvinky schopné požírat jiné buňky.

Granulocyty: (neutrofily,eozinofily,bazofily) typ bílých krvinek s vysoce destrukčním vlivem na mikroorganismy.

Makrofágy: bílé krvinky se schopností rozpoznat a požírat mikroorganismy. Mají schopnost na svém povrchu prezentovat antigen **APC (antigen presenting cells)**.

APC (antigen presenting cells): heterogenní skupina buněk, která upravuje a vystavuje antigen na svém povrchu v takové formě (MHC), aby ho mohly rozpoznat T-lymfocyty (popřípadě i B-lymfocyty) a zahájit vůči němu specifickou imunitní reakci. T-lymfocyty totiž nejsou schopny rozpoznat antigen přímo, ale pouze ve vazbě s MHC molekulami.

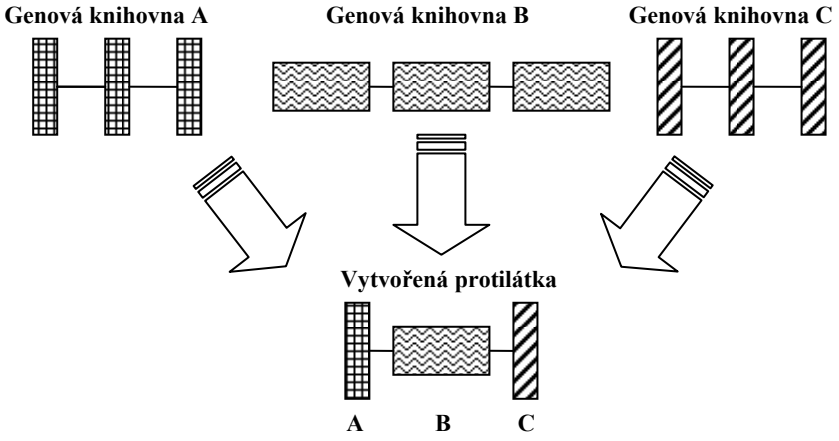
2.2.1.2 Buňky specifické imunity

Specifickou imunitní reakci zabezpečují dva různé typy lymfocytů: T-lymfocyty a B-lymfocyty [3] [1].

B-lymfocyty: po kontaktu s antigenem se mění na buňky syntetizující protilátky (plazmatické buňky: Plazmocyty a B-paměťové), a proto zodpovídají za protilátkový typ specifické reakce. Jsou produkovány v kostní dřeni, proto „B“ lymfocyty z anglického bone marrow. Jejich vývoj je možné rozdělit na 2 stádia. Na stádium *nezávislé* a *závislé na přítomnosti antigenu* v organismu. Stádium nezávislé na přítomnosti antigenu se odehrává v kostní dřeni a představuje přeměnu B-lymfocytů na zralé, plně funkční buňky. Předpokládá se, že se každý den vyprodukuje přibližně 5 x 10^7 nových B-lymfocytů, z čehož se do oběhu dostane přibližně pouze 10%. Zbylé jsou eliminovány negativní a pozitivní selekcí, přičemž většinou jde o lymfocyty citlivé na vlastní buňky těla. Zralé B-lymfocyty opouštějí kostní dřeň a dostávají se do sekundárních lymfatických orgánů. Jakmile se střetne s antigenem, pro který je trénovaný, aktivuje se a začne se přeměňovat na plazmatické buňky (Plazmocyty) a paměťové buňky (B-paměťové). Tato přeměna je závislá na přítomnosti antigenu v organismu, proto je toto stádium nazýváno závislé na antigenu.

Plazmatické buňky (Plazmocyty) mají životnost pouze několik dní a uplatňují se při akutní fázi infekce. Produkují velké množství protilátek, tzv. **imunoglobulinů** (**Ig**), jež jsou téměř identické s membránovým receptorem původního aktivovaného B-lymfocytu a tedy specifické vůči antigenu, který jejich produkci stimuloval. Protilátky

se nacházející převážně v plazmě, ale také slinách, slzách, mateřském mléce atp. Molekula protilátky má tvar písmene „Y“ a je tvořena čtyřmi polypeptidickými řetězci (viz obrázek 10). U člověka rozlišujeme podle stavby pět základních tříd imunoglobulinů: IgG, IgA, IgE, IgM, IgD. Imunoglobuliny nemohou patogenní organizmus zničit samy, ale označí jej jako cíl ostatních obranných systémů. Protilátky se váží přímo na antigen, „obalí“ ho a tím se antigen stává terčem pro makrofágy. Protilátky jsou vysoce specializované bílkoviny schopné reagovat prakticky s jakýmkoliv antigenem, a tedy se součástmi jakéhokoliv mikroorganizmu, který napadne organizmus. Existují tedy specifické protilátky, které reagují na konkrétní antigeny. Struktura protilátek je dána geny plazmocytů a to tak, že tvar protilátky je určen skupinou vybraných genů z genových knihoven mateřské buňky.



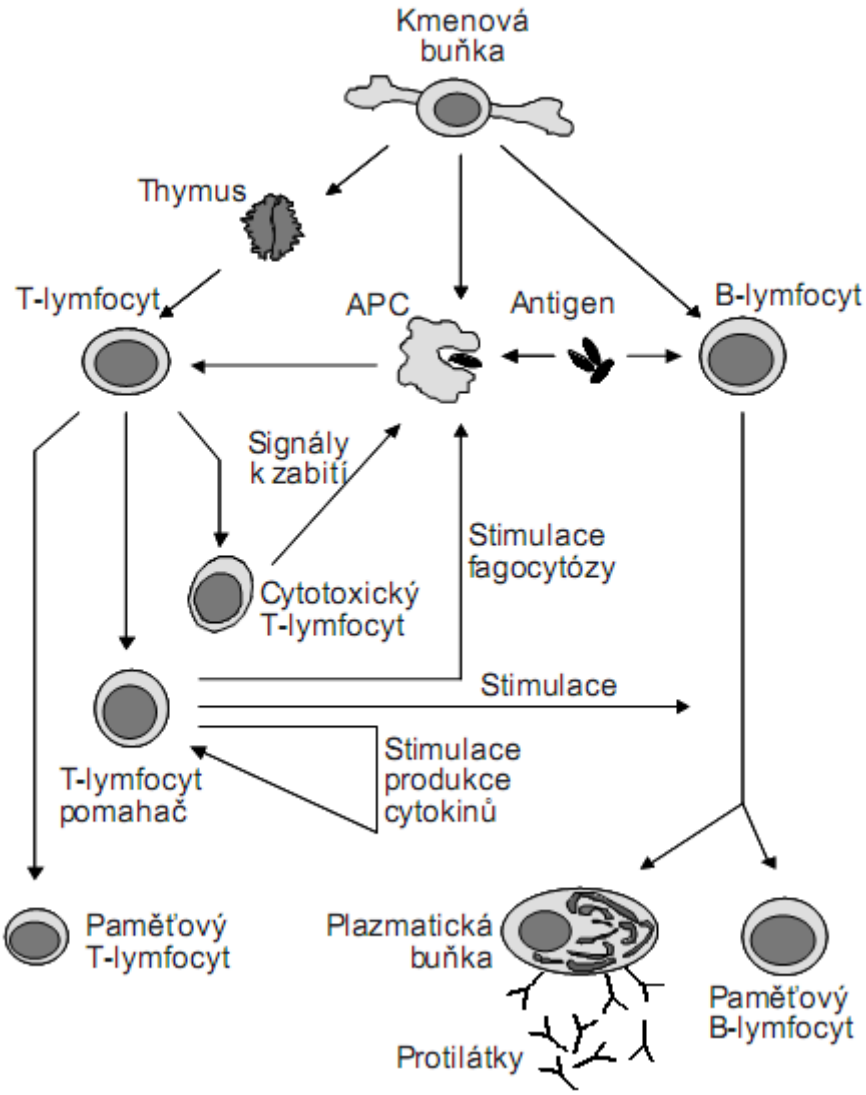
Obrázek 3 - Schematické znázornění vytvoření molekuly protilátky z knihoven genů

B-paměťové: vznikají v menším počtu a na rozdíl od plazmatických buněk se vyznačují dlouhodobou životností. Jsou připraveny při opakované infekci zasáhnout mnohem rychleji a masivněji než při prvním kontaktu s příslušným antigenem. Organizmus je tzv. **imunizován** proti určitému antigenu.

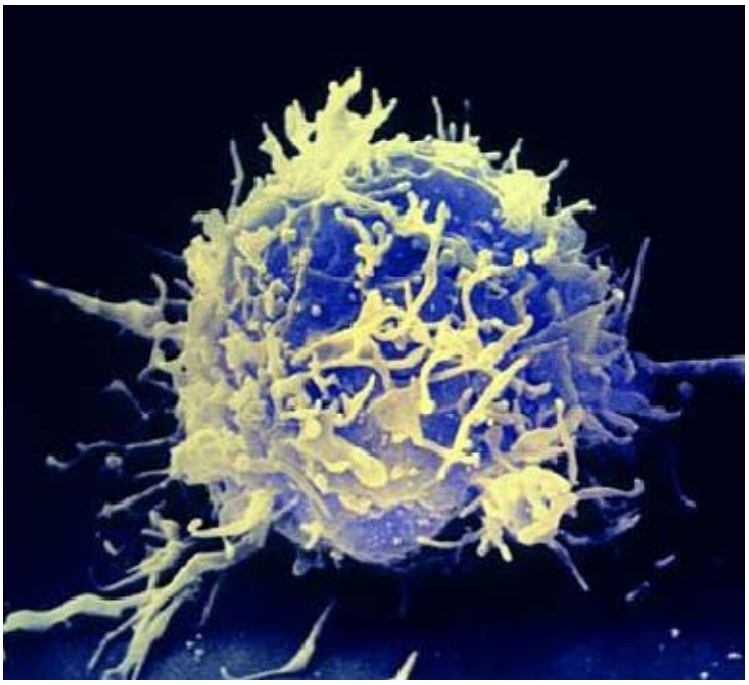
T-lymfocyty: jsou základní buňky specifické (získané) imunity dozrávající v thymusu. Některé jejich subpopulace mají typické výkonné funkce (**T-helper** – mají pomocnou funkci pro jiné imunitní buňky. Například aktivují NK, či T-supressor buňky. **T-supressor** - plní funkci tlumení imunitních reakcí. **T-cytotoxic –** ničí infikované buňky, které se shodují s jejich specifičností. Přímo zabíjejí buňky, které jsou napadené mikroorganizmy. Některé mikroorganizmy (zvláště viry) mají schopnost přežívat a dokonce se i množit. Tyto buňky musejí být zlikvidovány, aby se infekce dále nešířila, což právě zabezpečují Tc lymfocyty.

NK-buňky – jsou přirození zabíječi. Nespecifikovaným způsobem usmrcují vlastní nádorově transformované buňky nebo buňky infikované viry. Představují základní buňky přirozené obrany proti spontánně vznikajícím nádorů a virům a jsou příčinou odmítnutí transplantovaných orgánů.

NKT-buňky – jsou velmi málo početná skupina buněk nacházejících se v lymfatických orgánech majících některé vlastnosti NK-buněk a T-lymfocytů**.**



Obrázek 4 - Specifická imunita. Naivní B-lymfocyty jsou přímo stimulovány antigenem, dělí se, vznikají plazmatické buňky produkující protilátky a paměťové buňky. Naivní T-lymfocyty jsou stimulovány antigenem na povrchu APC. Převzato z [3].



Obrázek 5 - T-lymfocyt na elektronovém mikroskopu.

2.2.2 Nespecifická (vrozená) imunita

Jedná se o imunitu neadaptivní, je vrozená a její mechanismy mohou být v případě infekce použity okamžitě. Reaguje řádově v průběhu několika minut a představuje první linii obranného mechanismu. Tento typ imunity je evolučně starší a vyskytuje se v různých formách v celé živočišné říši od bezobratlých až po savce včetně člověka. Základní rysy této imunity jsou:

- **Je vrozená**. Mechanismy této imunity má organizmus od narození, bez ohledu na to, zda se setkal s příslušným antigenem nebo ne.
- **Není specifická**. Buňky podílející se na nespecifické imunitě zasahují stejným způsobem proti jakékoli částici, která byla rozpoznána jako cizorodá.
- **Nemá imunologickou paměť**. Buňky nespecifické imunity zasahují vždy stejnou silou a to i po opakovaném kontaktu s konkrétním antigenem.

Výkonné složky (součásti) vrozené imunity dle [3], [1], [6], jsou:

a) **Fyzikální a chemické bariéry organizmu.** Kůže, sliznice chráněné mukózním sekretem a řasinkové epitely vytvářejí mechanickou zábranu proti pronikání cizorodých látek do organizmu. Největší riziko vniknutí cizorodých částic je přes sliznice trávicího, dýchacího a močopohlavního ústrojí, kde je organizmus oddělen od vnějšího prostředí pouze tenkou vrstvou buněk. Ty však produkují antibakteriální látky a vytvářejí tak chemickou bariéru. Např. sliny a slzy obsahují lysozymy, narušující bakteriální stěnu. Také žaludeční šťávy svým nízkým pH představují antibakteriální prostředí.

b) **Basofily** uvolňující ze svých granul histamin. Účinkem histaminu dochází k dilataci cév a zvyšuje se permeabilita vlásečnic. Usnadní se tak průchod proteinů a leukocytů z krve do tkání, kde mohou čelit patogenu. Vzniká zánětlivá reakce doprovázená teplotou, otoky a lokálním zarudnutím.

c) **Fagocytóza makrofágy a neutrofily**. Pokud dojde k překonání fyzikálních bariér a cizorodá částice pronikne do tkání, vzniká zánět. Cizorodá látka se dostává do styku s fagocytujícími buňkami, které jsou nadané schopností améboidního pohybu, ty se k ní přiblíží a fagocytózou ji pohltí. V místě infekce vzniká **hnis,** tvořený odumřelými makrofágy a neutrofily.

d) **Komplementové proteiny**. Je to skupina tkáňových a membránových proteinů, které jsou lokálně aktivovány v místě zánětu. Působí jednak jako chemický atraktant pro leukocyty, dále obklopují buňku bakterie a tím usnadňují její rozpoznání fagocyty. V neposlední řadě pak některé proteiny komplementu ničí baktérie tím, že se zabudovávají do jejich membrán a vytvoří v ní póry, přes které následně nekontrolovaně pronikají dovnitř sodné ionty a dochází k destrukci buňky.

e) **Buňky NK** jsou typem leukocytů specializovaným na nespecifickou obranu proti virům a nádorovým buňkám. Rozpoznávají změny na povrchu buněk infikovaných viry a usmrcují je. Tím znemožní další množení virů a také zpřístupní viry dalším složkám obranného systému.

Patří sem i komplementární kaskádová reakce, což je způsob eliminace virů a bakterií prostřednictvím řetězce aktivovaných bílkovin nacházejících se v krvi a interferony. Interferony mají schopnost aktivovat produkci enzymů blokujících množení se viru a tím chránit buňky těla. Tento typ imunity je též zodpovědný za vyvolání signálů v APC buňkách (antigen presenting cells), které způsobují aktivaci T-lymfocytů a tím aktivaci specifické imunitní reakce.

Nespecifická imunita je tedy velice rychlá a reaguje na všechny druhy cizích organizmů, čímž vzniká základní požadavek organizmu a to spolehlivě rozpoznávat cizí a vlastní buňky organizmu.

2.2.3 Specifická imunita (získaná)

Její název je odvozen od specifické reakce buněk patřících do této skupiny na určitý konkrétní antigen, který se nejprve musí naučit rozpoznávat.

Je evolučně vyspělejším typem imunity. K jejím základním znakům patří [1], [6]:

- **Není vrozená**. Organizmus ji získává teprve během života a to zpravidla až po setkání s příslušným antigenem.
- **Specificky rozpoznává cizorodé látky (antigeny).** Každá specifická imunitní buňka určitého typu (klonu) je geneticky předurčena k rozpoznání pouze jediného druhu antigenu.
- **Vyznačuje se imunologickou pamětí**. Opakované setkání s konkrétním antigenem vyvolává stále silnější a rychlejší imunitní odpověď.

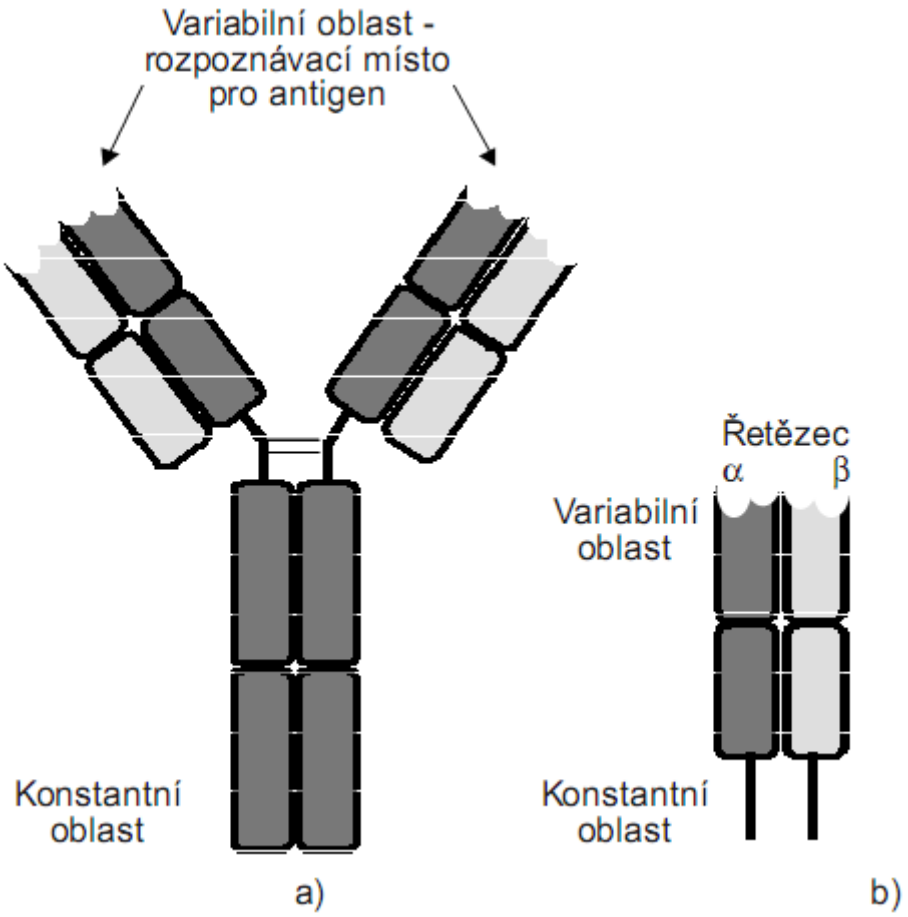
Na povrchu všech buněk těla se nacházejí individuálně specifické membránové proteiny (MHC[1]), což jsou jakési značky umožňující imunitnímu systému rozpoznat, která buňka je vlastní a která cizí. K formování těchto informačních proteinů na membránách buněk dochází již během embryonálního vývoje jedince. V tu dobu se s nimi také seznamuje dozrávající imunitní systém a dochází k vytvoření seznamu značek, které musí být imunologicky tolerovány [6]. To znamená, že veškeré tkáně vlastního těla, se kterými přijde formující se imunitní systém embrya do styku, jsou v budoucnu považovány za vlastní a není proti nim iniciována žádná imunitní odpověď. Po narození jsou už všechny odlišné molekuly považovány za cizorodé a je proti nim spuštěna imunitní reakce. Takovéto cizí molekuly jsou nazývány **antigeny**[2].

Antigeny jsou v těle rozpoznávány prostřednictvím B a T-lymfocytů. Ty mají na svém povrchu buněčné receptory, které jsou schopny se specificky vázat na konkrétní antigen. Buněčné receptory jsou membránové proteiny, jejichž stavba je analogická jak u B tak u T-lymfocytů.

---

[1] **MHC** je komplex molekul představující typ identifikace buňky, který obaluje antigen. Tento identifikátor, rozpoznatelný imunitním systémem, má na povrchu většina buněk. T-lymfocyty umějí rozpoznávat pouze tyto komplexy molekul, na rozdíl od B buněk rozpoznávajících přímo antigeny.

[2] **Antigen** je původně označení pro látku, která vyvolává tvorbu protilátky (z angl. antibody generating). Jedná se tedy o jakoukoli cizorodou látku, která vyvolává imunitní odpověď. V praxi pojmem antigen zpravidla označujeme **patogeny** - viry, bakterie, plísně (resp. části a produkty jejich těl), stejně jako tkáně jiných jedinců.



Obrázek 6 - a) Schéma struktury protilátky B-lymfocytu b) T-lymfocytu. Převzato z [3].

Protilátka obsahuje **vazební místo pro antigen**, jež je tvořeno **specifickou kombinací** několika málo aminokyselin. Protože antigenní molekula je zpravidla příliš velká, váže se membránový receptor svým vazebným místem jen na určitou malou část původního antigenu, na tzv. **epitop** neboli antigenní determinant. Specifičnost imunitní odpovědi proti určitému konkrétnímu antigenu je založena na unikátní komplementaritě jednoho epitopu s odpovídajícím receptorem. Receptor s epitopem do sebe tedy zapadají jako *klíč do zámku* [3].

Každý receptorově specifický typ leukocytu spolu s jeho shodnými kopiemi nazýváme **klon**. Každý klon je po jeho vzniku reprezentován pouze několika málo leukocyty, které se nazývají naivní lymfocyty. Jakmile se však **poprvé** (**primární imunitní odpověď**) setká s příslušným antigenem, začne se dělit, aby vytvořil dostatečný počet svých kopií, schopných zlikvidovat danou infekci. Tomuto procesu se říká **klonální** expanze. Po likvidaci infekce se zachovává pouze malá část buněk, sloužících jako **paměťové buňky**, které jsou při opětovném setkání (**sekundární imunitní odpověď**) se stejným antigenem schopny reagovat již mnohem rychleji a intenzivněji.

Formování buněčných receptorů je jedním z nejsložitějších procesů v imunologii, jelikož organizmus musí být připraven čelit invazi milionů typů antigenu. Na jedné straně je potřeba zajistit obrovskou **diverzitu,** pokud se týká rozlišovací schopnosti, na straně druhé je však nutné důsledně

eliminovat všechny receptory potenciálně schopné reagovat s molekulami vlastního těla. K tomuto formování membránových receptorů (tzv. dozrávání lymfocytů) dochází v kostní dřeni a thymusu, kde náhodnými rekombinacemi (mutací) vznikají milióny jeho variant.

