



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

EVOLUČNÍ OPTIMALIZACE ANALOGOVÝCH OBVODŮ

EVOLUTIONARY OPTIMISATION OF ANALOGUE CIRCUITS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ MIHULKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. BIDLO MICHAL, Ph.D.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Mihulka Tomáš, Bc.**

Obor: Bioinformatika a biocomputing

Téma: **Evoluční optimalizace analogových obvodů**
Evolutionary Optimisation of Analogue Circuits

Kategorie: Umělá inteligence

Pokyny:

1. Nastudujte principy vybraných typů analogových obvodů a možnosti jejich simulace pomocí systému SPICE.
2. Seznamte se s problematikou optimalizace pomocí evolučních algoritmů.
3. Navrhněte systém pro optimalizaci zvolených analogových obvodů pomocí evoluce. Zabývejte se různými způsoby vyhodnocování fitness.
4. Proveďte sady experimentů s využitím alespoň dvou typů evolučních algoritmů.
5. Zhodnoťte dosažené výsledky a diskutujte další možnosti projektu.

Literatura:

- Dle pokynů vedoucího projektu.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání, demonstrace prototypu optimalizačního systému z bodu 4.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

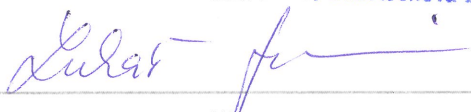
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bidlo Michal, Ing., Ph.D., UPSY FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Božetěchova 2



prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Cílem práce bylo vytvořit systém pro optimalizaci vybraných analogových obvodů pomocí evoluce s různými fitness funkcemi. Taktéž na tomto systému provádět sadu experimentů a na jejich základě vyhodnotit možnosti evoluční optimalizace analogových obvodů. Prekvizitou k tomuto cíli bylo nastudování a vybrání určitých typů analogových obvodů a evolučních algoritmů. Pro účely této práce byly z analogových obvodů vybrány zesilovače a oscilátory. Jako zástupci evolučních algoritmů byli zvoleni genetický algoritmy a evoluční strategie.

Abstract

The aim of this work was to create a system for optimisation of specific analog circuits by evolution using multiple fitness functions. A set of experiments was run, and the results analyzed to evaluate the feasibility of evolutionary optimisation of analog circuits. A requirement for this goal is the study and choice of certain types of analog circuits and evolutionary algorithms. For the scope of this work, amplifiers and oscillators were chosen as target circuits, and genetic algorithms and evolutionary strategies as evolutionary algorithms. The motivation for this work is the ongoing effort to automate the design and optimisation of analog circuits, where evolutionary optimisation is one of the options.

Klíčová slova

Evoluční optimalizace, analogový obvod, zesilovač, oscilátor, evoluční algoritmus, genetický algoritmus, evoluční strategie, fitness funkce, Pearsonův korelační koeficient, Darwin, Darwinovská evoluce, SPICE, Ngspice.

Keywords

Evolutionary optimisation, analogue circuits, amplifier, oscillator, evolutionary algorithm, Genetic algorithm, evolution strategy, fitness function, Pearson correlation coefficient, Darwin, Darwinian theory, SPICE, Ngspice.

Citace

MIHULKA, Tomáš. *Evoluční optimalizace analogových obvodů*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Bidlo Michal.

Evoluční optimalizace analogových obvodů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Bidla, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Mihulka
24. května 2017

Poděkování

Tímto bych rád poděkoval vedoucímu práce panu Ing. M. Bidlovi, Ph.D., za jeho trpělivost, čas, rady a vedení tohoto projektu.

Obsah

1 Úvod	3
2 Analogové obvody	4
2.1 Zesilovače	5
2.1.1 Parametry zesilovačů	5
2.1.2 Pracovní třídy zesilovačů	5
2.1.3 Jednostupňové zesilovače	6
2.2 Oscilátory	7
2.2.1 Zpětnovazební LC oscilátory	8
2.2.2 Zpětnovazební RC oscilátory	8
3 Evoluční algoritmy	10
3.1 Ch. Darwin, Darwinovská evoluce a neodarwinismus	11
3.2 Základní části EA	11
3.3 Průběh EA	12
3.4 Genetické algoritmy	12
3.4.1 Genotyp a kódování	12
3.4.2 Selektce	13
3.4.3 Křížení	14
3.4.4 Mutace	16
3.4.5 Obnova populace	16
3.5 Evoluční strategie	17
3.5.1 Genotyp a kódování	17
3.5.2 Křížení	17
3.5.3 Mutace	18
3.5.4 Obnova populace	19
3.5.5 Autoevoluce řídicích parametrů	19
4 Evoluční optimalizace analogových obvodů	20
4.1 Zadávání obvodů a jejich simulace	20
4.1.1 Zadávání obvodů	20
4.1.2 Simulace obvodů	21
4.1.3 Detekce nevhodných řešení	22
4.2 Kódování chromozomu	22
4.3 Fitness funkce	23
4.3.1 Fitness funkce s Pearsonovým korelačním koeficientem	23
4.3.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem	24
4.3.3 Poziční fitness funkce	25