2.2.4 Popis činnosti specifické obrany

Na specifické reakci organizmu se lymfocyty podílejí různě:

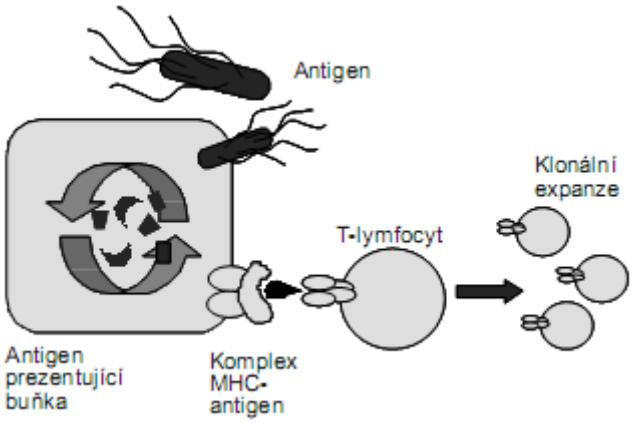
B-lymfocyty produkují protilátky tzv. **látková imunita**.

T-lymfocyty napadenou buňku buď zabijí nebo vydávají signály dalším buňkám k posílení imunitních reakcí proti danému patogenu tzv. **buněčná imunita**.

T-lymfocyty nejsou schopny s antigenem reagovat přímo, na rozdíl od B-lymfocytů. Ke své stimulaci potřebují tzv. antigen prezentující buňky.

Antigen prezentující buňky (**APC** = antigen presenting cell) jsou buňky vlastního organizmu, které jsou schopny fagocytovat. Neustále pohlcují ze svého okolí částice, které pak rozkládají na vzorky (tzv. epitopy), jež jsou neustále vynášeny na povrch buňky k testu, zda se jedná o látku cizorodou, škodlivou nebo o součásti vlastního těla. Mimo to mají na svých membránách tzv. **MHC molekuly** (major histocompatibility complex).

Teprve komplex MHC molekuly s antigenem vystavený na povrchu APC buňky je schopen aktivovat příslušný T-lymfocyt.



Obrázek 7 - Stimulace T-lymfocytu: Antigen je pohlcen APC buňkou a rozložen. Jeho část vystaví na svém povrchu na MHC molekulách. Takovýto komplex MHC-antigen specificky stimuluje příslušný T-lymfocyt, který se dále dělí a produkuje Tc buňky, které ničí nákazu. Převzato z [3].

MHC molekuly tedy určují individuální identitu všech tkání a jsou zodpovědné i za komplikace, které specifický imunitní systém působí při lékařských zákrocích např. za odmítnutí transplantovaného orgánu [1]. Pokud je jedinci transplantována cizí tkáň s jiným typem MHC molekul než má sám, jsou příjemcem rozeznány a je na ni aktivována specifická obrana. Pro zvýšení pravděpodobnosti úspěšného přijetí transplantátu se tedy velmi dbá na to aby [3]:

- byla dodržena co největší podobnost dárce a příjemce v MHC proteinech (podobnost se zvyšuje u příbuzných)
- se potlačila imunitní reakce organismu. To má i stinnou stránku, jelikož je pacient vzhledem k uměle potlačené imunitě vystaven většímu riziku infekce apod.

2.3 Základní vlastnosti a principy

Imunitní systém je charakteristický svojí adaptabilitou, robustností, mohutným paralelizmem s decentralizovaným způsobem řízení a distributivní strukturou [5]. Umělý imunitní systém je inspirován svým biologickým obrazem, nevyužívá však přesné jeho modely, ale spíše se inspiruje jeho základními vlastnostmi a principy. Jeho základní vlastnosti využitelné v informatice se dají shrnout do následujících bodů [5], [7]:

- **Distribuovanost a decentralizovanost**

V celém imunitním systému je absence centrálního prvku, který by řídil jeho činnost. Ta je koordinována aktivitou jednotlivých prvků na buňkové úrovni. Buňky imunitního systému jsou rovnoměrně distribuované po celém organizmu, čímž je docílena rychlá odezva.

- **Rozpoznávání vzorů (pattern recognition)**

Rozpoznání a eliminace téměř jakéhokoliv patogenu je umožněna díky obrovské diverzitě buněčných receptorů (jejich výběr probíhá negativní a pozitivní selekcí) a důmyslné komunikaci mezi buňkami. Utváření buněčných receptorů je částečně podmíněno genetickou výbavou jedince a z části i náhodnou kombinací. Konkrétní patogen však může být rozpoznán větším počtem podobných si receptorů a to díky další vlastnosti: odolnosti proti šumu.

- **Odolnost vůči šumu**

Na rozpoznání určitého patogenu není nutné, aby rozpoznávací buňky měly přesně nastavené receptory. Umožňuje to jistá tolerance, díky níž je možné reagovat i na podobné antigeny.

- **Paralelismus**

Organizmus musí nepřetržitě paralelně zpracovávat množství rozlišných druhů signálů, toho je schopný díky absenci centrálního řídícího prvku a jeho distribuovanosti.

- **Jedinečnost**

Imunitní systém každého organizmu je jedinečný a to dokonce i v rámci jednoho živočišného druhu. Je to zabezpečeno pomocí MHC molekul. Tato vlastnost se projevuje při transplantacích orgánů, kdy tělo nechce přijmout dárcův orgán a reaguje na něj jako na cizí objekt. Právě z tohoto důvodu se při transplantacích podávají látky snižující aktivitu T-lymfocytů, které napadají darovaný orgán jako nežádoucí a vyhledává se pokrevní dárce [3].

- **Paměť a schopnost učit se**

Díky schopnosti některých B a T lymfocytů přeměnit se na tzv. paměťové buňky je docílena rychlejší sekundární imunitní reakce organizmu na patogen, se kterým se již organizmus dříve setkal.

- **Autoregulace odezvy**

Imunitní systém sám reguluje množení se obranných buněk dle patogenu. Po zničení patogenu opět poklesne množství imunitních buněk na stálou hladinu.

- **Detekce anomálií**

Díky schopnosti rozpoznávat vlastní a nevlastní buňky pomocí negativní selekce, je organizmus schopen reagovat i na patogeny, se kterými se dosud nesetkal a také na buňky jemu vlastní, které se nechovají korektně (například nádorové buňky).

- **Robustnost**

Všechny výše uvedené vlastnosti se podílejí na vysoké robustnosti celého imunitního systému. Odpovídá tomu i to, že imunitní systém není závislý na jednotlivých buňkách (ty denně hynou a jsou nahrazovány novými). Imunitní systém funguje jako celek poskládaný z několika spolu komunikujících mechanismů, pracujících současně na různých úrovních, které se vzájemně překrývají a doplňují. Takže ani při vyřazení jednoho nedojde ke zhroucení celého systému.

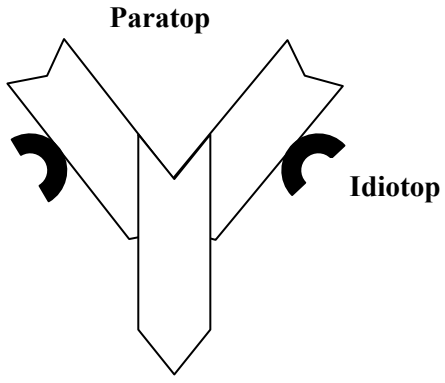
2.4 Teorie imunitní sítě

Teorie imunitní sítě představuje konceptuálně odlišný přístup a chápání toho, jak mezi sebou jednotlivé komponenty imunitního systému interreagují a odpovídají na cizí podněty (antigeny). Tato teorie, jinak nazývaná také teorií idiotypické sítě, kterou formálně představil Neils Kaj Jerne roku 1974 (za svoji práci dostal Nobelovu cenu), představovala nový přístup k důležitým vlastnostem, jako je například učení a paměť, vlastní tolerance, velikost a diverzita imunitního repertoáru. Jerne představuje imunitní síť jako novou fundamentální myšlenku, jak vysvětlit fenomény jako jsou selekce, tolerance, rozlišování vlastních a nevlastních prvků a paměť.

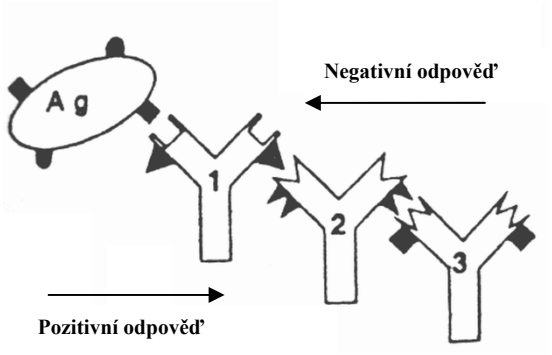
Předpokládá schopnost imunitního systému rozpoznat nejen cizí prvky tj. antigeny, ale i vlastní buňky a molekuly. Tuto teorii podpořili i pokusy se zvířaty, ta měla schopnost vytvářet protilátky rozpoznávající jiné protilátky produkované jedinci stejného druhu.

Model imunity popsaný v [4] popisuje protilátku plnící dvě funkce: rozpoznává a je sama rozpoznávána. Ta část molekuly protilátky, která plní funkci detekce antigenu, je nazývána **paratop** a část, která specificky reprezentuje protilátku samotnou, se nazývá **idiotop** viz obrázek 8, odtud je teorie imunitní sítě nazývána též jako teorie **idiotypické** sítě.

Imunitní systém je pak formálně popsán jako ohromná komplexní síť paratopů, které rozpoznávají množinu idiotopů a idiotopů, které rozpoznávají množinu paratopů. Potom může být každý element rozpoznán stejně tak dobře, jako sám rozpoznává jiné. Tato vlastnost vede k ustavení sítě, a jelikož se molekuly protilátek objevují jak volné tak i vázané receptory k B-buňkám, tato síť vzájemně propojuje buňky a molekuly. Potom dle protilátky rozpoznané apitopem nebo idiotopem, může odpovědět, buď pozitivně, nebo negativně, na tento rozpoznávací signál. Pozitivní odpověď by měla být na aktivaci buňky, růst buňky a protilátkovou sekreci, zatímco negativní odpověď by měla následovat na toleranci a potlačení protilátky. Obrázek 9 zobrazuje negativní a pozitivní odpověď v teorii imunitní sítě.

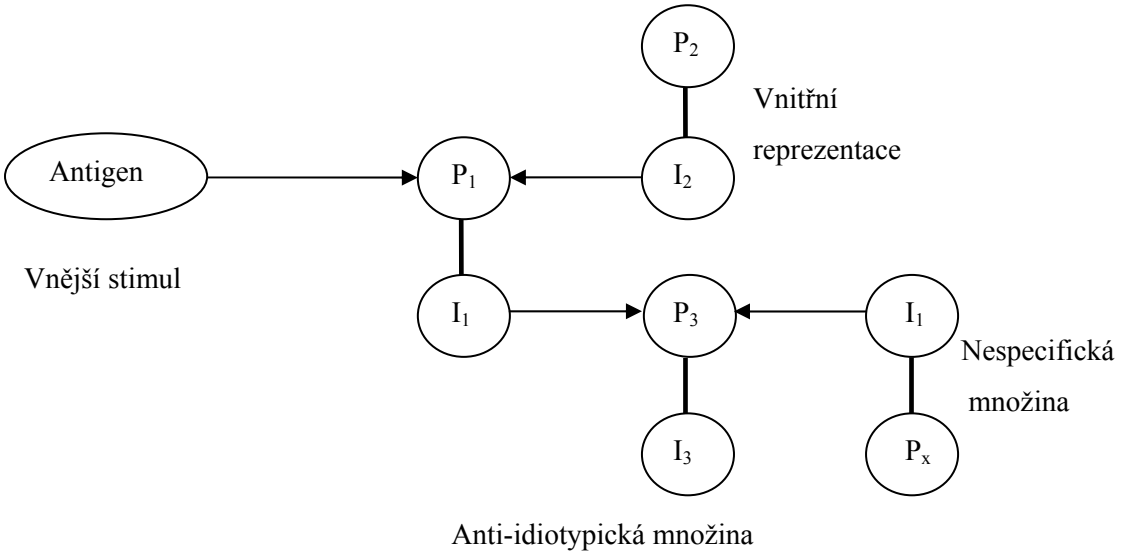


Obrázek 8 – Molekula protilátky v teorii imunitní sítě



Obrázek 9 – Pozitivní a negativní odpověď jako výsledek interakce mezi paratopem a idiotopem nebo epitopem.

Chování imunitní sítě je popsáno na obrázku 10. Jakmile se setká imunitní systém s antigenem, jeho epitopy jsou rozpoznány množinou různých paratopů nazvaných p_1. Tyto paratopy se vyskytují na protilátkách a receptorech molekul společně s určitými idiotopy, takže množina p_1 paratopů je asociována s množinou i_1 idiotopů. Symbol p_1i_1 značí celou množinu rozpoznávajících molekuly protilátky a potenciální odpovídající lymfocyty vzhledem k antigenu (Ag). Uvnitř imunitní sítě každý paratop množiny p_1 rozpoznává množinu idiotopů a celá množina p_1 rozpoznává úměrně větší množinu idiotopů. Množina i_2 idiotopů se nazývá vnitřní reprezentace epitopu (nebo antigenu), protože je rozpoznávána stejnou množinou p_1, která rozpoznává antigen. Množina i_2 je asociována s množinou p_2 paratopů vyskytujících se na molekulách a buněčných receptorech v množině p_2i_2. Navíc je každý idiotop množiny p_1i_1 rozpoznáván množinou paratopů, takže celá množina i_1 je rozpoznávána úměrně větší množinou p_3 paratopů, které se vyskytují společně s množinou i_3 idiotopů na protilátkách a lymfocytech v anti-idiotypické množině p_3i_3. Dle tohoto schéma dostáváme stále větší a větší množinu, která rozpoznává nebo je rozpoznávána předchozí definovanou množinou uvnitř sítě. Kromě množiny p_1i_1 ještě existuje paralelní množina p_xi_1 protilátek, které zobrazují idiotopy množiny i_1 v molekulové asociaci a takové kombinaci, která neodpovídá cizímu epitopu. Šipka indikuje stimulující efekt, kdy jsou idiotopy rozpoznávány paratopy a potlačující efekt kdy, paratopy rozpoznávají idiotopy.



Obrázek 10 - Molekulová interakce v imunitním systému dle idiotypické teorie sítě. Vnitřní reprezentace má stimulační efekt na rozpoznávací množinu, zatímco anti-idiotypická množina má inhibitující efekt.

Takováto struktura by pak umožňovala protilátkám vytvářet skupiny, které se dokážou samočinně vázat na sebe do řetězců, což by vedlo k autonomii a samoorganizaci celé imunitní sítě. Podobné systémy byly předmětem zájmu pana Varela [14], který přišel s myšlenkou, že takovéto samoorganizované sítě mohou být aktivované sami a rozpoznávat sami sebe i bez přítomnosti antigenů. Potom by hlavní činností imunitního systému bylo udržování vnitřní dynamické rovnováhy sítě pomocí změny koncentrace imunitních buněk při rozhození rovnováhy vstupem antigenu do organizmu. Této teorii napomáhá i fakt potřeby neustálé stimulace paměťových B a T buněk, aby se udržovali v dostatečném množství na sekundární reakci. Právě tato teorie umožňuje vysvětlit způsob stimulace paměťových buněk řetězcem navzájem navázaných protilátek, které tvoří interní reprezentaci antigenu i bez jeho fyzické přítomnosti v organizmu.

2.4.1 Model imunitní sítě

Původní formální model imunitní sítě byl navržený v [4] a dále rozvinutý v [18]. Tyto modely byli spojité a založené na modelování sítě prostřednictvím diferenciálních rovnic, které vyjadřují dynamiku množiny identických lymfocytů c_i [7]:

$$\frac{\delta c_i}{\delta t} = c_i \sum_{j=1}^{N_1} f(E_j, K_j, t) - c_i \sum_{j=1}^{N_2} g(I_{j,} K_j, t) + k_1 - k_2 c_i$$

Kde funkce f a g vyjadřují stimulující a inhibitující signály od jiných lymfocytů v síti, přičemž k_1 je koeficient přísunu nových lymfocytů a k_2 koeficient určující úhyn lymfocytů. Nevýhoda tohoto spojitého modelu je problém nalezení funkce f a g. Tato práce se opírá o výzkum Stewarta a Varela [19], která ukazuje kognitivní vlastnosti modelu idiotypické sítě. Pokusily se v ní popsat matematický model, který byl však vysoce nelineární a tedy neproveditelný. Vytvořily proto výpočetní model idiotypické síť v symbolickém prostoru, kde dimense koresponduje se stereochemickou charakteristikou, jež popisuje kombinaci sítě. Potom vzdálenost mezi dvěma entitami v tomto prostoru representuje jejich charakteristické rozdíly. Více o této práci je pojednáno v [20]. Aplikací modelu imunitní sítě je aiNet[3] od autorů [8].

3 Umělé imunitní systémy

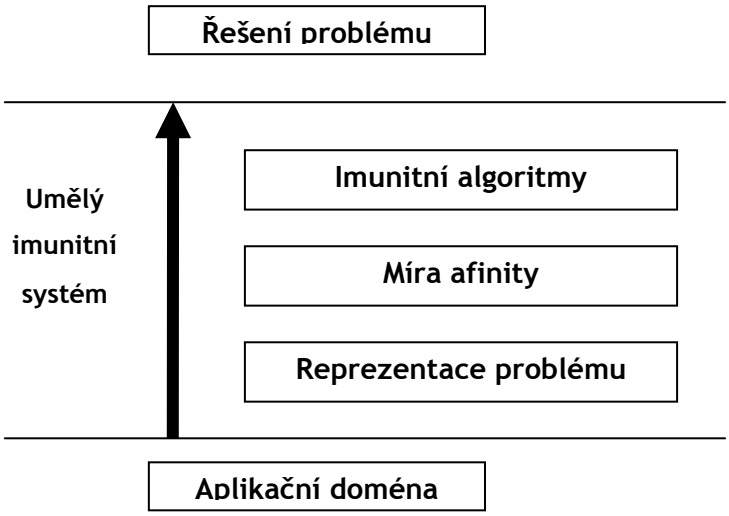
3.1 Obecné schéma umělého imunitního systému

De Castro a Timmis ve své publikaci Artificial Immune Systems: A New Computational Intelligence Approach [5] definují umělý imunitní systém takto:

„Umělé imunitní systémy jsou adaptivní systémy inspirované teoretickou imunologií pozorovanými funkcemi imunity a jejími principy a modely. Takovéto systémy jsou pak aplikovány na řešení problému“.

[3] Protilátkám odpovídají prvky v modelu, tj. prvky odpovídají řetězcům atributů reprezentovaným v Euklidovském příznakovém prostoru. Prezentace prvků mezi sebou je určena mírou afinity. Algoritmus vybírá prvky s nejvyšší afinitou, klonuje prvky úměrně k jejich afinitě, nepřímou úměrou mutuje prvky nové vzhledem k jejich afinitě a vybírá skupinu prvků s vysokou afinitou, aby prezentovala v síti paměť [7].

Umělý imunitní systém je určité řešení konkrétního problému, které je inspirováno principy biologického imunitního systému. Typické schéma uveřejněné v [5] vypadá následovně:



Obrázek číslo 11 - Obecné schéma umělého imunitního systému. Převzato z [5].

3.1.1 Reprezentace problému

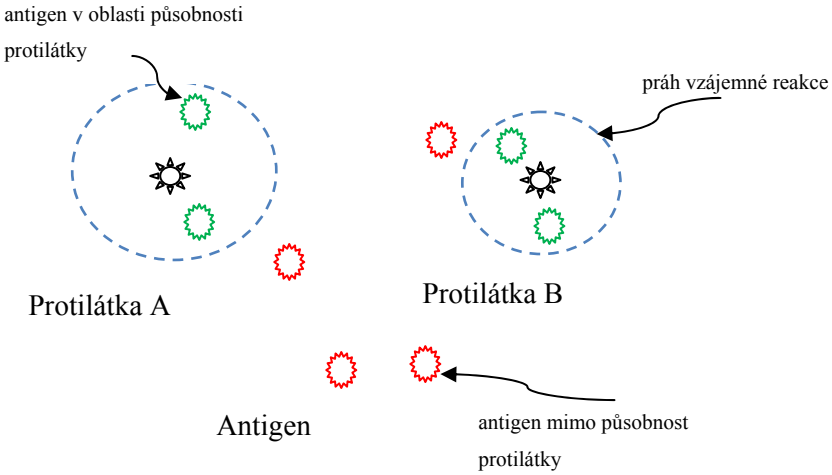
Způsob, jakým jsou obvykle mapovány prvky biologického imunitního systému na umělý imunitní systém, ukazuje následující tabulka [6]:

Biologický imunitní systém	Umělý imunitní algoritmus
Antigen	**Problém**
Protilátka	**Řešení problému**
Afinita	**Míra vhodnosti daného řešení - fitness**
Produkce protilátek z paměťových buněk	**Vyvolání a obnovení minulých úspěšných řešení**
Produkce protilátek	**Použití genetických operátorů k tvorbě nových řešení**

Tabulka 1. Analogie mezi imunitním algoritmem a biologickým imunitním systémem.

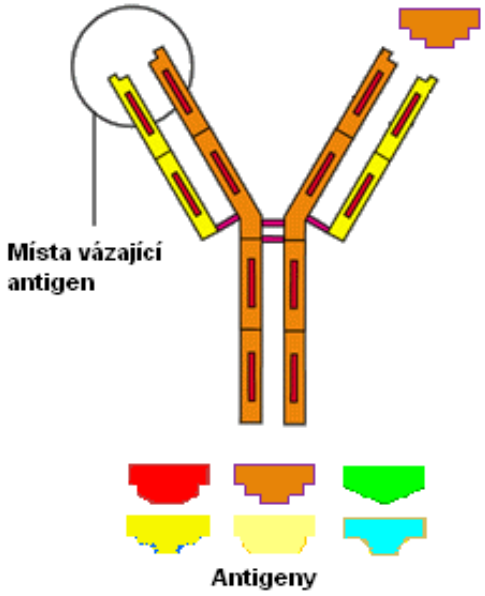
3.1.2 Míra afinity

Míra afinity je analogií v umělých imunitních systémech pro vhodnost daného řešení. Biologický imunitní systém se vyznačuje svou odolností vůči šumu, tj. receptory buněk nereagují pouze na přesně daný patogen ale i na patogen jemu blízký. To, do jaké míry odlišnosti patogenu je na něj buňka ještě schopna reagovat, se nazývá míra afinity. Tato tolerance afinity je reprezentována oblastí vlivu nazývanou **práh vzájemné reakce** (cross-reactivity threshold) viz obrázek 12.



Obrázek 12 – Ilustrace prahu vzájemné reakce protilátky

Analogicky v umělém imunitním systému si představme binární řetězec představující patogen a jiný binární řetězec jako detektor shody. K tomu, aby takovýto detektor navázal na antigen, by musela nastat naprostá shoda v jednotlivých bitech binárních řetězců. V praxi je toto těžce splnitelný požadavek a využívá se tedy (analogie odolnosti vůči šumu) **míra shodnosti daného řešení**.
Pro posouzení podobnosti existuje několik metod, pomocí kterých se dá určit, jak moc si jsou dva prvky podobné. Patří mezi ně například **Hammingova** vzdálenost, **Euklidova vzdálenost** nebo tzv. **Manhattanská vzdálenost**.



Obrázek 13 – Ilustrace molekuly protilátky s místem navazujícím antigeny a její vhodnosti na konkrétní antigen.

Hammingova vzdálenost

Je jednou z nejpoužívanějších metod, která se nejčastěji definuje jako počet odlišných prvků ve dvou řetězcích neboli počet změn, které musíme provést, abychom z jednoho řetězce dostali druhý. Pokud by tedy binární řetězec představující patogen vypadal například takto: 0010110 a binární řetězec jako detektor shody následovně: 1010100 je jejich Hammingova vzdálenost rovna 2.

Hammingova vzdálenost:

$$D = \sum_{i=1}^{L} \delta \quad \text{kde} \quad \delta \begin{cases} 1 \; pokud \; A_i \neq B_i \\ 0 \; jinak \end{cases}$$

Euklidova vzdálenost:

$$D = \sqrt{\sum_i (A_i - B_i)^2}$$

Manhattanská vzdálenost:

$$D = \sum_i |A_i - B_i|$$

Kde A_1 až A_n a B_i až B_n jsou souřadnice dvou prvků. Uvažujme například řetězce 1234 a 1324. Euklidova vzdálenost je pak $\sqrt{2}$ a Manhattanská vzdálenost je 2.

3.1.3 Imunitní algoritmy

Imunitní algoritmy říkají, jakým způsobem se bude nakládat s populací modelu imunitního systému.

- Jak bude vytvářena generace imunitních buněk, respektive receptory pro model imunitního systému – někdy nazýváno jako **model kostní dřeně**.
- Jak vybírat vhodné buňky z vygenerované populace pro správný chod imunitního systému – **algoritmy pozitivní** a **negativní selekce** (**model thymusu**).
- Jak dále vylepšovat detekci – algoritmus **klonální selekce** (evoluce detektorů).

Generování nové populace

Jelikož jsou všechny krvinky i lymfocyty tvořeny v kostní dřeni, je tento model tvorby nové populace nazýván modelem kostní dřeně.

Nejjednodušším takovýmto modelem je náhodné vygenerování řetězce určitých atributů dané délky. Tyto atributy se odvíjejí od daného problému, může to být například binární řetězec či vektor.

Složitější model zas náhodně vybírá z knihovny genů, která obsahuje různé možné hodnoty atributů a vytváří tak nový prvek.

Paměť

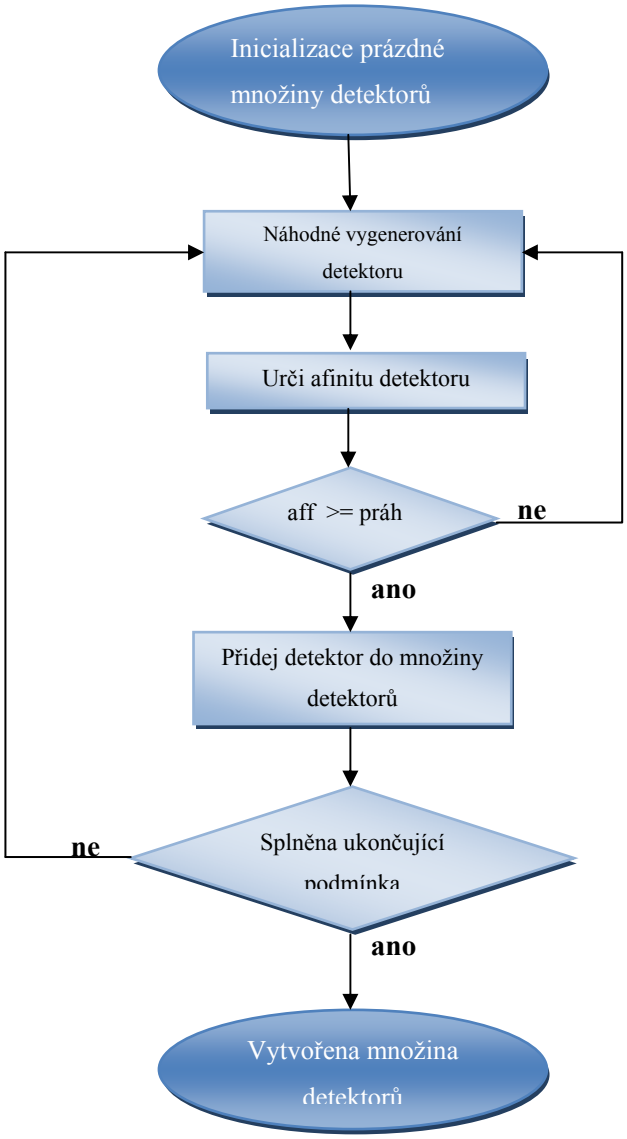
Jednou z velmi důležitých vlastností imunitního systému je paměť, tedy schopnost si zapamatovat typ již dříve eliminovaného antigenu. Jak již bylo výše uvedeno, po prvním setkání organizmu s antigenem (primární obrana) je vyhrazena skupina T a B buněk, které zajišťují při následném opětovném setkání rychlejší produkci protilátek a tím i odezvu organizmu (sekundární obrana).

Tento princip je využíván při očkování, kdy je do organizmu dodána látka obsahující malé množství chorobných zárodků, což způsobí vytvoření si paměti na tento antigen a následně tedy rychlejší reakci při výskytu této choroby.

Algoritmus pozitivní selekce

V imunitním systému je principem pozitivní selekce odstranění takových lymfocytů, které nereagují na MHC komplex, tedy mají receptory buď poškozené nebo je nemají vůbec. Takovéto lymfocyty jsou totiž pro organizmus zbytečné a nijak by se nemohly podílet na obraně imunitního systému.

Výsledný algoritmus tedy pracuje tak, že nejprve vyhodnotí míru afinity jednotlivých prvků (analogicky v imunitním systému to znamená míru afinity mezi T buňkou a všeobecným MHC komplexem) a následně vybere jen ty prvky, u kterých je míra afinity větší jak definovaný práh vzájemné reakce. Zbytek populace je eliminován. Buňka, která projde úspěšně pozitivní selekcí, je schopna rozpoznat MHC molekuly a je připravena bojovat s antigeny v organizmu.



Obrázek 14 - Vývojový diagram pozitivní selekce

Algoritmus pozitivní selekce uvedený v [5]:

```
Function positive_selection (S, r, n)
S - množina řetězců, která definuje vlastní buňky („SELF“)
r - práh vzájemné reakce
n - počet detektorů k vygenerování
Výstupem je množina detektorů A

begin
  j ← 0
  while j <= n do                    //opakuj do n
    m ← rand(1,L)                    // vygenerování nového řetězce délky L
     for every s of S do
      aff ← affinity(m,s,r)          //urči afinitu prvků m s. oblast rozpoznávání r
      if aff >= r then               //pokud je afinita větší jak oblast rozpoznávání
        A ← insert(A, m)             // přidej tento prvek do množiny A
      endIf
    endFor
    j ← j+1
  endWhile
  return A
end
```

Slovní popis kroků algoritmu:

1) Inicializace prázdné množiny **A**, jenž na výstupu bude obsahovat jen takové prvky, jejichž míra afinity vůči některému detektoru ***s*** bude větší než práh vzájemné reakce.

2) Vygenerování náhodného řešení. Jak již bylo výše řečeno k vygenerování nového řešení (detektoru), se může použít čistě náhodný přístup nebo knihovna genů.

3) Pro takto vygenerované řešení se určí afinita vůči všem prvkům **s** z množiny **S** (SELF), přičemž se uvažuje ta nejvyšší z nich.

a) Je-li afinita řešení větší jak práh vzájemné detekce **r**, je toto řešení přidáno do výsledné množiny **A**.

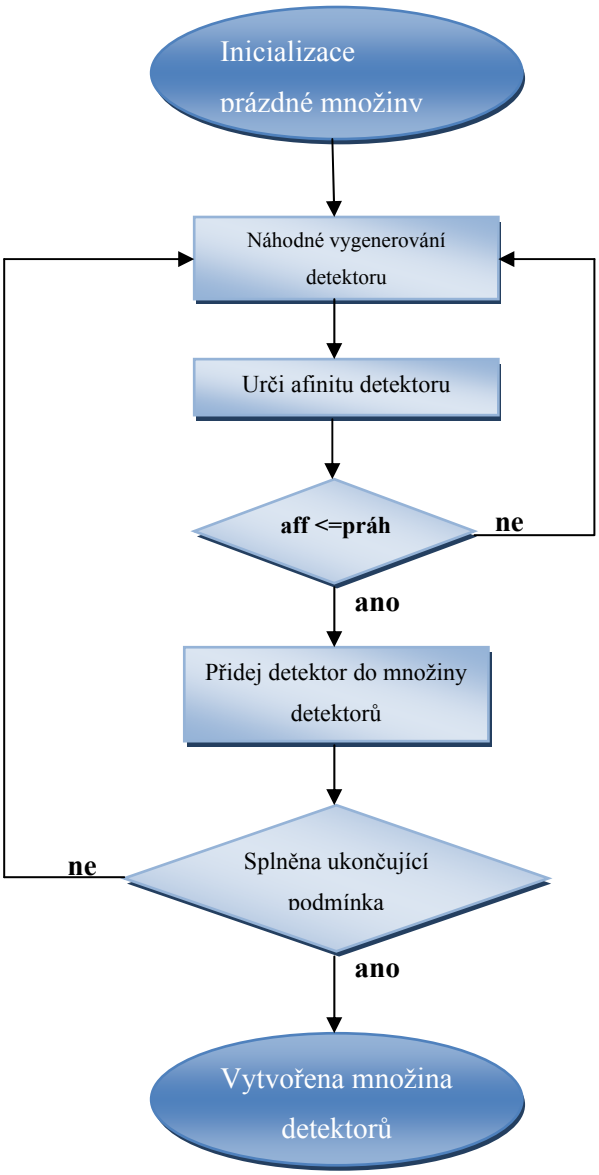
b) Jinak je řešení zahozeno.

4) Opakuj předchozí 2 kroky, dokud nebude výsledná množina **A** obsahovat **n** prvků.

Algoritmus negativní selekce

Algoritmus negativní selekce umožňuje organizmu udržovat kontrolu nad lymfocyty, citlivými na organizmu vlastní buňky, tj na antigeny obsahující MHC vlastního těla. Takové buňky, u kterých se tato vlastnost projeví, jsou eliminovány. Tím je zabezpečeno, že imunitní systém bude útočit pouze na buňky cizí tj. patogeny a buňky vlastní zůstanou imunitním systémem nepovšimnuty. Tento mechanismus je tedy základní obranou organizmu před tzv. autoimunitními chorobami.

Tento algoritmus je velice podobný algoritmu pozitivní selekce, má však změněno kritérium výběru detektoru, které vybírá pouze ty prvky, jež nereagují s množinou vlastních prvků.



Obrázek 15 - Vývojový diagram negativní selekce

Algoritmus negativní selekce uvedený v [5]:

```
Function negative_selection (S, r, n)
S - množina řetězců, která definuje vlastní buňky („SELF“)
r - práh vzájemné reakce
n - počet detektorů k vygenerování
Výstupem je množina detektorů A

begin
  j ← 0
  while j <= n do                  //opakuj do n
    m ← rand(1,L)                  // vygenerování nového řetězce délky L
     for every s of S do
      aff ← affinity(m,s,r)        //urči afinitu prvků m,s. oblast rozpoznávání r
      if aff <= r then             //pokud je afinita menší jak oblast rozpoznávání
        A ← insert(A, m)           // přidej tento prvek do množiny A
      endIf
    endFor
    j ← j+1
  endWhile
  return A
end
```

Slovní popis kroků algoritmu:

1) Inicializace prázdné množiny **A**, jenž na výstupu bude obsahovat jen takové prvky, jejichž míra afinity vůči některému detektoru ***s*** bude menší než práh vzájemné reakce. Tedy takové detektory, které nebudou schopny navázat na žádný prvek ***s*** z množiny vlastních prvků **S**.

2) Vygenerování náhodného řešení. K vygenerování nového řešení (detektoru) se může použít čistě náhodný přístup nebo knihovna genů.

3) Pro takto vygenerované řešení se určí afinita vůči všem prvkům ***s*** z množiny **S** (SELF), přičemž se uvažuje ta nejvyšší z nich.

    a) Je-li afinita řešení menší jak práh vzájemné detekce **r**, je toto řešení přidáno do výsledné množiny **A**.

    b) Jinak je řešení zahozeno.

4) Opakuj předchozí 2 kroky, dokud nebude výsledná množina **A** obsahovat **n** prvků.

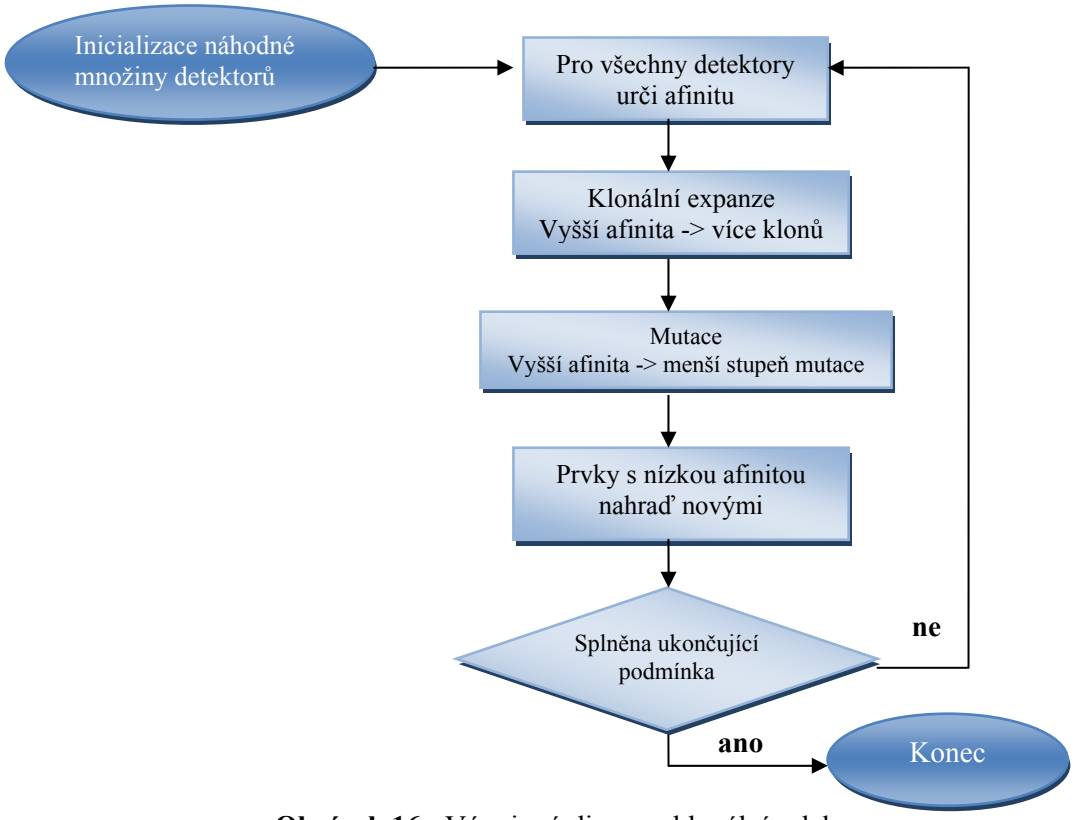
Uplatnění pozitivní a negativní selekce v biologii:

Jak již bylo řečeno, T-lymfocyty jsou tvořeny v kostní dřeni a následně putují do thymusu. Na tyto naivní T-lymfocyty je aplikována pozitivní selekce, aby byly zachovány pouze ty T-lymfocty, jejichž detektory jsou schopny detekce (jsou v pořádku a funkční). V thymusu dále dozrávají a procházejí negativní selekcí. To znamená, že zůstanou pouze ty, které jsou schopny navázat na cizí buňky (patogeny) a jsou odstraněny ty, které by se vázaly i na buňky vlastní.

Algoritmus klonální selekce

Při setkání organizmu s patogenem se začnou rozmnožovat pouze ty lymfocyty, které se dokážou na tento patogen navázat. Takovéto lymfocyty dostávají možnost hypermutovat a tím ještě zvyšovat svoji afinitu. Výsledkem je pak skupina lymfocytů, které mají schopnost rozpoznat daný patogen a i jemu blízké patogeny. Na tomto principu je postaven algoritmus klonální selekce.

Klonální algoritmus popisuje vznik nových detektorů, které jsou schopny reagovat na konkrétní problém. To pomáhá udržet množinu možných řešení co nejpřesnější a nejefektivnější. V praxi je využíván například při rozpoznávání znaků v obrazu, kde rozpoznání jednoho písmene je jeden z problémů v množině problému rozpoznání všech písmen.



Obrázek 16 - Vývojový diagram klonální selekce

Algoritmus klonální selekce uvedený v [5]:

Function clonal_selection (*S, g, N, n1, n2*)
S - množina řetězců, která definuje vlastní buňky („SELF")
G - počet iterací
N - velikost populace
n1 - počet prvků s vysokou afinitou, které budou vybrány pro klonování
n2 - počet prvků s vysokou afinitou, které budou uloženy do paměti **M**
Výstupem je **množina paměťových prvků *M***

```
begin
  j ← 0
  P ← rand(N,L)
  while j < g do
     for every s of S do
        for every p of P do
          aff ← match(s,P)
        end for
        P  ← sort(P,aff)
        P1 ← select(P,n1)
        for i<n1 do
          C ← clone(P1,aff(P1))
        end for
        for every c of C do
          C1 ← hypermut(c,inv(aff(P1)))
        end for
        for every c1 of C1 do
          aff(c1) ← afinity(c1,s)
        end for
        M1 ← sort(aff(C1))
        M(s) ← select(M1,1)
        M ← rand(n2,L)
        P ← replace (P,m,n2)
     end for
     j ← j+1
  end while
  return M
  end
```

Kroky algoritmu:

1. *Náhodná inicializace množiny P:* Množina P je množina detektorů (řešení), která po inicializaci obsahuje určitý počet náhodně vytvořených detektorů (řešení).

2. *Test řešení daného problému.* Pro každý problém z množiny problémů udělej:

   a) *Pro každý prvek z množiny P urči jeho afinitu:* Na daný problém aplikujeme postupně všechna řešení a určíme jejich afinitu.

   b) *Klonální expanze:* Vytvoříme kopie (klony) prvků v množině P podle jejich afinity. Čím větší je afinita (lepší řešení), tím více klonů tohoto prvku vytvoříme.

   c) *Mutace prvků:* Na všechny prvky z množiny P aplikujeme mutaci podle jejich afinity. Čím větší afinita, tím menší stupeň mutace. Jinými slovy, dobrá řešení pozměníme málo a špatná se změní více.

   d) *Nahraď určitý počet prvků s nízkou afinitou novými náhodnými prvky*.

3. *Pokud je dosaženo určitého kritéria afinity, ukonči algoritmus, jinak pokračuj krokem číslo 2*

3.2 Aplikace umělých imunitních systémů

V praxi se obyčejně při aplikaci výše uvedených principů imunitního systému nezachází příliš do detailu a využívají se spíše klíčové mechanizmy imunitního systému, které se však uplatňují v nejrůznějších doménách. Postupně se začínají objevovat příklady použití principů imunity například v rozpoznávání vzorů, počítačové bezpečnosti, dolování dat, strojovém učení či detekci chyb. Pokud není v následujícím textu uvedeno jinak, bylo čerpáno z [7], [8], [9].

3.2.1 Optimalizace

Proces optimalizace může být definován jako děj, během kterého se systém upravuje do co nejefektivnější a nejvíce funkční podoby. Nejčastěji se jedná o prohledávání prostoru parametrů systému a nalezení extrému.

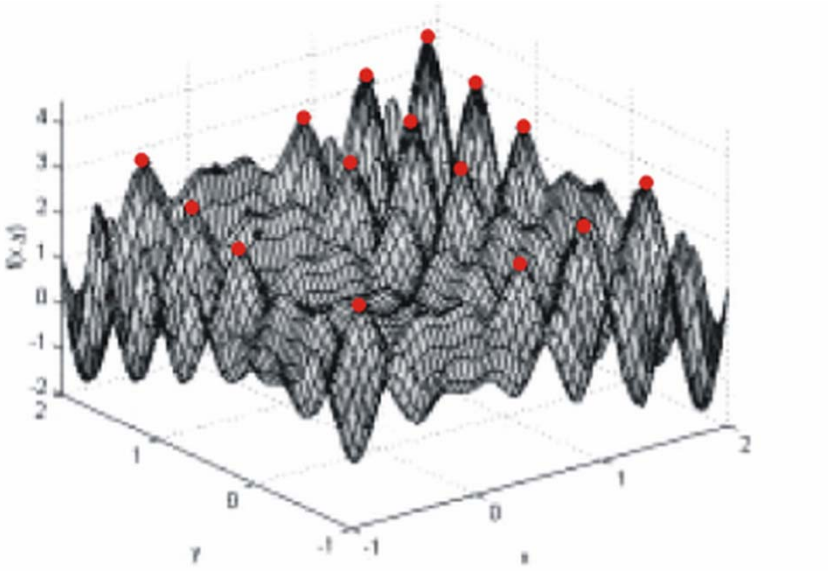
Optimalizace numerických funkcí

Autoři Bersini a Varela využili model imunitní sítě navržený v [14] a rozšířili jej. Prohledávací techniku optimalizace funkcí pak postavili na dvou vlastnostech a to senzitivitě sítě a metadynamice. Jejich přístup přinesl několik nových aspektů jako je kombinace afinity a fitness funkce, kdy fitness zodpovídá za kvalitu jedince vůči prostředí a afinita za míru podobnosti jedinců [15].

Multimodální optimalizace

Multimodální problémy jsou takové, které mají více než jedno lokální optimum. U těchto problémů jde o nalezení globálního optima a neustrnutí v některém z lokálních. Experiment autorů v [8] popisuje úlohu multimodální optimalizace funkce:

$$f(x, y) = x\sin(4\pi x) - y\sin(4\pi y + \pi) + 1$$



Obrázek 17 - Výsledek prohledávání po 100 generacích. Převzato z [15].

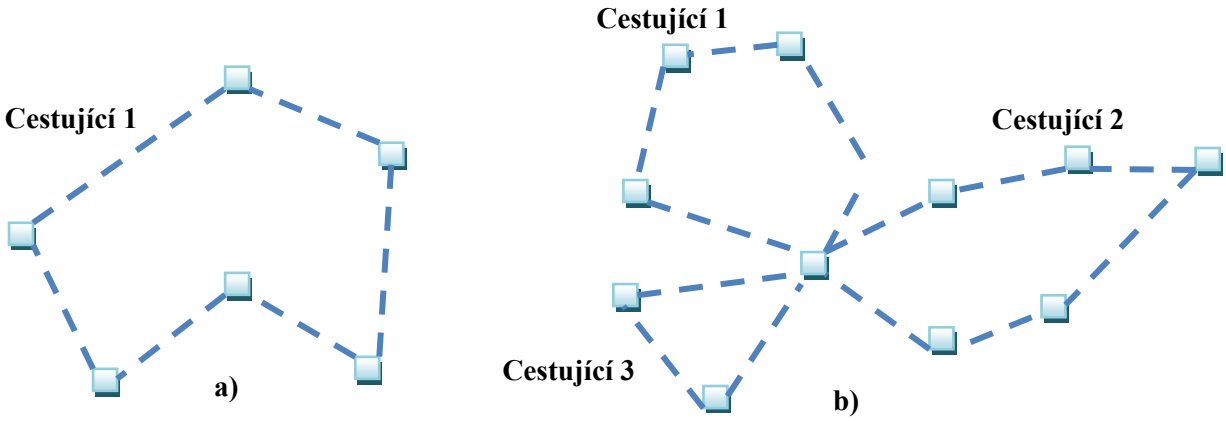
Problém *n* obchodních cestujících

Samotný problém obchodního cestujícího lze popsat takto:
Na mapě je zvoleno N měst a jejich vzájemná vzdálenost (popřípadě cena cesty) pro každou z dvojic. Problémem obchodního cestujícího je určit takové pořadí návštěv jednotlivých měst, aby každé město bylo navštíveno právě jednou, výsledná délka trasy byla nejkratší (nejlevnější) a cestující se vrátil zpět do města, kde cestu započal.

Matematická formulace používající pojmosloví teorie grafů zní: *Jak v daném ohodnoceném úplném grafu efektivně najít nejkratší hamiltonovskou kružnici?*

Tento problém náleží do kategorie NP-úplných problémů, pro které je typická výpočetní náročnost. Pro tento případ vždy existuje nějaké ideální řešení (případně více, ale se stejným výsledkem), které můžeme lehce najít tak, že prostě spočítáme délku (cenu) všech možných cest a vybereme tu nejkratší. Již pro malá N je však počet takovýchto rovnic vysoký a roste exponenciálně. Pro řešení tohoto problému se často používá genetický algoritmus, který nemusí nalézt vždy v konečném počtu kroků správné řešení ale pouze tzv. suboptimální, které je však většinou dostačující.

Případ *n* obchodních cestujících je rozšířením kombinatorického problému obchodního cestujícího.



Obrázek 18 – a) Problém obchodního cestujícího, b) Problém n obchodních cestujících

Pro tento problém navrhli autoři v publikaci [16] adaptivní optimalizační algoritmus inspirovaný teorií sítě a MHC komplexem. Tabulka 3 popisuje mapování mezi problémem *n* obchodních cestujících a imunitním systémem.

Imunitní systém	Problém n obchodních cestujících
Antigen	Obsahuje informace o městech a cestujících
Makrofág	Vybírá číslo města, které cestující musí navštívit
T-buňka	Napomáhá aktivaci B-buňky
B-buňka	Produkuje protilátky
Protilátky	Chování agenta

Tabulka 3 - Mapování mezi problémem *n* obchodních cestujících a imunitním systémem. Převzato z [16].

Rozpoznávání vzorů

Autoři de Castro a von Zuben navrhli algoritmus na rozpoznávání vzorů a optimalizaci funkcí. Jedná se o upravený klonální selekční algoritmus, jež pojmenovali CLONALG [8]. Množina vlastních prvků (SELF) je v případě rozpoznávání vzorů množina hledaných prvků a v případě optimalizace je vlastní prvek množiny funkce, kterou se snažíme zoptimalizovat.

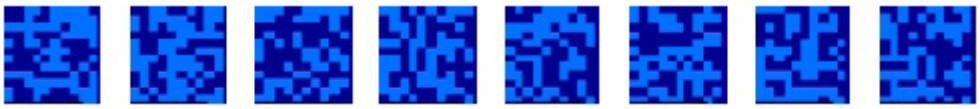
Algoritmus se dá shrnout do následujících kroků:

1) Vygenerování populace P možných řešení skládajících se z paměťových buněk M. Do nové generace jsou zahrnuty nově náhodně vytvořené prvky + prvky z paměťových buněk. P = P_{random} + M.
2) Na základě afinity je vybráno *n* nejlepších jedinců
3) Klonování vybraných *n* nejlepších jedinců
4) Mutace
5) Výběr *m* jedinců pro nahrazení nejslabších jedinců v populaci P
6) Náhrada protilátek novými. Pravděpodobnost náhrady se zvyšuje s klesající mírou afinity.

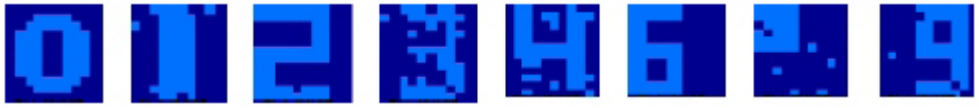
Příklad použití rozpoznávání vzorů

V publikaci [8] nastínili autoři využití algoritmu rozpoznávání vzorů na problému rozpoznávání binárních vzorů. Každý obrázek má délku L=120. Míra afinity je určena jako Hammingova vzdálenost D mezi protilátkou a antigenem. Populace se skládá z 10 protilátek, z nichž 8 protilátek tvoří paměť.

Obrázek 19 - Vstupní vzory. Převzato z [8].



Obrázek 20 - Populace po inicializaci. Převzato z [8].



Obrázek 21 - 100tá generace. Převzato z [8].

3.2.2 Další aplikace v praxi

Chůze robota

V oblasti robotiky byla řešena otázka chůze robota. Pro vyřešení problému chůze 6ti-nohého robota, který by dokázal chodit, přičemž by pohyboval každou nohou zvlášť a to sehraně s ostatními nohami, za rozložení váhy na všechny zbylé nohy na zemi. Autoři v publikaci [17] použili variantu druhé generace imunitní sítě popsanou v [14]. V tomto řešení má robot 6 buněk zodpovědných za pohyb každé nohy s těmito parametry:

- Koncentrace
- Horní práh
- Dolní práh

Aby robot pohnul svoji nohou vpřed, přičemž se s ní nedotýká země, musí parametr koncentrace přesáhnout přes horní práh. Naopak pokud koncentrace klesne pod dolní práh, robot ponechá svoji nohu na zemi, pohne s ní zpět a původní buňka je nahrazena novou [15].

4 Počítačová bezpečnost

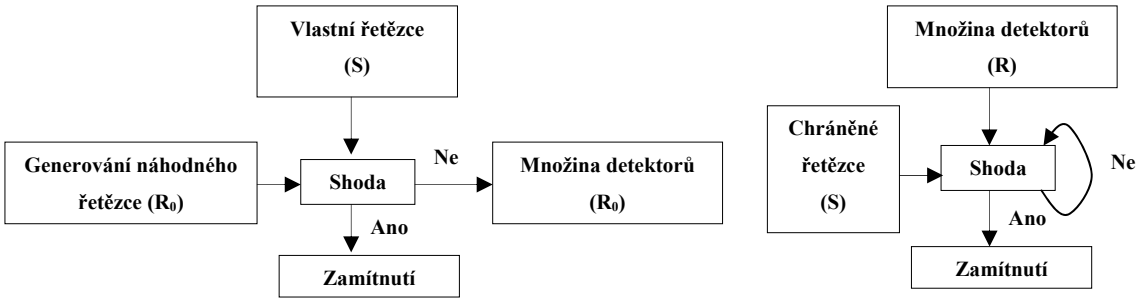
Oblast počítačové bezpečnosti se jako vůbec jedna z prvních začala intuitivně zajímat o mechanizmy imunitního systému. Z výzkumů imunitního systému je jasně patrná jeho úloha a to rozpoznávání a eliminace infekčních mikroorganismů. Hlavním kladem je jeho pozoruhodná schopnost dynamické adaptace na předtím dosud nepoznané infekce.

Výzkum umělých imunitních systému tak zasáhl do dvou významných částí počítačové bezpečnosti: detekce průniků do sítě (neautorizovaný uživatelský přístup), eliminace počítačových virů a červů[4]. Hlavní myšlenkou prací zabývajících se touto problematikou je, zda a jak může být myšlenka přírodního obranného mechanismu převedena do počítačové domény. Tento obor zájmu se stává stále častějším středem pozornosti a neustále zde probíhá bouřlivý výzkum. Přibližme si tedy tuto problematiku a některé práce v této oblasti.

[4] Konkrétní popis rozdílu mezi počítačovým virem a červem a další názvosloví z počítačové bezpečnosti viz server viry.cz

4.1 Principy imunitního systému používané v počítačové bezpečnosti

Forrest a kolektiv v práci [23] přirovnává problém ochrany počítačových systémů k učení rozlišení mezi *vlastními* a *nevlastními* prvky. Popisují strategii detekce změn založené na negativní selekci, která je vlastní pro imunitní systém. Jejich algoritmus běží ve dvou fázích. V první je generována množina detektorů a v druhé fázi jsou chráněná data monitorována porovnáváním s vygenerovanými detektory. Princip popisuje obrázek 22.



Obrázek 22 - a) Generování platných detektorů b) Monitorování chráněných dat

Pokud počet společných bezprostředně následujících bitů dvou řetězců je větší nebo rovno *r*, je prohlášena shoda. Autoři také prezentovali rovnici odhadující pravděpodobnost výskytu shody na méně jak *r*-bezprostředně následujících pozicích ve dvou náhodných řetězcích a rovnici pro určení pravděpodobnosti detekce změny pro různé konfigurace systému.

Mezi hlavní vypozorované vlastnosti algoritmu patří:

- vztah pravděpodobnosti detekce,
- počet detektorů nemusí nutně růst s počtem chráněných řetězců,
- pravděpodobnost detekce exponenciálně stoupá s počtem nezávislých detekčních algoritmů,
- detekce je symetrická,
- je zde exponenciální vztah mezi generováním detektorů a velikostí množiny vlastních řetězců, což je nežádoucí.

Pro odstranění tohoto nedostatku představil D'haeseleer [24] nový algoritmus generování detektorů s lineární složitostí s ohledem na velikost množiny vlastních řetězců. Algoritmus je rozdělen na dvě fáze. V první je počítán rekurentní problém počtu neshodných řetězců v množině S.

Druhá zůstává v podstatě nezměněna, jen s tím rozdílem, že se snaží vybírat co nejvíce od sebe vzdálené detektory, s cílem lepšího pokrytí prostoru řetězců se stejným počtem detektorů.

Tyto principy byly použity v implementované aplikaci pro názornou ukázku možnosti využití umělých imunitních sítí v počítačové bezpečnosti.

4.2 Detekce a eliminace virů

Většina současných antivirových aplikací identifikují vir na základě detekce určitého řetězce (otisku), který je unikátní pro tento konkrétní vir, popřípadě využívají techniky detekce viru podle jeho specifického chování. Z toho je patrno, že tyto znalosti o virech musejí být neustále pravidelně zdokonalovány a obnovovány. Naproti tomu antivirový program založený na imunitních systémech by tento problém neměl. Naskytuje se však otázka, jak korektně namapovat principy imunitního systému na antivirový program, který by dokázal reagovat na nové, neznámé viry a přitom předcházel zbytečným poplachům. S myšlenkou takovéhoto imunitního antivirového programu přišli autoři Kephart, Sorkin a Swimmer v publikaci [10].

V práci [7] se autoři zabývali myšlenkou, že každá aplikace infikovaná virem, může být zpětně upravena na svůj originál tj. neinfikovanou podobu. Jakmile totiž vnese virus do programu své funkce, nejlepším způsobem jak se vyhnout okamžité detekci, je zachovat činnost hostitelské aplikace. To znamená, že virus musí být schopný rekonstruovat hostitelský program (žádný z původních bytů aplikace není zničený). V této skupině virů, tj. virů nepřepisujících, platí, že infikovaná aplikace A' je reversní transformací A. Tato transformace se nezachovává při legitimní změně aplikace (update). Důraz byl hlavně kladen na mechanizmy rozpoznávání virů a adaptibilitu na viry nové. V jejich práci také zdůrazňují potřebu chránit nejen konkrétní počítač, ale i sdílet informace o virech přes celý internet.

O navržení nového přístupu k problematice bezpečnosti inspirované imunitním systémem se pokusil ve své práci [22] J.O.Kephart, jenž navrhl umělý imunitní systém schopný vyvíjet protilátky na předtím neznámý počítačový virus. Takovéto protilátky jsou pak schopny extrahovat informaci z tohoto viru a zapamatovat si jej, pro rychlejší reakci na jeho případnou budoucí opětovnou infekci (v biologii známé jako sekundární imunitní odpověď). Autor se zaměřil na minimalizaci tzv. autoimunitní odpovědi, což je mylná reakce imunitního systému, kdy je legitimní software identifikován jako škodlivý.

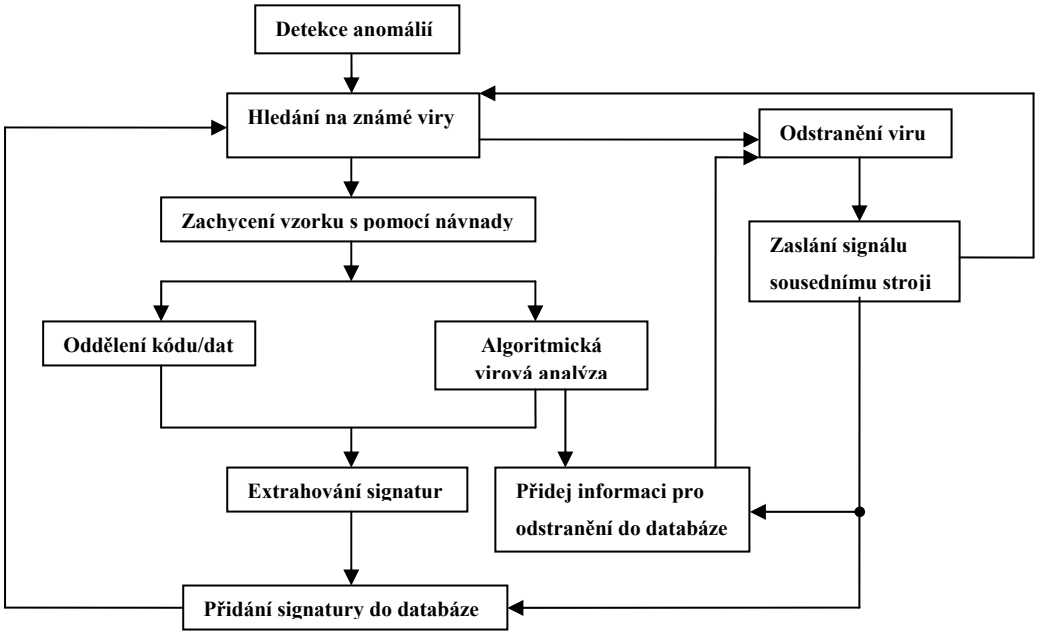
Rozpoznání viru se provádí pomocí přesného nebo alespoň částečného rozpoznání relativně krátké sekvence bytů nacházející se ve viru nazývané signatura. Proces, dle kterého navržený imunitní systém stanoví, zda je daný software infikován, je rozdělen na dvě fáze.

Monitor integrity, který slouží ke sledování rozdílů mezi originální a aktuální verzí programů a dat (vlastní prvky). Avšak pro vyvolání imunitní odpovědi, zde důkaz o nevlastním prvku v systému

není dostatečný. Proto je zapojen mechanismus, znající předem některé z virů a na základě heuristiky rozhodne, zda tato anomálie bude označena za vir. To redukuje výskyt autoimunitní odpovědi.

Byly také vytvořeny programy, které sloužily jako návnada na vybrané viry. Takto nakažené programy byly následně zpracovány komponentou imunitního systému nazvanou extraktor signatury, aby mohli být později identifikovány.

Jelikož je na počítačový imunitní systém ještě navíc kladen požadavek, který v biologickém imunitním systému není .Mít možnost se pokusit o extrahování informace o tom, jak je na danou návnadu vir připojen, aby mohl být infikovaný jedinec opraven. Je vytvořen ke každému viru rozpoznávající a opravující algoritmus. Po detekci viru jedním počítačem, je vyvolána vlna signálů k jeho zničení, propagovaná po cestě tohoto viru. Celkový princip shrnuje obrázek 23.



Obrázek 23 - Hlavní komponenty a jejich vztahy počítačového imunitního systému.

Autoři studie [25] zase přišli s myšlenkou vývoje počítačového imunitního systému za využití inspirace v imunitních systémech obratlovců. Popsali zde několik možných návrhů systému, založeném na přímém mapování mezi komponentami imunitního systému a počítačovou architekturou jak ukazuje tabulka 4. Navzdory přitažlivému přirozenému zdroji inspirace, bylo odhaleno 5 problémů, jež jsou součástí počítačového imunitního systému, avšak v biologickém nejsou přímo zakomponovány. Jsou to: utajení, integrita, dostupnost, zodpovědnost a korektnost. Argumentovali tím, že imunitní systém obratlovců je primárně zaměřen na přežití, z pohledu kombinace integrity a dostupnosti. Tyto problémy byly označeny, jako možná limitace užití tohoto mapování.

Imunitní systém	Prostředí sítě
Vlastní prvek	neporušená data
Nevlastní prvek	nějaké změna vlastního prvek
Ve smyslu ochrany aktivního procesu na jednom hostiteli	
Buňka	aktivní proces v počítači
Vícebuněčný organismus	spuštěno více procesů
Populace organismů	množina počítačů v síti
Kůže a vrozená imunita	Bezpečnostní mechanismus (hesla,skupiny,oprávnění)
Adaptivní imunita	Proces schopný vyhledat abnormální chování
Autoimunitní odpověď	Falešný alarm
Vlastní prvek	Normální chování
Nevlastní prvek	Abnormální chování

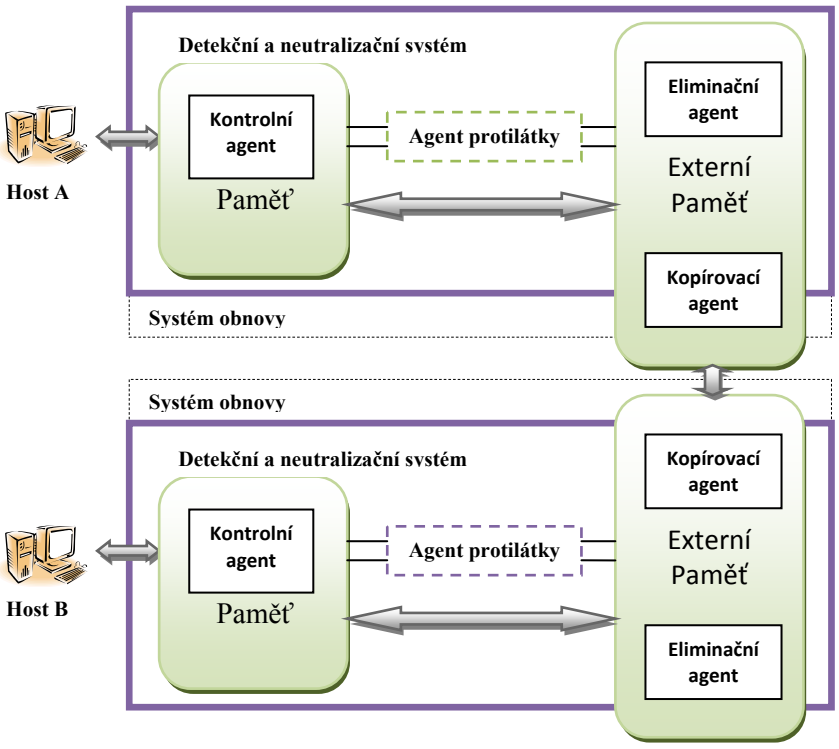
Tabulka 4 - Mapování mezi komponentami imunitního systému a počítačovou architekturou.

Detekce virů založená na agentech[5]

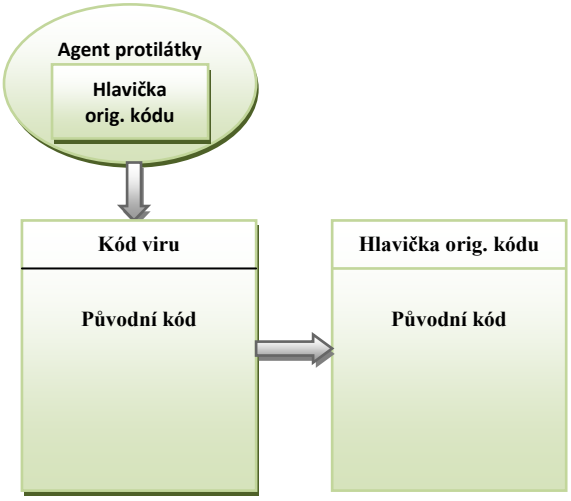
Distribuovaný přístup k detekci a neutralizaci autonomními a heterogenními agenty představili Okamoto & Ishida [26]. Tento systém detekuje viry porovnáním *vlastních* informací (jako několik prvních bajtů hlavičky souboru, velikost soubor, cesta, atd.) vzhledem k aktuálním hostitelským souborům. Viry jsou neutralizovány přepisem *vlastních* informací o infikovaných souborech a obnovení je dosažené zkopírováním stejného souboru z jiných neinfikovaných hostitelů přes počítačovou síť. Celý systém a mechanismus neutralizace virů popisuje obrázek 24 a předpokládá existenci následujících agentů:

- Detekční a neutralizační agent nazvané agenty protilátky.
- Eliminační agenti zodpovědné za odstranění změněných souborů, které byly neutralizovány agenty protilátek.
- Kopírovací agenti, kteří odesílají a přijímají neinfikované soubory přes počítačovou síť.
- Kontrolní agenti zodpovědné za kontrolní proces antiviru.

[5] Agentem může být chápáno cokoliv, co vnímá své prostředí pomocí svých senzorů a ovlivňuje jej pomocí svých efektorů. Imunitní systém je složený z obrovského množství buněk, molekul a orgánů, které mohou být viděny, jako imunní agenti distribuované všude v našem těle.



a)



b)

Obrázek 24 - a) Diagram antivirového systému b) Neutralizace viru agentem protilátky

Hlavním nedostatkem jejich přístupu je rozhodnutí o velikosti hlavičky, kterou využívá agent protilátky (příliš velká hlavička by zabírala hodně paměti, zatímco příliš malá by mohla být neúčinná) a obnovení přes síť by mohlo způsobit deadlock nebo sekundární infekci.

4.3 Detekce průniků do sítě

V současné době většina vyvíjených softwarových řešení na detekci průniku do počítačové sítě využívá k detekci rozpoznání specifického chování (vzoru chování) nebezpečného softwaru. Tj. například specifická adresace spojení (škodlivá aplikace naslouchá na daném portu a odesílá soukromá data na určitou IP adresu). Předpokladem k odhalení útoku je tedy znalost chování popřípadě otisku tohoto útoku, podobně jako u rozpoznávání virů. Většina komerčních detektorů průniku využívá tzv. znalostní databáze, ve kterých může uživatel přiřadit ke každému takovémuto vzoru chování pravidlo, podle kterého se bude řídit povolení, popřípadě omezení přístupu [11]. Jak je patrno, tento přístup je velice statický a velmi vázán na zadaná pravidla. Pokud před takovýto ochranný software postavíme útočníka, který se dokáže alespoň částečně přizpůsobovat a měnit své chování v čase, jen stěží se s ním vypořádá. Dalším možným řešením je **detekce anomálií.**

Detekce anomálií

Normální chování systému bývá často charakterizováno sérií pozorování v čase. Na problém detekcí nového chování nebo anomálií v systému bývá často nahlíženo jako nalezení odchylek v charakteristických vlastnostech systému. V tomto případě se nejdříve zachytává běžný provoz na síti a následně jakákoliv jiná aktivita na síti je detekována jako abnormální a je označena za útok. Tento přístup se tedy může vypořádat i s dynamicky se měnícími formami útočníka a učit se.

O popsání analogie mezi přírodním imunitním systémem a systémem pro ochranu proti vniknutí do počítačové sítě se pokusili autoři Kim a Bentley [12]. Zaměřili se na významné vlastnosti imunitního systém, které by mohly být úspěšně aplikovány na úlohu detekce průniku do počítačové sítě. V jejich analýze identifikovali 3 základní požadavky na design systému pro detekci průniku do počítačové sítě: **distribuce**, **samoorganizace** a **lehkost**.

Charakteristika SDP	Imunitní systém
Distribuce	Imunitní síť
	Unikátní množina protilátek
Samoorganizace	Vývoj genové knihovny
	Negativní selekce
	Klonální selekce
Lehkost	Přibližné vazby
	Paměťové buňky
	Vyjádření genu

Tabulka 5 - Prvky a funkce imunitního systému, které splňují uvedené 3 požadavky na vývoj systému detekce průniku (SDP).

Prostředí sítě	*Imunitní systém*
Primární SDP	Kostní dřeň a thymus
Lokální hosté	Sekundární lymfatické uzly
Detektory	Protilátky
Průniky do sítě	Antigeny
Normální aktivita	Vlastní prvky
Abnormální aktivita	Nevlastní prvky

Tabulka 6 - Vztah mezi prostředím sítě a imunitním systémem.

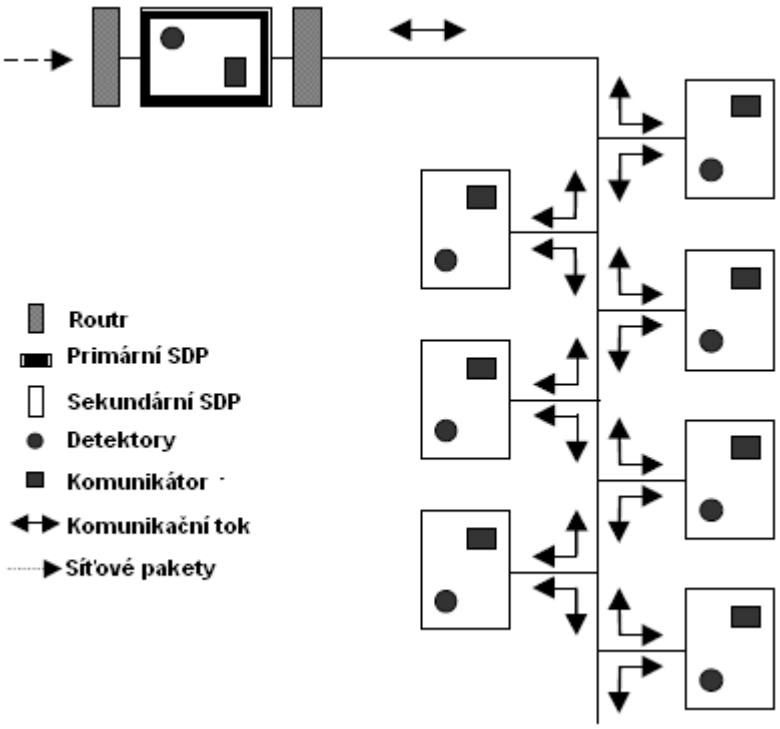
Následně navrhli, aby celkový systém pro detekci průniků do sítě zahrnoval 3 evoluční stupně:

1) negativní selekce

2) klonální selekce

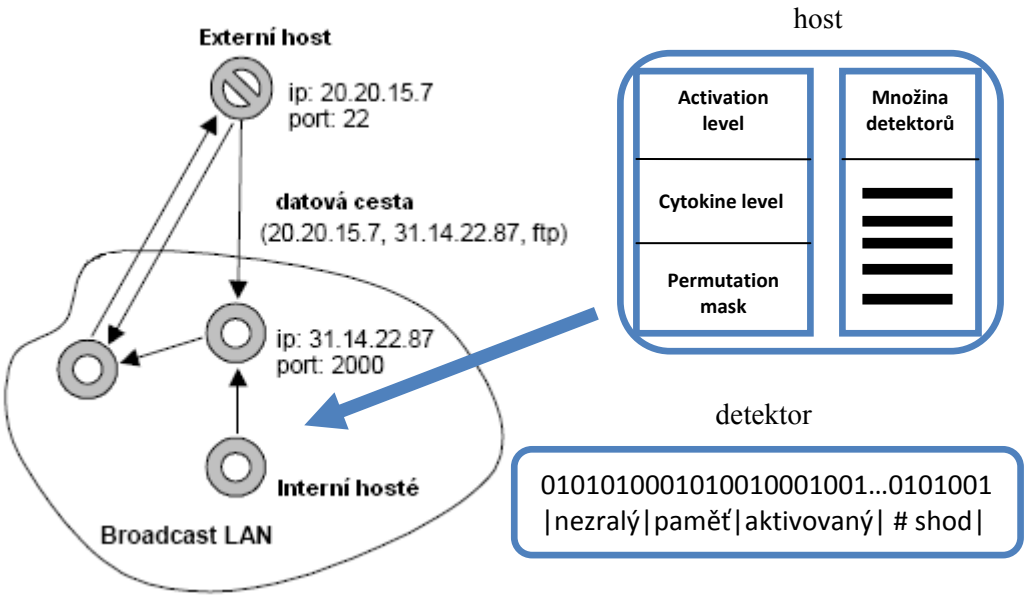
3) evoluce knihovny genů

V algoritmu negativní selekce nahrazuje náhodné generovaní detektorů evoluce nevlastních prvků.



Obrázek 25 - Fyzická realizace systému pro detekci průniků. Převzato z [8].

Velice zajímavým výzkumem použití imunitní sítě v bezpečnosti počítačové sítě se zabývali Forrest ve své práci [25] a Hofmeyr a Forrrest v [27]. V jejich systému bylo několik imunitních buněk a molekul zjednodušeno na definici základních typů detektorů, které kombinují užitečné vlastnosti těchto elementů. Detektory jsou reprezentovány řetězci bitů dané délky a několika stavy. Detekce je provedena porovnáním dvou řetězců, přičemž jejich shoda je určena při shodě *r* po sobě bezprostředně následujících bitů na odpovídajících si pozicích. Tento přístup byl již použit v práci [23], kterou představil Forrest již v roce 1994. Za dozrávání naivních detektorů na paměťové, společně s negativní selekcí, je zodpovědná část pro učení systému. Pro uchování diverzity detektorů byla použita permutační maska. Model systému popisuje obrázek 26.



Obrázek 26 - Model systému dle autorů Hofmeyr a Forrrest.

Stejně jako lze v ochraně proti virům využít softwarové agenty je to možné i v ochraně sítí. Touto možností se zabývá odborné pojednání [13], které mluví o možnosti využít pro detekci anomálií v počítačové síti mobilní softwarové agenty, kteří se dokáží pohybovat v uzlech monitorované sítě, jsou schopni interreagovat na své okolí a komunikovat mezi sebou. Byly zde navrženy i parametry pro sledování při detekci anomálií: práva a typ uživatele, typ připojení do sítě, velikost volné paměti, četnost a poloha přihlášení a odhlášení na síť.

5 Implementace programu pro optimalizaci multimodálních úloh

Implementace programu AIS pro optimalizaci multimodálních úloh je založena na algoritmu popsaném v publikaci [21] a nazvaném opt-aiNet. Stejně jako evoluční algoritmy je opt-aiNet algoritmus adaptivní optimalizační technika. S každou iterací se řešení postupně zlepšuje, dokud není nalezeno lokální nebo globální optimum. Algoritmus je schopen nalézt nejen globální optimum, ale také lokální optima, která mohou být také zajímavá.

5.1 Popis algoritmu

Shrňme si nejdříve jednotlivé komponenty použité v algoritmu:

Buňka sítě (Network Cell): Buňka sítě *c* představuje pravděpodobné řešení daného problému. Každá takováto buňka je reprezentována vektorem reálných hodnot $c \in IR^n$.

Populace (Population): Populace $P_t = \{c_{0,\ldots\ldots,}c_{|Pt-1|}\}$ je množina všech buněk sítě v daném čase *t*.

Fitness: Fitness funkce popisuje kvalitu daného řešení (buňky sítě). V případě hledání maxima, či minima, může být hodnota fitness odvozena přímo od vlastní definice funkce. $\overline{F}_t$ je průměrná hodnota fitness populace P_t v daném čase *t*.

Affinita (Affinity): Míra podobnosti je dána eukleidovskou vzdáleností mezi dvěma buňkami.

Klon (Clone): Klon je identická kopie rodičovské buňky sítě.

Algoritmus opt-aiNet:

(1) $t=0$
(2) P_0=náhodná populace o velikosti N_e
(3) $\overline{F_0} = \frac{1}{|P_0|}\sum_{i=0}^{|P_0|-1} fitness(c_i)$ //průměrná fitness
(4) **while** *not abort* **do**
(5) **repeat**
(6) t=t+1
(7) $P_t=P_{t-1}$
(8) **for** i ∈ {0,…,$|P_t|$-1} **do**
(9) **for** j ∈ {1,…,n_c} **do** //pro vytvářený počet klonů
(10) mc = hypermutate(c_i,fitness(c_i)) //vytvoř klon
(11) **if** fitness(mc) > fitness(c_i) **then** //pokud má nový klon lepší fitness
(12) c_i=mc // nahradí rodiče
(13) **end if**
(14) **end for**
(15) **end for**
(16) $\overline{F_0} = \frac{1}{|P_0|}\sum_{i=0}^{|P_0|-1} fitness(c_i)$
(17) **until** $\overline{F_t} < \overline{F_{t-1}} + \sigma_f$ //opakuj, dokud nárůst fitness v populaci není menší jak σ_f
(18) i=0
(19) P_{mem}={c_o}
(20) **for** j ∈ {0,…,$|P_t|$-1} **do** //pro všechny prvky v populaci
(21) k=0
(22) **repeat**
(23) **if** affinity(c_j,c_k) < σ_s //pokud jsou si dvě buňky podobné víc, než stanoví práh σ_s
(24) **if** fitness(c_j) > fitness(c_k)
(25) nahraď c_k v P_{mem} c_j //buňka s lepší fitness bude zachována
(26) **end if**
(27) **else**
(28) k=k+1
(29) **end if**
(30) **until** (k > i) **or** (affinity(c_j,c_k)< σ_s)
(31) **if** k > i
(32) $P_{mem}=P_{mem}\cup${c_j}
(33) i=i+1
(34) **end if**
(35) **end for**
(36) $P_t=P_{mem}$
(37) P_t=přidej($|P_{mem}|$*d) náhodně vygenerovaných buněk //doplň populaci
(38) **end while**

V tomto algoritmu je analogií k hledanému optimu - protilátka a k síťové buňce - B-buňka imunitního systému, fitness hodnota buňky v síti je míra podobnosti k protilátce (optimum).
Algoritmus kombinuje lokální prohledávání v krocích (5)-(17) a globální prohledávání v krocích (18)-(35).
Během lokálního prohledávání je fitness hodnota aktuálních buněk v síti postupně zlepšována, dokud nedosáhne předem stanovené hodnoty prahu σ_f (**σf - fitness threshold**) tj. dokud navýšení oproti předchozí iteraci není menší jak σ_f.

$$\sigma_{f(i)} = average\_fitness(i-1) - average\_fitness(i)$$

Mutace v kroku (10) je dána vztahem:

$$mc = c + \frac{N}{\beta} e^{-f_n(c)}$$

Kde *mc* je nová buňka sítě vzniklá mutací původní buňky *c*. N je náhodné číslo gaussova rozložení s nulovým základem a standardní odchylkou σ=1.
β je **d- factor newcommers** a $f_n(x)$ je normalizovaná hodnota fitness buňky c definována vztahem:

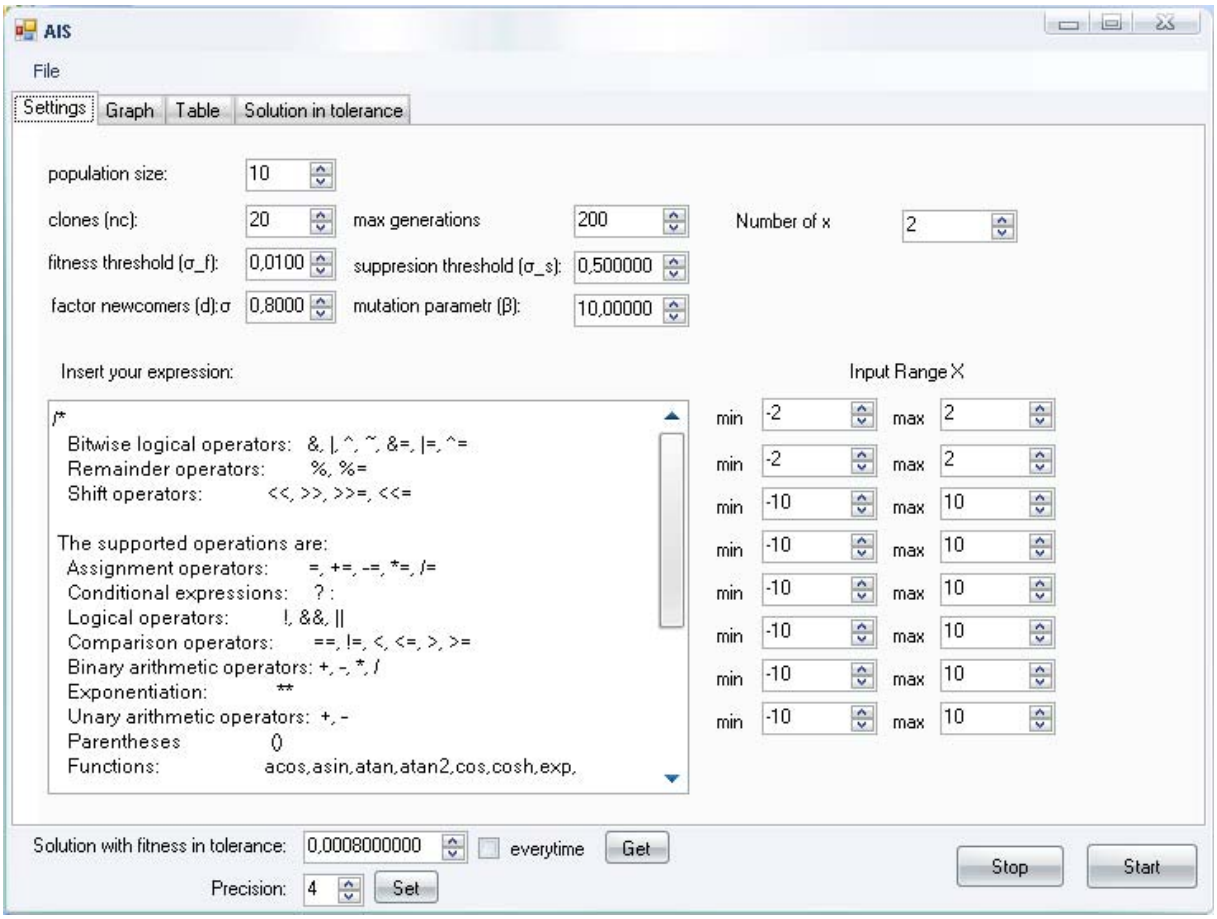
$$f_n(c) = \frac{f(c) - \min_{c' \in P}(f(c'))}{\max_{c' \in P}(f(c')) - \min_{c' \in P}(f(c'))}; \quad f(c): \ hodnota\ fitness$$

Pokud je fitness hodnota mutovaného klonu lepší jak hodnota rodičovské buňky, rodičovská buňka je jím nahrazena – provedeno v krocích algoritmu (11)-(13). Tento mechanismus popisuje klonální selekci v lokálním prohledávání.
Proces potlačení v krocích (23)-(26) popisuje klonální selekci v globálním prostoru. Pokud je rozlišnost dvou buněk menší jak stanovený práh σ_s (**σs - suppresion threshold**) (jsou si více podobné, než stanovuje předem daný práh σ_f), lepší buňka z nich (dle fitness) bude představovat paměť do další populace. Míra afinity je dána jako eukleidovská vzdálenost mezi dvěma buňkami.

V kroku (37) je populace doplněna o $d * |P_{mem}|$, kde *d* je daný parametr (**d- factor newcommers**) a $|P_{mem}|$ je velikost populace představující paměť.

5.2 Popis vlastní aplikace

Aplikace obsahuje několik záložek, ve kterých lze definovat vstupní parametry aplikace, ale také sledovat průběh výpočtu. Představme si tedy jednotlivé možnosti.



Obrázek 27 – Implementovaná aplikace pro optimalizaci multimodálních úloh.

5.2.1 Nastavení programu

Nastavení programu se provádí v záložce Settings. Zde je možno nastavit následující parametry:

N - population size: Udává velikost počáteční populace buněk (potenciálních řešení).

Nc - clones: Určuje počet klonů, které se budou vytvářet během klonální selekce z každé buňky v populaci, následně se nechají mutovat a bude z nich vybrán nejlepší klon, který v případě, že bude lepší než rodičovská buňka, ji nahradí.

σ_f - fitness threshold: Lokální selekce postupně zvyšuje průměrnou fitness celé populace, dokud navýšení oproti předchozí iteraci není menší jak zadaný práh fitness threshold.

d- factor newcommers: Populace je na konci cyklu algoritmu doplněna o $d * |P_{mem}|$, kde d je daný parametr (factor newcommers) a $|P_{mem}|$ je velikost populace představující paměť.

Max - max generation: Udává maximální počet generací algoritmu.

σ_s - suppresion threshold: Algoritmus porovnává jednotlivé buňky sítě (řešení) mezi sebou a pokud je jejich rozlišnost menší jak stanovený práh, je pouze ta buňka (řešení), která má větší fitness, použita jako paměťová (tj. do další generace).

β - mutation parametr: Pro výpočet nového klonu je použita rovnice (1) a tento parametr je dosazen do této rovnice jako β.

Jelikož program umožňuje zadání multimodální funkce jako textový výraz, který je pak vyhodnocen, jsou zde ještě další vstupní pole a parametry:

pole pro zadání funkce: Zde je možné libovolně používat operátory, definované funkce a závorky pro zadání požadované funkce. Předpokládá se výraz typu y = funkce ;
Kde funkce je výraz pro danou mutlimodální funkci. Pro použití proměnné ve výrazu (hledáme hodnotu této proměnné) se předpokládá použití proměnné s názvem $x_0.....x_{n-1}$, kde n je počet proměnných ve výrazu. Je zde také možno používat komentáře ve stejné podobě a významu jako v jazyku C. Pokud je pole prázdné, je použita pro výpočet fitness rovnice zadaná ve zdrojovém kódu ve funkci eval v souboru Optim_function.cpp. Pro složitější výpočty se doporučuje využít tuto možnost (přeložený kód je rychlejší jak vyhodnocování textové funkce).

count of variables: Udává použitý počet proměnných ve výrazu multimodální funkce udané v poli pro zadání funkce.

input range x: Dimenze hodnot, kterých mohou nabývat hledané proměnné $x_0.....x_{n-1}$.

Veškeré tyto parametry lze ukládat do konfiguračního souboru aplikace a následně znovu načítat. Tento konfigurační soubor má koncovku .ais a je uložen v textové podobě. Parametry, které se neukládají do tohoto souboru, jsou:

Solution with fitness in tolerance – tento parametr je popsán dále v kapitole 4.2.1.4.

Precision – tímto parametrem lze nastavit, kdy již bude hodnota buňky zaokrouhlena. Například při nastavení precision 4, bude hodnota buňky zaokrouhlena při její vzdálenosti od zaokrouhlené hodnoty menší jak 0,0001 tzn. Pokud bude například hodnota buňky 0,12000345 a 1,100001234, bude tato buňka zaokrouhlena na 0,12000345 a 1,1. Veškeré hodnoty buňky, jejichž vzdálenost od zaokrouhlené hodnoty je větší jak 0,0001, budou nezměněny.

5.2.2 Graf fitness populace

Program obsahuje záložku Graph, v níž je umístěn graf znázorňující nejhorší, průměrnou a nejlepší fitness řešení pro danou generaci.

Obrázek 28 - Graf fitness populace v programu AIS

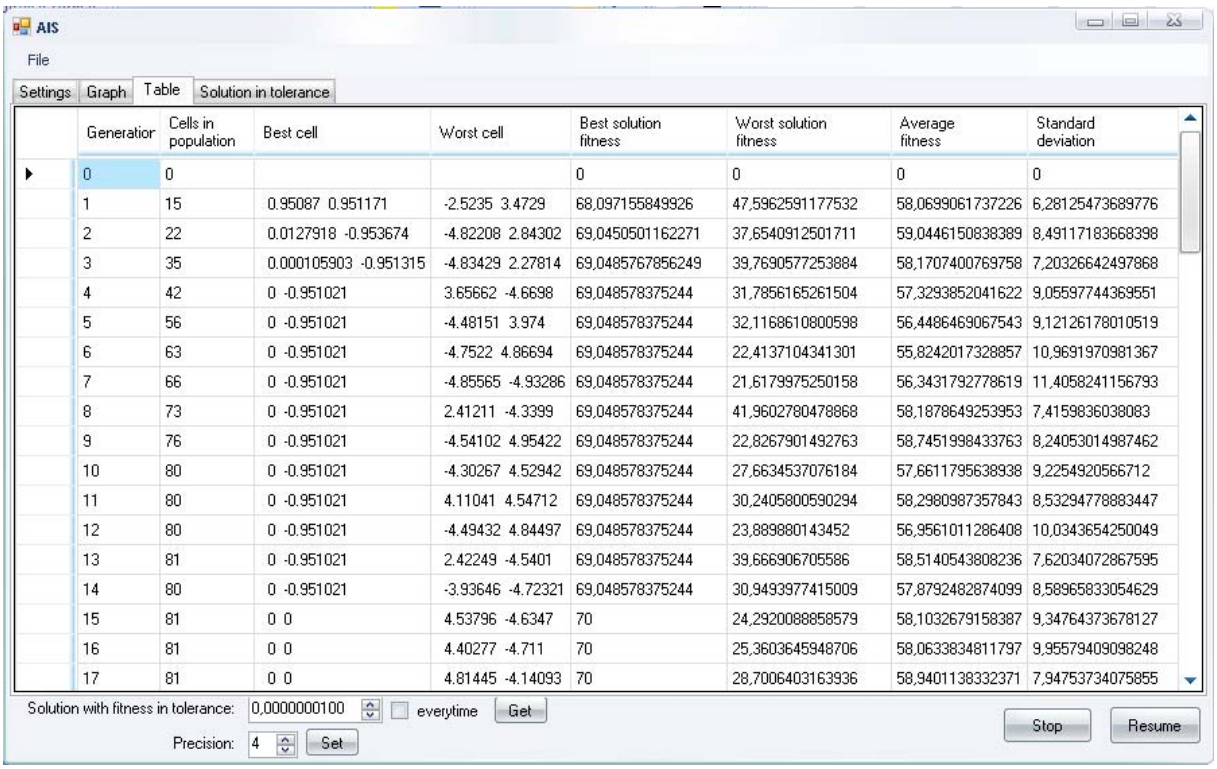
Pro vykreslení grafu využívá komponentu ZedGraph, která je šířena pod licencí LGPL, což umožňuje její volné použití v této aplikaci. Jedná se o množinu tříd napsanou v C# pro vykreslování 2D grafů.

5.2.3 Tabulka výsledků

Aplikace obsahuje záložku Table, ve které je k dispozici přehledná tabulka do které se každou generaci zapíše:

Generation (Aktuální číslo generace)	Cell in population (Počet buněk (řešení) v populaci)	Best cell (Nejlepší řešení)	Worst cell (Nejhorší řešení)

Best solution fitness (Fitness nejlepšího řešení)	Worts solution fitness (Fitness nejhoršího řešení)	Average fitness (Průměrná fitness populace)	Standard deviation (Standardní odchylka)



Generation	Cells in population	Best cell	Worst cell	Best solution fitness	Worst solution fitness	Average fitness	Standard deviation
0	0			0	0	0	0
1	15	0.95087 0.951171	-2.5235 3.4729	68,097155849926	47,5962591177532	58,0699061737226	6,28125473689776
2	22	0.0127918 -0.953674	-4.82208 2.84302	69,0450501162271	37,6540912501711	59,0446150838389	8,49117183668398
3	35	0.000105903 -0.951315	-4.83429 2.27814	69,0485767856249	39,7690577253884	58,1707400769758	7,20326642497868
4	42	0 -0.951021	3.65662 -4.6698	69,048578375244	31,7856165261504	57,3293852041622	9,05597744369551
5	56	0 -0.951021	-4.48151 3.974	69,048578375244	32,1168610800598	56,4486469067543	9,12126178010519
6	63	0 -0.951021	-4.7522 4.86694	69,048578375244	22,4137104341301	55,8242017328857	10,9691970981367
7	66	0 -0.951021	-4.85565 -4.93286	69,048578375244	21,6179975250158	56,3431792778619	11,4058241156793
8	73	0 -0.951021	2.41211 -4.3399	69,048578375244	41,9602780478868	58,1878649253953	7,4159836038083
9	76	0 -0.951021	-4.54102 4.95422	69,048578375244	22,8267901492763	58,7451998433763	8,24053014987462
10	80	0 -0.951021	-4.30267 4.52942	69,048578375244	27,6634537076184	57,6611795638938	9,2254920566712
11	80	0 -0.951021	4.11041 4.54712	69,048578375244	30,2405800590294	58,2980987357843	8,53294778883447
12	80	0 -0.951021	-4.49432 4.84497	69,048578375244	23,889880143452	56,9561011286408	10,0343654250049
13	81	0 -0.951021	2.42249 -4.5401	69,048578375244	39,666906705586	58,5140543808236	7,62034072867595
14	80	0 -0.951021	-3.93646 -4.72321	69,048578375244	30,9493977415009	57,8792482874099	8,58965833054629
15	81	0 0	4.53796 -4.6347	70	24,2920088858579	58,1032679158387	9,34764373678127
16	81	0 0	4.40277 -4.711	70	25,3603645948706	58,0633834811797	9,95579409098248
17	81	0 0	4.81445 -4.14093	70	28,7006403163936	58,9401138332371	7,94753734075855

Obrázek 29 – Screenshot aplikace s tabulkou výsledků

Data v této tabulce mohou být libovolně řazena například dle nejlepší fitness apod., kliknutím na příslušný název sloupce.

5.2.4 Zobrazení prvků s fitness v dané toleranci

Pro zobrazení prvků s hodnotou fitness, která je v dané toleranci, slouží záložka solution in tolerance. Toleranci je možno nastavit v editboxu: Solution with fitness in tolerance. Při zaškrtnutí checkboxu evertyme jsou tyto prvky zobrazovány každou generaci, jinak jsou zobrazovány na stisknutí tlačítka Get.

5.2.5 Ovládání a použití

Parametry programu lze ukládat a zpětně načítat do konfiguračního programu buď přes položky v menu nebo pomocí klávesových zkratek CTRL+S pro uložení a CTRL+L pro načtení. Výpočet probíhá v jiném vlákně než v tom, které se stará o vykreslování a obsluhu událostí z GUI, tudíž program během výpočtu reaguje na vstupní podněty od uživatele. Výstupní data z tabulky lze lehce přenášet do aplikace Excel pouhým CTRL+C a CTRL+V.
Program implicitně považuje za nejlepšího jedince toho s nejvyšší fitness a hledá tedy maximum dané funkce, pokud je třeba hledat minimum pro danou funkci a ne maximum, je nutné dopsat nakonec do výrazu pro výpočet funkce:

$$y = \mathrm{Max}(f) - f \text{ nebo } y = \frac{1}{1+f}$$

Kde Max (f) je maximální hodnota funkce na daném intervalu proměnných. Tím si zajistíme hledání minima.
Vzhledem k tomu, že se jedná o stochastický proces, není nikdy zaručeno, že aplikace po určité generaci dospěje do globálního optima, proto byla vytvořena ukázková citlivostní analýza programu pro předem danou multimodální funkci.

5.3 Popis implementace

Program je implementován ve vývojovém prostředí Microsoft Visual Studio 2008 a využívá pro svůj běh platformu. NET Framework 2. Nese v sobě implementaci již zmiňovaného opt-aiNet algoritmu pro optimalizaci multimodálních funkcí a umožňuje nastavit řadu parametrů pro optimální běh programu. Proto obsahuje několik tříd usnadňující implementaci algoritmu:

Třída **CELLS:** Třída CELLS reprezentuje buňku (jedno konkrétní řešení). Tato buňka může obsahovat 1...*n* hodnot, kde *n* je rovno parametru **count of variables.** Obsahuje metody pro práci s touto buňkou, mezi nejdůležitější patří:

Mutate: Metoda zmutuje danou buňku, mutace je provedena dle algoritmu. Vstupním parametrem je parametr mutace a dimense hodnot, kterých může buňka nabývat.

SetFitnessNorm: Nastaví parametr fitnessNorm na normovanou hodnotu fitness. Parametry funkce je minimální a maximální hodnota fitness.

getAffinity: Vrátí affinitu vůči jiné buňce, která je vstupním parametrem této metody.

Calculate_Fitness: Vypočítá fitness hodnotu dané buňky.

Dále jsou zde další metody pro klonování a manipulaci s hodnotou buňky apod. Více viz dokumentace.

Třída **Population:** Třída Population představuje celkovou populaci buněk. Obsahuje veškeré buňky v populaci, jejich průměrnou, nejlepší a nejhorší fitness apod. Její nejdůležitější metody jsou:

AddCell: Metoda slouží k přidání buňky, která je zadána vstupním parametrem metody.

Generate_random: Slouží k přidání n náhodně vygenerovaných buněk do populace. Vstupními parametry jsou počet buněk k přidání, počet jednotlivých hodnot v buňce, dimenze přípustných hodnot v buňce.

Calculate_pop_fitness: Vypočítá pro všechny buňky v populaci jejich fitness a nastaví parametry nejlepší, nejhorší a průměrnou fitness a také pro každou buňku volá metodu pro výpočet její normované hodnoty fitness.

Dále obsahuje metody pro vrácení nejlepší a nejhorší buňky v populaci apod. Více viz dokumentace.

Třída **Optim_Function:** Tato třída slouží pro výpočet fitness dle zadané funkce v textovém poli. Pokud je textové pole prázdné použije se výpočet definovaný v metodě eval. Použití této varianty se doporučuje při časté evaluaci této funkce (například s nízkým parametrem fitness threshold je evaluace funkce volána často a textové vyhodnocování je nesrovnatelně pomalejší než zkompilovaný kód.)

eval: Metoda slouží k výpočtu fitness zadané buď v textovém poli nebo přímo v metodě eval. Pro vyhodnocení textového výrazu využívá evaluátor výrazů, který je volně šířený jako ukázkový příklad programu AnaGram Parser Generator, který z dané gramatiky a pravidel vygeneruje kód jazyka C.

Třída **Opt_aiNet:** Zastřešuje celý algoritmus a metody pro práci s ním. Konstruktor instance této třídy očekává zadání všech vstupních parametrů algoritmu.
Její metody jsou jednotlivé části algoritmu opt-aiNet. Mezi ně patří metody: **ClonalSelection**, **CellInteraction, Update_with_Clones, Standard_deviation, SetPrecision.** Více viz dokumentace.

Uživatelské rozhraní bylo implementováno s využitím vývojového prostředí Microsoft Visual Studia 2008. Samotný běh algoritmu probíhá v separátním vláknu, což umožňuje interakci s GUI programu i za běhu a sledovat tak vývoj generací v grafu a tabulkách apod. Běh výpočetního vlákna lze kdykoliv přerušit kliknutím na tlačítko STOP, lze také bez problému pouze pozastavit a následně spustit a to opětovným stisknutím tlačítka START (Tlačítko START během běhu algoritmu slouží jako pausa algoritmu a jeho popisek je změněn na PAUSE. Následně slouží jako opětovné spuštění algoritmu a nese popisek RESUME.)

Ovládání aplikace bylo koncipováno s ohledem na co nejlepší a intuitivní ovládání uživatelem. Hlavním úskalím programu je vhodně zvolit vstupní parametry algoritmu pro správnou konvergenci populace, proto vznikla níže popsaná citlivostní analýza. Výstupní hodnoty lze lehce přenášet do programu Excel a následně tyto data zpracovávat. Toho bylo využito i při citlivostní analýze.

Aplikace byla implementována za využití vývojového prostředí Microsoft Visual Studio 2008 a byla k němu vytvořena dokumentace dostupná přímo z programu klávesou F1 a je také vystavena online na adrese: http://hop.unas.cz/dipl/dokumentace/AIS/index.html

5.4 Citlivostní analýza

Jelikož většinou bývá největším problémem u těchto typů[6] aplikací nalézt vhodnou kombinaci vstupních hodnot parametrů jako je velikost populace, parametr mutace, fitness threshold apod., byla provedena citlivostní analýza na vstupní parametry na Rastriginovu multimodální funkci dvou proměnných.

5.4.1 Vybraná multimodální funkce pro analýzu

Pro analýzu byla vybrána Rastriginova funkce. Tato funkce je často používaná k testování algoritmů, jelikož má mnoho lokálních minim a to ji dělá složitou pro standardní metody prohledávání při hledání globálního minima. Je definována dle následujícího vztahu:

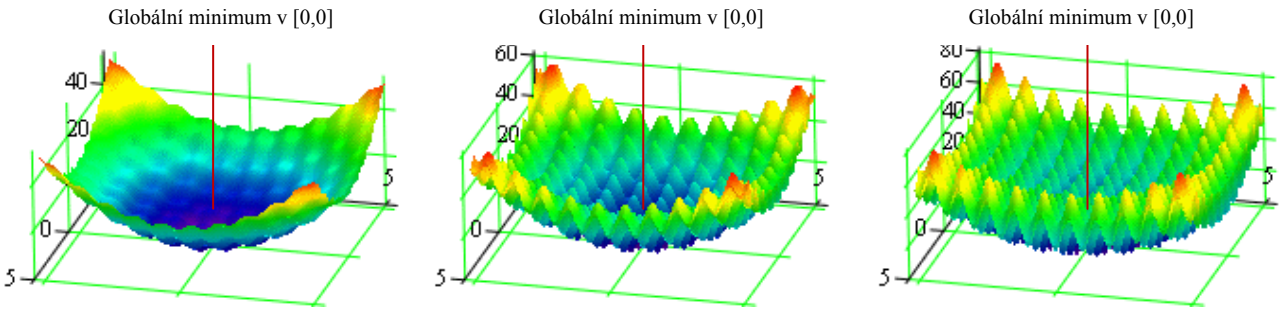
$$f(x) = nA + \sum_{i=1}^{n} (x_i^2 - Acos(2\pi x_i)); \quad \forall i \in [1..n], x_i \in [-5.12, 5.12]$$

Kde parametr:

n……. určuje dimensi funkce.

A……..určuje strmost lokálního optima.

5.4.2 Graf průběhu funkce

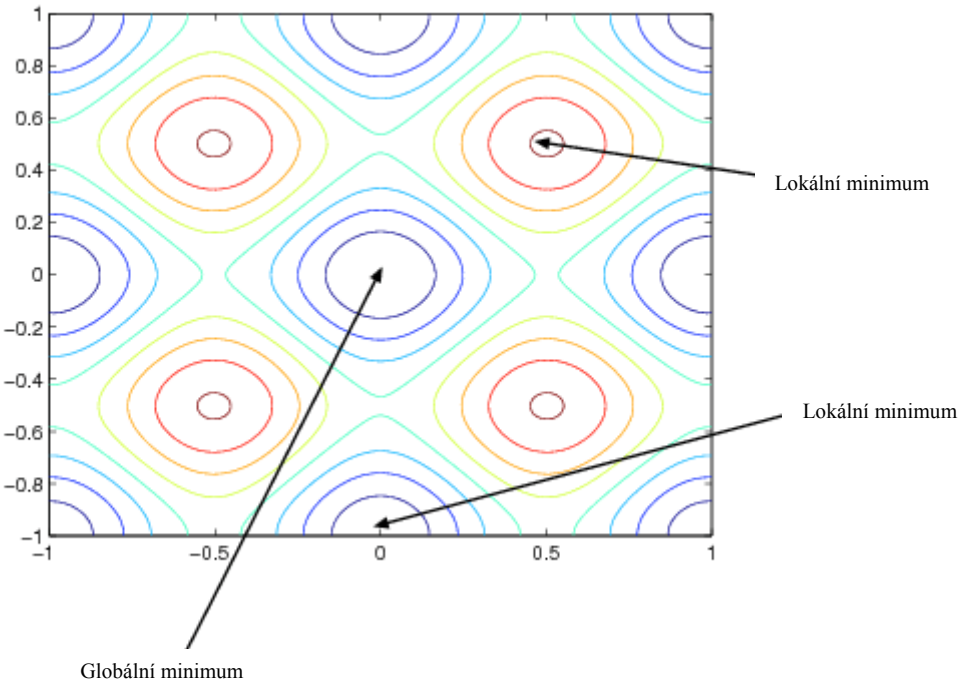


n=2, A=1 n=2, A=5 n=2, A=10

Obrázek 30 - Průběh Rastriginovy funkce pro různé parametry A.

[6] Evoluční a genetické algoritmy a podobné aplikace.

Rastriginova funkce má mnoho lokálních minim avšak pouze jedno globální a to pro n=2 v bodě [0,0].



Obrázek 31 - Rozložení hodnot Rastriginovy funkce.

5.4.3 Postup zpracování

Byla vytipována sada nejdůležitějších vstupních parametrů a pro tuto sadu byla provedena analýza. Pro každé nastavení parametrů bylo provedeno opakovaně 10 nezávislých běhů a pro tyto běhy byly vyneseny následující výsledné hodnoty:

- Průměrná hodnota řešení nejlepších jedinců z 10 běhů.
- Průměrná hodnota fitness nejlepších jedinců z 10 běhů.
- Průměrný rozptyl fitness nejlepších jedinců z 10 běhů.
- Průměrná velikost populace v 10 běžích.
- Průměrná hodnota úspěšné generace v 10 běžích.
- Procentuelní úspěšnost 10 běhů.

Za parametry, jejichž hodnoty jsou postupně měněny během analýzy, byly zvoleny:

Fitness threshold σ_f	Suppresion threshold σ_s	Factor newcommers d
0,01	0,3	0,1
0,108	1,04	0,38
0,206	1,78	0,66
0,304	2,52	0,94
0,402	3,26	1,22
0,5	4	1,5

Fitness threshold: Vyjadřuje, zda zlepšení střední hodnoty fitness funkce v i-té populaci je minimálně o $\sigma_f * 100$ procent lepší než v i-1 generaci. Tedy čím menší hodnota σ_f, tím delší dobu/větší počet generací bude populace zlepšována během klonální selekce. S nižší hodnotou tedy roste časová složitost a populace ztrácí diverzitu.

Suppresion threshold: Během globálního prohledávání prostoru je mezi jednotlivými řešeními určena afinita (Euklidova vzdálenost mezi všemi buňkami navzájem) a pokud je tato afinita nižší jak zadaný práh Suppresion threshold (jsou si více podobné, než dovoluje práh Suppresion threshold), je z populace odstraněna to řešení, jež má menší hodnotu fitness. Při nízké hodnotě prahu je odstraňováno málo řešení a populace pak neustále roste v důsledku parametru Factor newcommers.

Factor newcommers: Ovlivňuje, kolik náhodně vygenerovaných buněk bude před koncem aktuální generace doplněno do populace dle vztahu |P|**d*, kde *d* je Factor newcommers a |P| je aktuální velikost populace.

Zbylé parametry byly ponechány konstantní a to:

Population size = 10; Velikost počáteční populace.

Clones = 10; Počet vytvářených klonů pro každou buňku.

Max generations = 500; Maximální počet generací byl stanoven na 500.

Mutation parametr = 10; Parametr mutace.

Count of variables = 2; Budeme prohledávat prostor řešení pro 2 proměnné.

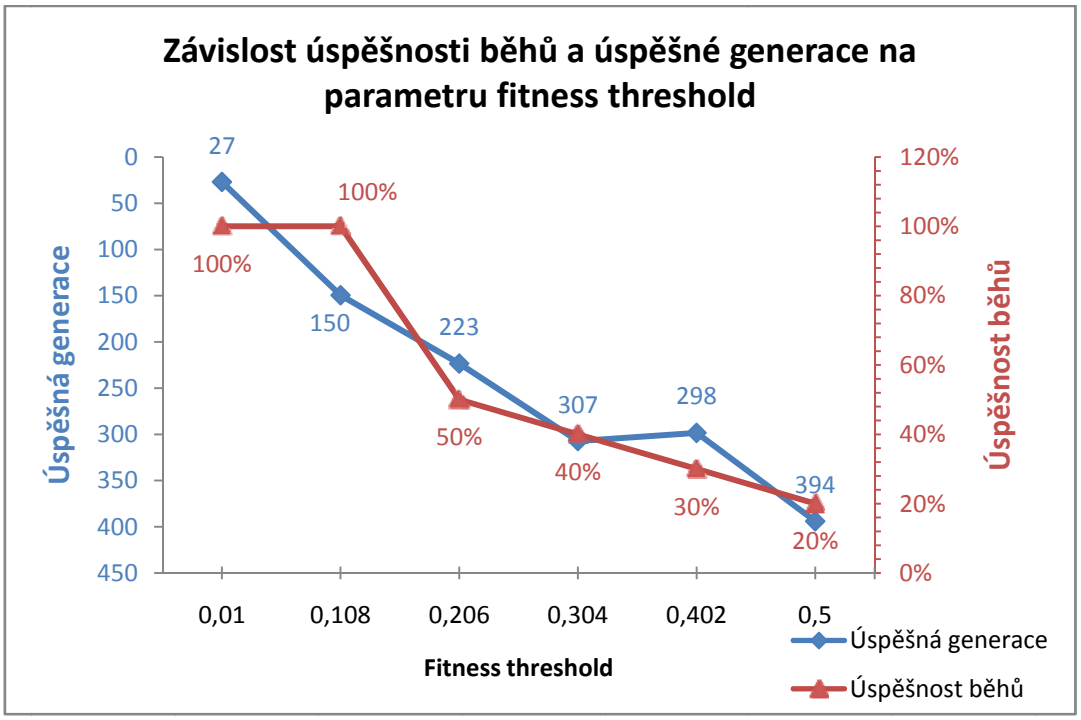
Input Range -5..5; Hodnoty řešení budou v tomto rozsahu.

Analýza byla provedena tak, že se hodnota jednoho ze zkoušených parametrů postupně měnila s tím, že ostatní parametry byly nastaveny na prostřední hodnotu z rozsahu tedy na hodnoty:
Fitness threshold=0,206; Suppresion threshold=1,78; Factor newcommers=0,66.
Z takto vzniklých dat byly vybrány nejlepší hodnoty jednotlivých parametrů. Parametr Precision byl přitom nastaven na hodnotu 4. Jako úspěšná generace je brána ta, která nalezla globální minimum, které je v bodě [0,0].

5.4.4 Naměřené údaje

Výsledky optimalizace získané pro proměnné σ_f a pro konstantní hodnoty parametrů.
$\sigma_s = 1,78$; $d = 0,66$:

Fitness threshold (σ_f)	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace	Úspěšnost běhů
0,01	0 0	70,000000000000	0	12	27	100%
0,108	0 0	70,000000000000	0	16	150	100%
0,206	0,0000294269 0,0000791367	69,999999676081	1,27765E-13	18	223	50%
0,304	0,000067566 0,0000741903	69,999999454197	4,08764E-13	18	307	40%
0,402	0,0000919303 0,0000635576	69,999999389675	2,36976E-13	18	298	30%
0,5	0,0000973787 0,0001102544	69,999999307585	2,46845E-13	18	394	20%

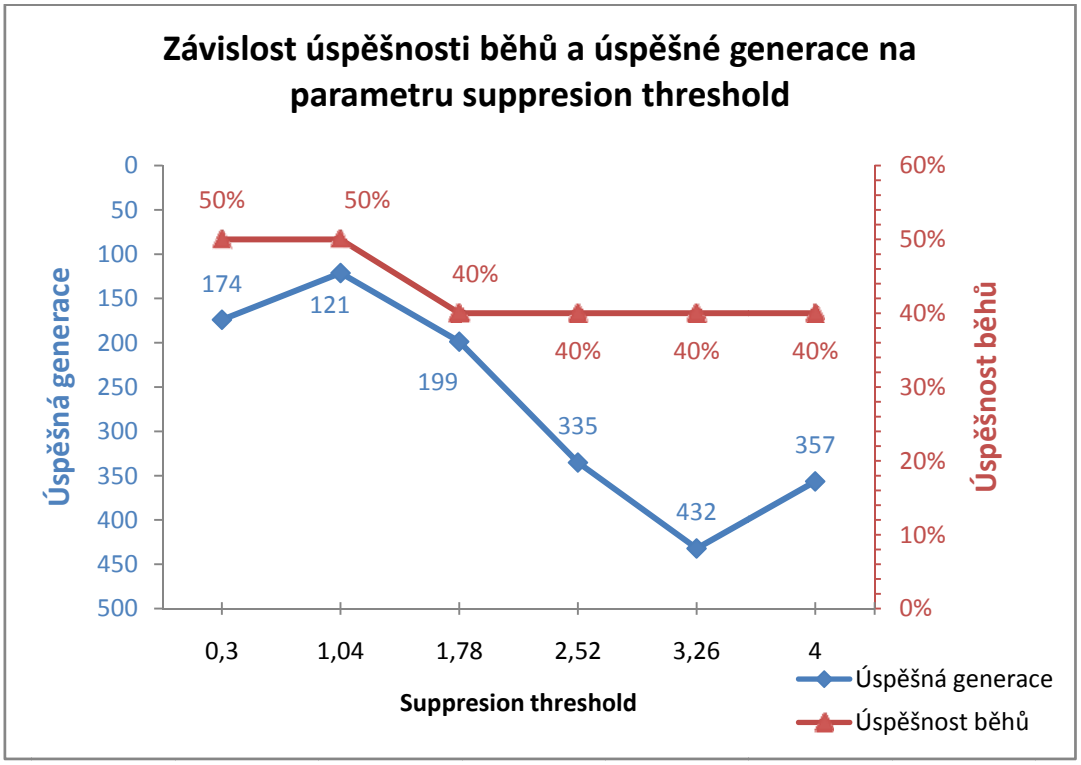


Obrázek 32 - Graf závislosti úspěšnosti běhů a úspěšné generace na parametru fitness threshold

Výsledky optimalizace získané pro proměnné σ_s a pro konstantní hodnoty parametrů.

$\sigma_f = 0{,}206$; $d = 0{,}66$:

Suppresion threshold (σ_s)	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace	Úspěšnost běhů
0,3	0,0000476009 0,0000294039	69,999999760403	8,3473E-14	528	174	50%
1,04	0,0000540733 0,0000157733	69,999999805794	5,20385E-14	51	121	50%
1,78	0,0000797166 0,0000542792	69,999999586759	2,44272E-13	18	199	40%
2,52	0,0027268272 0,0044167863	69,995002209400	0,000224785	9	335	40%
3,26	0,0000785236 0,0000115798	69,999999729915	7,33953E-14	6	432	40%
4	0,0992509314 0,0000865581	69,896157526166	0,09704239	4	357	40%

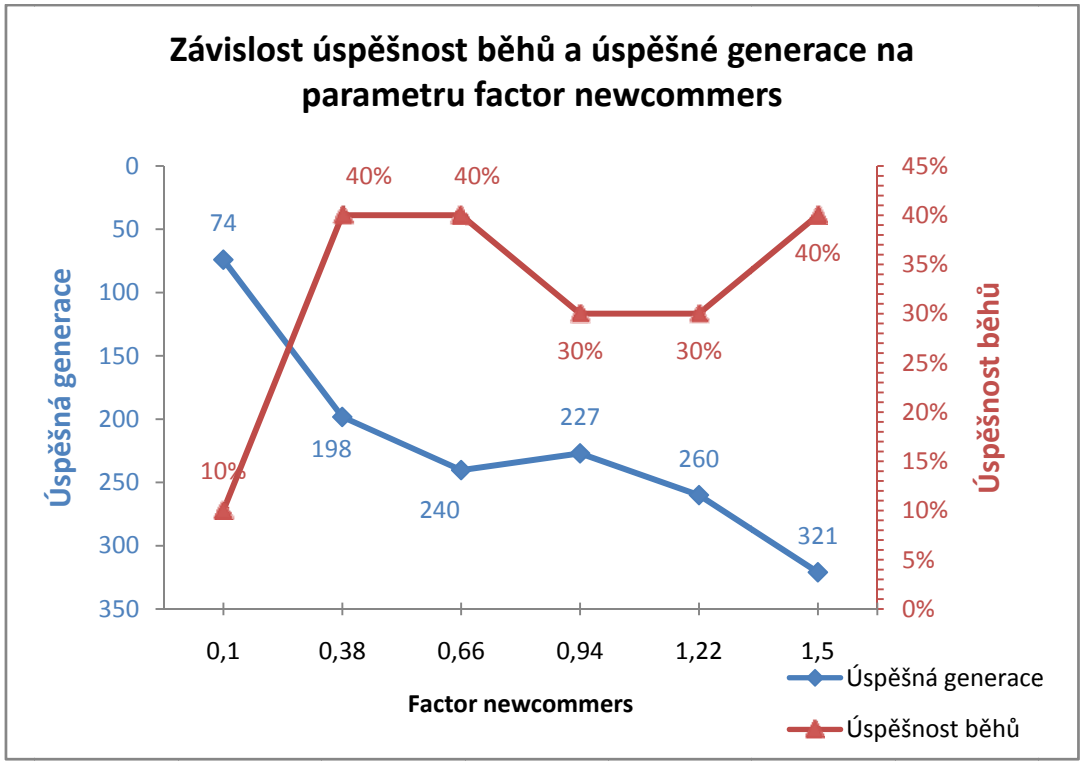


Obrázek 33 - Graf závislosti úspěšnosti běhů a úspěšné generace na parametru suppresion threshold

Výsledky optimalizace získané pro proměnné σ_s a pro konstantní hodnoty parametrů.

$\sigma_f = 0{,}206$; $\sigma_s = 1{,}78$:

Factor newcommers (d)	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace	Úspěšnost běhů
0,1	0,1982846908 0,0991623852	69,688482182004	0,226433679	11	74	10%
0,38	0,000037742 0,0000795013	69,999999599697	2,72184E-13	15	198	40%
0,66	0,0000542058 0,0000597691	69,999999674355	1,10006E-13	18	240	40%
0,94	0,0000884452 0,0000626066	69,999999536201	1,66442E-13	20	227	30%
1,22	0,0000482917 0,0000721054	69,999999640599	1,50224E-13	22	260	30%
1,5	0,0000664796 0,0000332101	69,999999713559	1,23667E-13	25	321	40%



Obrázek 34 - Graf závislosti úspěšnost běhů a úspěšné generace na parametru factor newcommers.

Z výsledných dat si vybereme nejlepší hodnoty parametrů a pro tyto hodnoty provedeme znovu analýzu. Pro dané výsledky běhů vyšli nejlepší kombinace vstupních parametrů v těchto rozsazích:

Fitness threshold	Suppresion threshold	Factor newcommers
0,01	0,50	0,38
0,04	0,83	0,47
0,08	1,17	0,57
0,11	1,50	0,66

Opět si z nich vybereme za referenční prostřední hodnotu.

Fitness threshold	Suppresion threshold	Factor newcommers
0,04	0,83	0,47

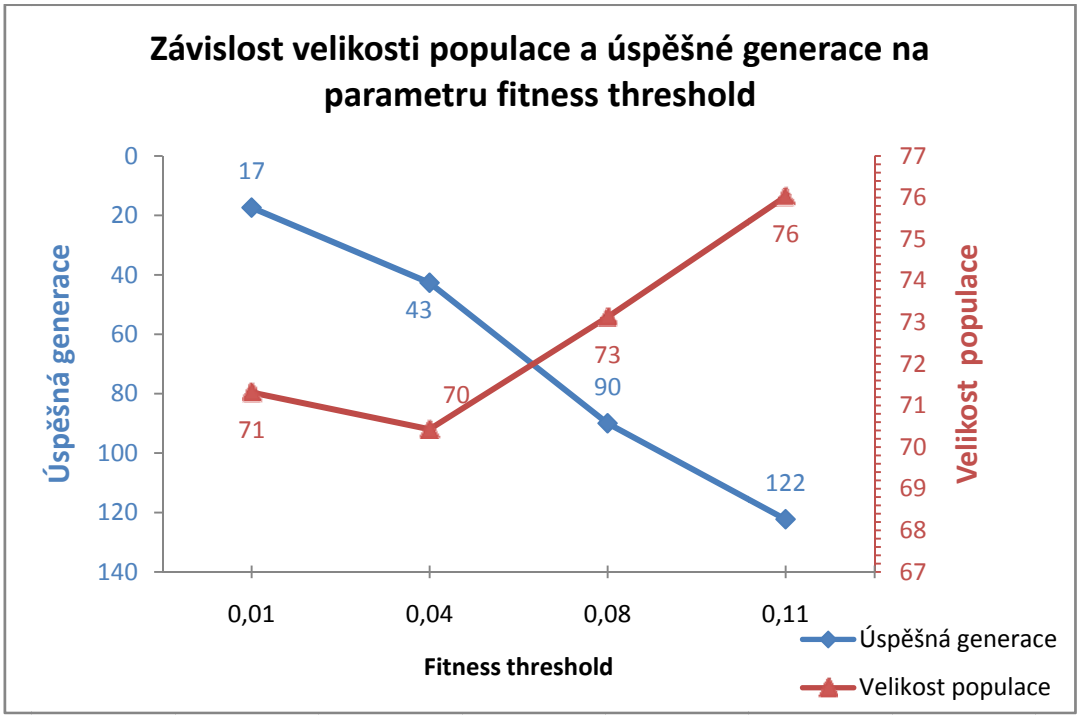
A provedeme další analýzu s novými rozsahy parametrů.

Výsledky optimalizace získané pro proměnné σ_f a pro konstantní hodnoty parametrů.

$\sigma_s = 0{,}83$; $d = 0{,}47$:

Fitness threshold (σ_f)	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace	Úspěšnost běhů
0,01	0 0	70	0	71	17	100%
0,04	0 0	70	0	70	43	100%
0,08	0 0	70	0	73	90	100%
0,11	0 0	70	0	76	122	100%

Jelikož je nyní již úspěšnost všech běhů stoprocentní, do grafu byla na vedlejší osu vynesena průměrná velikost populace.

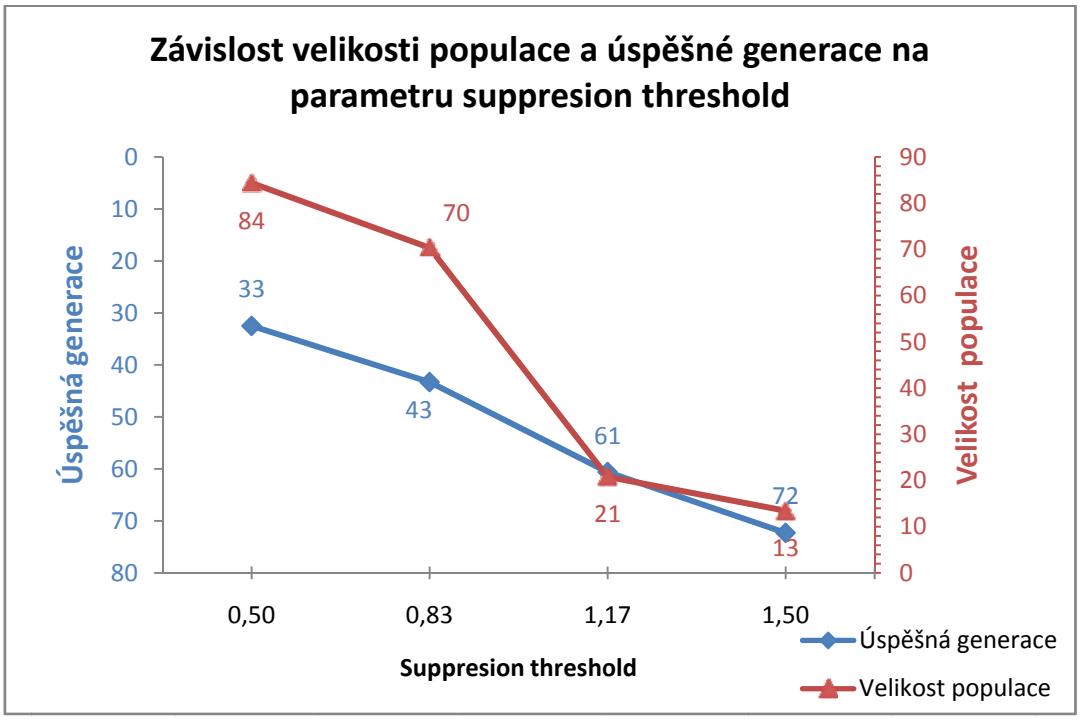


Obrázek 35 - Graf závislosti velikosti populace a úspěšné generace na parametru fitness threshold

Výsledky optimalizace získané pro proměnné σ_s a pro konstantní hodnoty parametrů.

$\sigma_f = 0{,}04$; $d = 0{,}47$:

Suppresion threshold (σ_s)	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace	Úspěšnost běhů
0,50	0 0	70	0	84	33	100%
0,83	0 0	70	0	70	43	100%
1,17	0 0	70	0	21	61	100%
1,50	0 0	70	0	13	72	100%

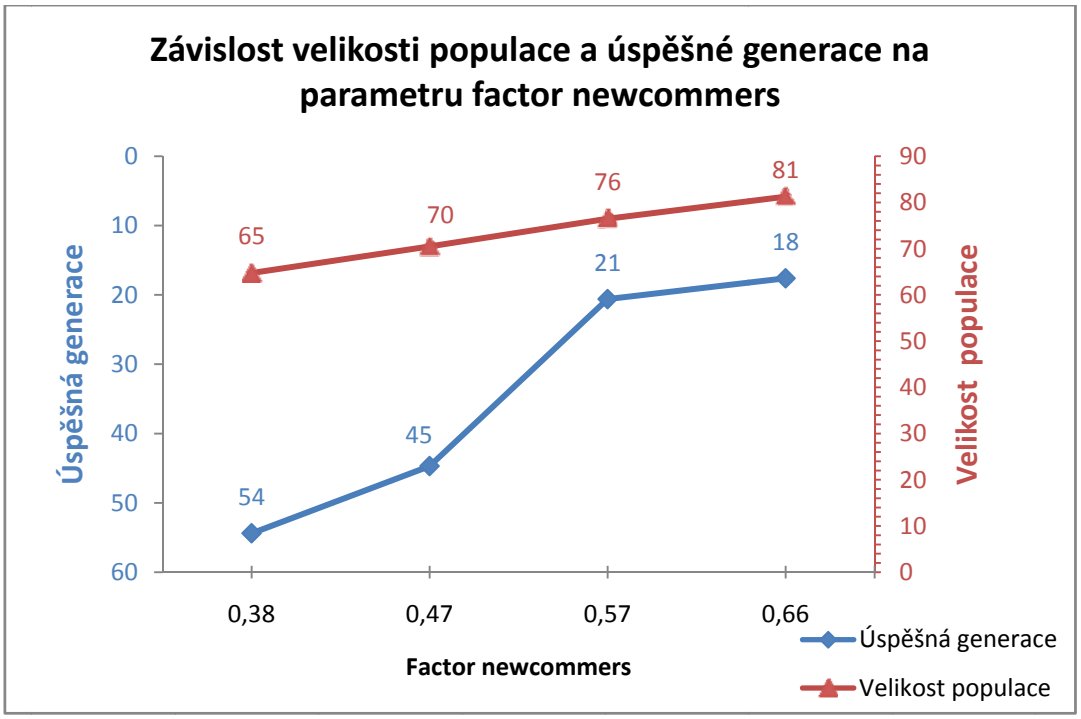


Obrázek 36 - Graf závislosti velikosti populace a úspěšné generace na parametru suppresion threshold

Výsledky optimalizace získané pro proměnné d a pro konstantní hodnoty parametrů.

$\sigma_f = 0{,}04$; $\sigma_s = 0{,}83$:

Factor newcommers (d)	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace	Úspěšnost běhů
0,38	0 0	70	0	65	54	100%
0,47	0 0	70	0	70	45	100%
0,57	0 0	70	0	76	21	100%
0,66	0 0	70	0	81	18	100%



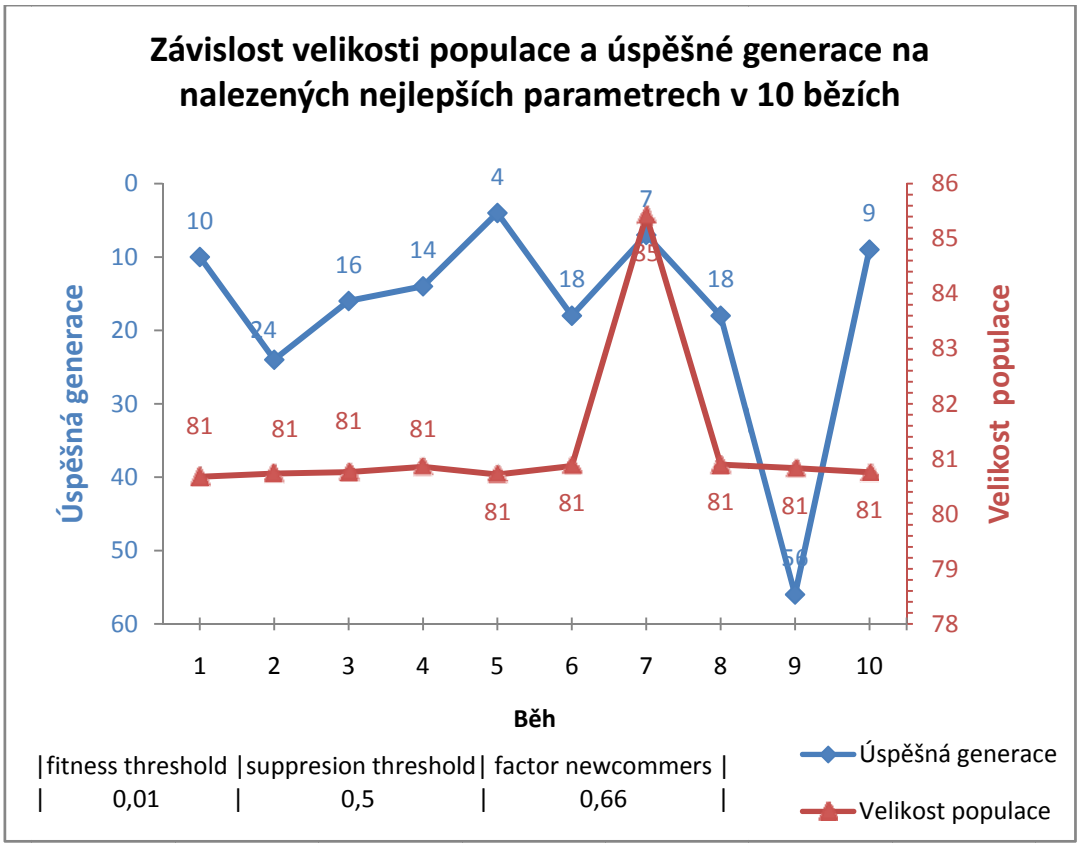
Obrázek 37 - Graf závislosti velikosti populace a úspěšné generace na parametru factor newcommers

Z těchto získaných výstupů je patrné, že pravděpodobně nejlepší kombinace hodnot parametrů bude:

Fitness threshold	Suppresion threshold	Factor newcommers
0,01	0,5	0,66

Pro tuto pravděpodobně nejlepší nalezenou kombinaci parametrů vychází test 10 běhy následovně:

Průměrná hodnota nejlepších jedinců	Průměrná hodnota nejlepších jedinců	Průměrná hodnota fitness nejlepších jedinců	Průměrný rozptyl fitness nejlepších jedinců	Průměrná velikost populace	Průměrná hodnota úspěšné generace
0 0	70	0	81	18	100%



Obrázek 38 - Výsledný graf jednotlivých běhů se získanými parametry

5.4.5 Závěr analýzy

Je velmi složité předem říci, jak nastavit vstupní parametry pro danou konkrétní funkci. Analýza vznikla pro lepší orientaci a bližší pochopení významu jednotlivých parametrů pro implementovaný algoritmus a dá se z ní vyvodit několik faktů. Pro hodnotu fitness threshold není vhodné volit příliš nízké hodnoty, jelikož při hodnotách nižších jak 0,01 je algoritmus nucen velmi mnohokrát zlepšovat danou populaci, než dosáhne tak malého zlepšení oproti předchozí populaci. S vysokými hodnotami zas tento parametr ztrácí význam, jelikož je zlepšení vždy menší než zadaný práh a tudíž se vůbec neprojeví. S hodnotou suppresion threshold je situace podobná. S nízkou hodnotou je v populaci ponecháno příliš jedinců (práh podobnosti je malý), což vede k obrovským populacím a s vysokou hodnotou je velikost populace naopak příliš „škrcena“, což vede ke slabé konvergenci. Parametr factor newcommers zas při malých hodnotách neudržuje populaci dostatečně velkou a při velkých naopak populaci neúměrně zvětšuje.

Nejlepším se zdá být řešení, kdy hodnoty vstupních parametrů vedou ke stabilní velikosti populace během celého běhu, dílčí nejlepší řešení se postupně neustále zlepšuje a výpočet trvá přiměřenou dobu.

V praxi to znamená velikost populace do 100 jedinců (z naměřených výsledků plyne, že není ani tak podstatná velikost populace jako nastavení ostatních parametrů), fitness threshold okolo 0,02 (s nižšími hodnotami trvá výpočet neúměrně dlouho) a factor newcommers 0,7.

Jelikož se jedná o stochastický proces a o konkrétní problém, nelze tato čísla brát obecně, ale jako náznak, kterým směrem se ubírat při výběru vstupních parametrů a vysvětlení jejich významu je jistě postačující.

6 Implementace imunitního systému v počítačové bezpečnosti

Jako součást této práce byla vytvořena aplikace pro ukázku možnosti využití inspirace v imunitním systému v počítačové bezpečnosti. Program je postaven na principech imunitní sítě používané v počítačové bezpečnosti, které byly popsány v [23] a již zmiňovány v kapitole 4.1.

Problém zajištění bezpečnosti počítačových systémů zahrnuje takové aktivity jako odhalování neautorizovaného použití počítačových zařízení, garantování integrity datových souborů a předcházení šíření počítačových virů. Na tyto problémy ochrany počítačového systému je zde nahlíženo jako na problém naučení se rozlišit mezi *vlastními* a *nevlastními* prvky systému.

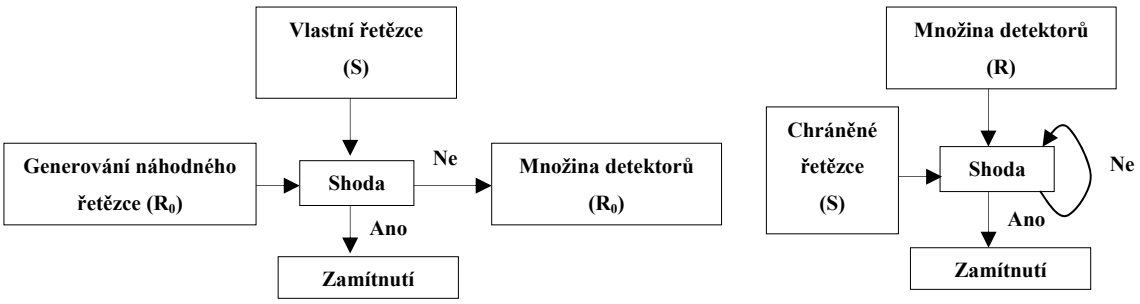
Algoritmus je rozdělen do dvou částí:

1. Vygenerování detektorů, kde každý detektor je řetězec[7], který neobsahuje *shodu* s chráněnými daty (vlastními prvky systému). Význam slova *shoda* bude přesněji popsán dále. Tuto fázi lze také nazvat „cenzurováním“.
2. Monitorování chráněných dat určováním *shody* s detektory. Pokud je detekována *shoda*, znamená to, že chráněná data byla pozměněna.

Takovýto přístup je z dosavadního nahlížení na ochranu počítačového systému velmi neortodoxní. Pokud se podíváme na množinu chráněných dat, jako na množinu řetězců nad konečnou abecedou a na změněná data jako na nějaký řetězec nepatřící do této množiny, potom vygenerované detektory identifikují všechny řetězce nepatřící do této chráněné množiny. Velice zajímavým jevem je, že i relativně malá množina detektorů, je schopna s velkou pravděpodobností detekovat náhodnou změnu

[7] Řetězcem je zde míněna posloupnost bitů, ale může jím být cokoliv v závislosti na aplikaci (instrukce assembleru, reprezentace komunikace na síti apod.)

ve chráněných datech. Dokonce počet detektorů může zůstávat konstantní vzhledem ke zvětšování množiny chráněných dat. Každá instance detekčního systému generuje jedinečnou množinu vlastních detektorů, kterými následně monitoruje systém. Funkci algoritmu popisuje obrázek 39.



Obrázek 39 – Generování detektorů a monitorování systému.

Chráněná data jsou tedy řetězce *vlastní* systému a nevlastní jsou všechny ostatní. Pro vygenerování platných detektorů, jsou nejdříve vlastní řetězce logicky rozděleny na segmenty stejné délky. Pro příklad si představme chráněný 32bitový řetězec logicky rozdělený na 8 4bitových vlastních řetězců.

0010 1000 1001 0000 0100 0010 1001 0011

Máme tedy kolekci chráněných dat S. Dalším krokem je vygenerování náhodných řetězců R_0 a určením jejich *shody* s řetězci v S. Ty, u kterých dojde ke *shodě,* jsou vyřazeny, ostatní jsou přidány do množiny detektorů R nazývané repertoár. Tato fáze je fáze první a nazývá se cenzurování. Za předpokladu, že jsou vygenerovány v množině R_0 tyto řetězce: 0111, 1000, 0101, 1001, budou množiny detektorů R zahrnuty řetězce: 0111, 0101. Jakmile je vygenerován požadovaný počet detektorů (pro detekci změny v chráněných datech s danou pravděpodobností), přejde se do druhé fáze, kdy jsou těmito detektory sledovány chráněná data. Pokud takto monitorovaná data budou pozměněna, detektory tuto změnu s danou pravděpodobností odhalí.

Tento příklad předpokládal *shodu* řetězců na všech odpovídajících si pozicích. Avšak takováto *shoda* na použitelné délce řetězce v praxi je velice vzácná a proto je *shoda* definována parametrem *r*, který určuje kolik, po sobě bezprostředně následujících znaků ve dvou řetězcích musí být stejných, aby byla mezi těmito řetězci prohlášena shoda.

Pro generování řetězců je užitečné vědět, s jakou pravděpodobností P_m dojde ke *shodě* dvou náhodných řetězců (převzato z [23]):

$$P_m \approx m^{-r}[\frac{(l-r)(m-1)}{m} + 1]$$

Kde je: m……..velikost abecedy

l……počet znaků řetězce

r……parametr shody

Další užitečné vzorce pro běh algoritmu:

$$P_f = (1 - P_m)^{N_R}$$

$$N_R = \frac{-lnP_f}{P_m}$$

$$f = (1 - P_m)^{N_S}$$

P_f……Pravděpodobnost, že N_R detektorů neodhalí infekci.

N_R……..Počet vygenerovaných detektorů v první fázi (cenzurování).

N_S……...Počet vlastních řetězců (chráněných dat)

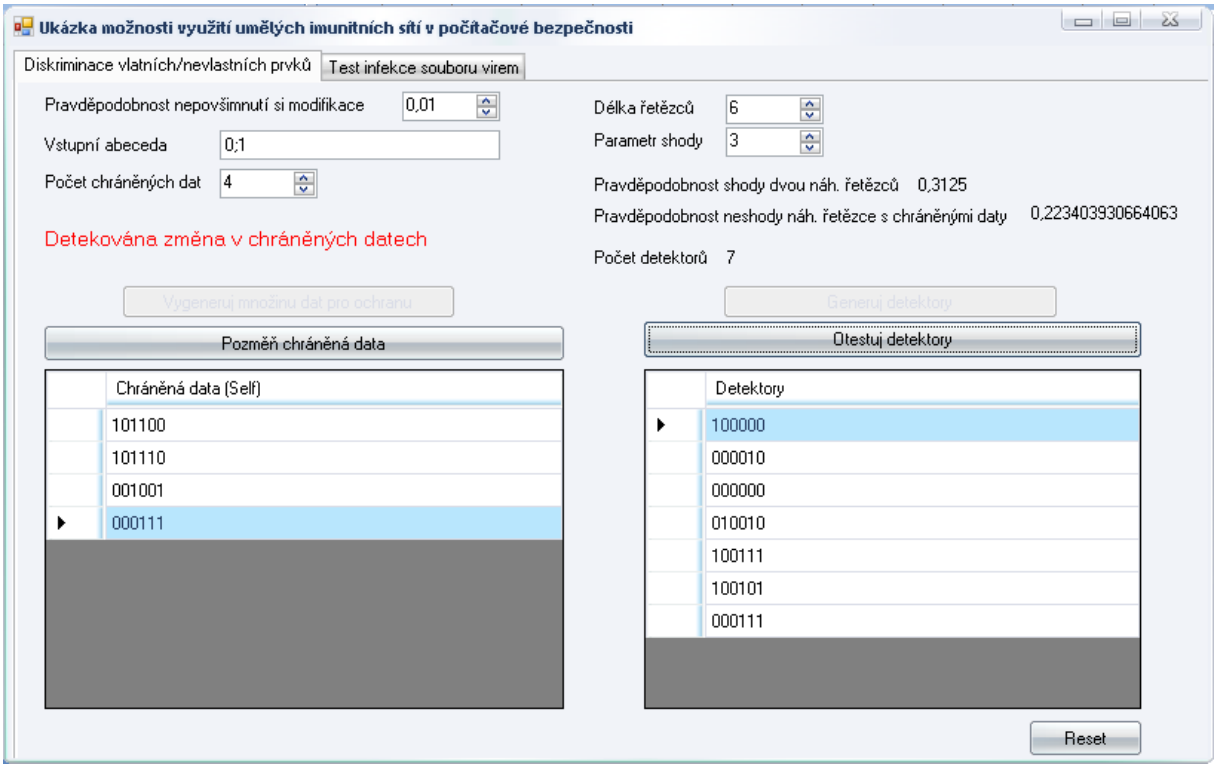
f.....Pravděpodobnost, že se bude náhodný řetězec shodovat s některým z N_S vlastních.

Tento algoritmus se inspiruje v dozrávání T buněk v imunitním systému. Jelikož je imunitní systém zodpovědný za rozlišení cizích buněk a molekul, musí rozlišovat mezi tělu vlastními a nevlastními prvky. Právě T buňky mají na svém povrchu receptory, které dokážou detekovat antigeny. T buňky jsou v těle vytvářeny pseudo náhodným způsobem a jsou cenzurovány v brzlíku (thymus), aby se nevázaly na tělu vlastní prvky. Právě tomuto procesu se říká negativní selekce. Na tomto principu je také aplikace postavena a slouží k prezentaci této možnosti využití v počítačové bezpečnosti obecně, jelikož řetězec nemusí reprezentovat jen nějaká data souborů, ale například také ve vhodně zvoleném formátu provoz na síti apod.

6.1 Popis aplikace

Aplikace je rozdělena do dvou částí. V té první je názorná prezentace funkčnosti algoritmu, s možností:

- ✓ zvolit libovolnou abecedu řetězce,
- ✓ zvolit délku řetězce,
- ✓ počet chráněných řetězců,
- ✓ pravděpodobnost P_f a
- ✓ parametr shody *r*

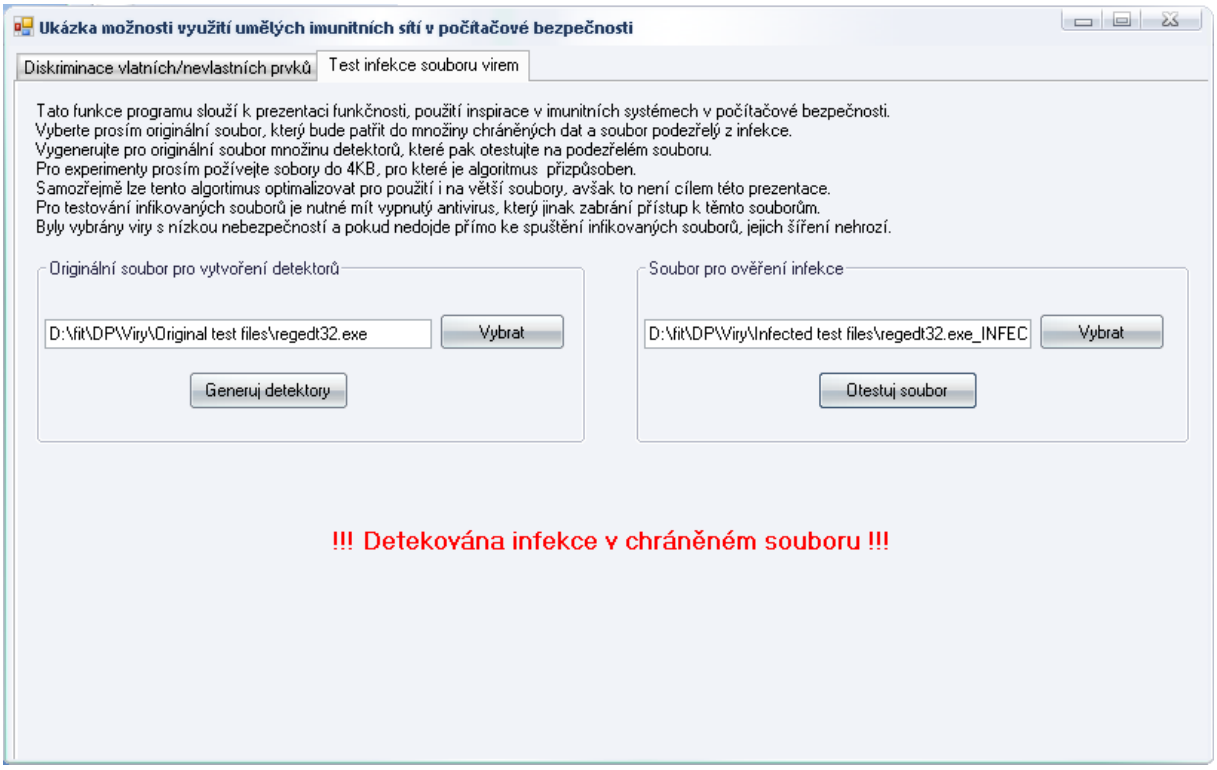


Obrázek 40 - Screenshot aplikace

Po nastavení zvolených parametrů algoritmu je možné náhodně vygenerovat množinu dat, která bude chráněna. Následně pro tato data vygenerovat sadu detektorů, která je bude monitorovat a provést jejich test. Pokud chráněná data zůstanou nepozměněna, detektory se nebudou shodovat s žádným z řetězců a nebude tedy detekována jejich změna.Pokud ovšem dojde ke změně v chráněných datech, (k tomuto účelu je zde možnost pozměnit náhodnou část chráněných dat) dojde s danou pravděpodobností k detekci této změny a program na to uživatele upozorní.

Pro znovuustavení parametrů algoritmu slouží tlačítko Reset.

Druhá část aplikace slouží k prezentaci funkčnosti tohoto algoritmu na konkrétních funkčních virech. V aplikaci je nejprve nutné vybrat originální soubor (tento soubor představuje chráněná data před nakažením infekcí) a vygenerovat na něj příslušné detektory. K tomuto souboru pak vybrat jeho podobu po infikaci virem (takovýto soubor reprezentuje pozměněný původní soubor po nákaze systému virem) a provést test. Spuštění testu simuluje monitoring chráněných dat. Pokud byl soubor, na nějž byly detektory vygenerovány, nějak změněn, je tato změna detektory odhalena a aplikace na to upozorní.



Obrázek 41 – Screenshot aplikace při testu souboru na infekci virem

Pro tyto účely byly vytipovány skutečné viry dostupné z internetu a byly s nimi nakaženy vybrané soubory skutečných aplikací. Nicméně byly vybrány viry s nízkým rizikem, které se nešíří bez přímého spuštění nakažené aplikace. Vybranými viry jsou:

Alxe.1287

K_Hate.1128

BadSize.369

A jsou přiloženy k aplikaci. Jejich detailní popis je uveden například na stránkách http://www.virus-database.com/.

Aplikace byla implementována za využití vývojového prostředí Microsoft Visual Studio 2008 a byla k němu vytvořena dokumentace dostupná přímo z programu klávesou F1 a je také vystavena online na adrese: http://hop.unas.cz/dipl/dokumentace/Viry/index.html

7 Závěr

První část této práce je zaměřena na seznámení se s problematikou biologického imunitního systému. Byly zde vysvětleny základní vlastnosti a principy fungování imunitního systému a vyzdviženy hlavní vlastnosti aplikovatelné v umělých imunitních systémech.

V druhé části již bylo popsáno obecné schéma umělého imunitního systému, základní algoritmy imunitních systémů, způsob určování míry afinity a reprezentace problému. V návaznosti na to byly představeny již existující aplikace a použité principy umělých imunitních systémů v technické praxi.

Konkrétně zde byly představeny možnosti využití aplikací inspirovaných imunitním systémem v počítačové bezpečnosti, principy funkce takovýchto aplikací a byly zde prezentovány různé práce zabývající se touto tématikou.

Součástí této práce jsou také 2 vytvořené aplikace inspirované imunitním systémem. První pro optimalizaci multimodálních úloh a druhá pro představení možnosti využití inspirace v imunitním systému v počítačové bezpečnosti například pro detekci virů.

Literatura

[1] FERENČÍK, Miroslav. Imunitní systém: informace pro každého. Praha : Grada Publishing, 2005. 236 s. ISBN 80-247-1196-6.

[2] Buc M. a kol., Imunológia, skripta LF Univerzita Komenského, Bratislava, 1998.

[3] VÁCHA, Martin, et al. SROVNÁVACÍ FYZIOLOGIE ŽIVOČICHŮ. 2. aktualiz. vyd. Masarykova univerzita v Brně : [s.n.], 2004. 157 s. ISBN 80-210-3379-7.

[4] JERNE, N.K. Towards a Network Theory of the Immune System : Annals of Immunology. (Inst.Pasteur) 125C, pp. 373-389, 1974

[5] DE CASTRO, L – TIMMIS, J: Artificial Immune systems: A new computational intelligence approach. London: Springer, 2002. ISBN: 1-85233-594-7.

[6] Hricková M., Zvirinský P., "Umelé imunitné systémy", *In: Proc. of 3rd Slovak-Czech Seminar on Cognition, Artificial Life and Computational Intelligence (CALCI'03)*, eds. P. Sincak, V. Kvasnicka, J. Pospichal, J. Kelemen, J. Navrat, Stara Lesna, SK, 2003.

[7] ZVIRINSKÝ, Peter. Umelé imunitné systémy. [s.l.], 2003. 36 s. Technická univerzita v Košicích Fakulta elektrotechniky a informatiky Katedra kybernetiky a umělé inteligence. Dizertační práce.

[8] De Castro, L.N., Von Zuben, F.J., 2000 Artificial Immune Systems: Part II. - A Survey of Applications, Technical Report - RT DCA 02/00

[9] ČERMÁK, Michal. Simulácia imunitnej odozvy pri interakcii s vírusmi. [s.l.], 2004. 46 s. Technická univerzita v Košicích Fakulta elektrotechniky a informatiky Katedra počítačů a informatiky. Diplomová práce.

[10] Sorkin, B. S., Kephart, J. O., Swimmer, M.: *Blueprint for a Computer Immune System,* In Proc. of the Seventh International Virus Bulletin Conference, Virus Bulletin Ltd., 1997.

[11] JACKSON, K. *Intrusion detection system product survey*. Los Alamos Nation Laboratory : Research report LA-UR-99-2882. 1999.

[12] KIM, Jungwon, BENTLEY, Peter. The Human Immune System and Network Intrusion Detection., 1999. Department of Computer Science, University Collge London. Dostupný z WWW: <http://www1.cs.columbia.edu/~locasto/projects/candidacy/papers/kim99human.pdf >.

[13] Dasgupta, D.: "Immunity-Based Intrusion Detection System: A General Framework", *Proc. of the 22nd NISSC*, 1, pp. 147-160, 1999a

[14] Varela, F.J., Coutinho, A., 1991 Second Generation Immune Networks, Immunlogy Today,12(5), pp. 159-166

[15] HASIČEK, Martin. Simulácia imunitnej odozvy pri interakcii s nádorovými bunkami. [s.l.], 2004. 54 s. Technická univerzita v Košicích: Fakulta elektrotechniky a informatiky Katedra kybernetiky a umělé inteligence. Diplomová práce.

[16] Endo S., Toma N., Yamada K., 1999 Immune Algorithm with Immune Network and MHC for Adaptive Problem Solving, Proc. of the IEEE System, Man, and Cybernetics, pp. 271-276

[17] Ishiguro, A., Ichikawa, S., Shibata, T., Uchikawa, Y,. 1998 Moderationism in the Immune System: Gait Acquistion of a Legged Robot Using the Metadynamics Function, Proc.of the IEEE System, Man and Cybernetics Conference, pp. 2827-3832

[18] Jon Timmis, T.K. (2002). Artificial Immune Systems, chapter 11, page 209. Idea Group Publishing.

[19] Stewart J. (2004), Un systeme cognitive sans nerones: les capacite d´adaptation.d´apprentisage et de memoire du systeme immunitaire. Intellectica.

[20] FRANZOLINI, Julien, OLIVIER, Damien. Self-organization in an artificial immune network system. [s.l.] : [s.n.], [2000?]. 5 s. University of Le Havre.

[21] KÖSTER, M., et al. A New Paradigm of Optimisation by Using Artificial Immune Reaction. Springer Berlin / Heidelberg. [s.l.] : [s.n.], 2003. 5 s. ISBN 978-3-540-40803-1.

[22] KEPHART, J.O., et al. Blueprint for a Computer Immune System : In Artificial Immune Systems and Their Applications. D. Dasgupta. Springer-Verlag, 1999. pp. 242-259.

[23] Forrest, S., A. Perelson, Allen, L. & Cherukuri, R. (1994), Self-Nonself Discrimination in a Computer, Proc. of the IEEE Symposium on Research in Security and Privacy, pp. 202-212. Dostupný z WWW: < http://www.cs.unm.edu/~immsec/publications/virus.pdf>.

[24] D'haeseleer, P., S. Forrest & P. Helman (1996), An Immunological Approach to Change Detection: Algorithms, Analisys and Implications, Proc. of the IEEE Symposium on Computer Security and Privacy. Dostupný z WWW: < http://www.cs.unm.edu/~forrest/publications/ieee-sp-96-neg-selec.pdf>.

[25] Forrest, S., Hofmeyr S. A. & Somayaji A. (1997), Computer Immunology, Communications of the ACM, 40(10), pp. 88-96.

[26] Okamoto, T. & Ishida, Y. (1999a), A Distributed Approach to Computer Virus Detection and Neutralization by Autonomous and Heterogeneous Agents, In Proc of the ISADS'99, pp. 328-331.

[27] Hofmeyr S. A. & Forrest, S. (2000), Architecture for an Artificial Immune System, submitted to Evolutionary Computation.

Seznam příloh

Popis obsahu CD

Na přiloženém CD je k dispozici elektronická podoba tohoto dokumentu v souborech Diplomová práce.doc (Dokument Microsoft Word) a Diplomová práce.pdf (Portable Document Format). Dále jsou přiloženy zdrojové texty vytvořených aplikací, jejich zkompilovaná podoba, knihovny potřebné pro jejich spuštění, pomocné soubory pro jejich testování (ukázkové konfigurační soubory, infikované soubory pro detekci viry) a technická dokumentace v podobě webových stránek a ve formátu chm (Microsoft Compiled HTML).

Technická dokumentace

V technické dokumentaci jsou uvedeny a popsány všechny metody použité v aplikacích. Na CD jsou ve formátu webových stránek (soubor index.html) a ve formátu Microsoft Compiled HTML (soubor dokumentace.chm). Jsou také vystaveny na internetu na adresách:
http://hop.unas.cz/dipl/dokumentace/Viry/index.html a
http://hop.unas.cz/dipl/dokumentace/AIS/index.html.

Strom obsahu CD