5 Experiment - jednostupňový zesilovač	27
5.1 Testovací sestavy	28
5.2 Genetický algoritmus	28
5.2.1 Fitness funkce s Pearsonovým korelačním koeficientem	28
5.2.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem . .	30
5.2.3 Poziční fitness funkce	31
5.3 Evoluční strategie	33
5.3.1 Fitness funkce s Pearsonovým korelačním koeficientem	34
5.3.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem . .	35
5.3.3 Poziční fitness funkce	37
5.3.4 Shrnutí a diskuze	38
6 Experiment - oscilátor s Wienovým článkem	40
6.1 Genetický algoritmus	41
6.1.1 Fitness funkce s Pearsonovým korelačním koeficientem	41
6.1.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem . .	43
6.1.3 Poziční fitness funkce	44
6.2 Evoluční strategie	46
6.2.1 Fitness funkce s Pearsonovým korelačním koeficientem	46
6.2.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem . .	47
6.2.3 Poziční fitness funkce	48
6.2.4 Shrnutí a diskuze	49
7 Závěr	51
Literatura	52
Přílohy	54
Seznam příloh	55
A Obsah CD	56
B Návod k použití systému	57
B.1 Instalace prerekvizit	57
B.2 Instalace systému	57
B.3 Spuštění testu	57
B.4 Parametry systému	58

Kapitola 1

Úvod

Tato diplomová práce se zabývá evoluční optimalizací analogových obvodů. V poslední době se rozšířil zájem o biologii inspirované prohledávací algoritmy. K těmto algoritmům patří evoluční algoritmy, které jsou založené na principu evoluce. Díky jejich vysoké efektivnosti se používají v nejrůznějších odvětvích. Jedna z možností využití je evoluční návrh a optimalizace analogových obvodů.

Na analogové obvody jsou kladeny požadavky různého typu, jako např. velikost obvodu, počet použitých součástek, výsledná cena obvodu, provozní podmínky, požadavky na správný výsledný výstup obvodu a další. Při návrhu a nastavení parametrů daného obvodu se musí s těmito požadavky počítat. Návrh a analytické řešení nastavení parametrů obvodů není triviální [8]. K výslednému řešení je potřeba použít různých technik a přístupů, jako jsou např. princip superpozice, Kirchhoffovy zákony a mnoho dalšího. Tento proces je náročný a proto je snaha celý tento proces automatizovat. Tato automatizace je motivací této práce. Evoluční algoritmy by mohly být vhodné k optimalizaci již navržených analogových obvodů. Sloužily by jako část automatizace, která najde vhodné parametry daného obvodu vzhledem ke kladeným požadavkům na něj. Oproti konvenčním metodám výpočtu obvodu, evoluční algoritmus umožní prohledání většího prostoru možných řešení a možná i nalezení lepšího, než požadovaného výstupu obvodu. Mezi nevýhody evolučních algoritmů oproti konvenčním metodám výpočtu obvodu patří škálovatelnost řešení. Dalšími problémy jsou čas evoluce potřebný k nalezení požadovaného řešení nebo možnost nenalezení řešení, i pokud řešení existuje, případně nenajde nejlepší možné řešení.

Cílem této práce je ověřit možnosti evolučních algoritmů v optimalizaci navržených analogových obvodů a to zejména zesilovačů a oscilátorů. Na těchto obvodech vyzkoušet různé evoluční techniky z evolučních algoritmů, kdy se pomocí dané techniky nastavují hodnoty jednotlivých prvků obvodu. Snahou je získat nastavení obvodů, jehož výstup se co nejvíce blíží k požadovanému výstupu, nebo je i lepší. V rámci této práce bude využito a porovnáno několik různých přístupů fitness funkce. Všechny tyto varianty fitness funkcí a evolučních technik budou analyzované a vyhodnocené.

V kapitole 2 jsou popsány základní informace o analogových obvodech. Blíže specifikované jsou zde zesilovače a oscilátory, na kterých jsou prováděny testy této diplomové práce. V další kapitole 3 jsou prvně obecně popsány evoluční algoritmy, poté následuje bližší popis genetického algoritmu a evoluční strategie, které jsou použité v této práci. V kapitole 4 je blíže specifikováno řešení této práce, popis implementovaných technik, fitness funkcí a vyhodnocení kandidátních řešení. V kapitole 5 jsou popsány experimenty prováděné na zesilovači a v kapitole 6 experimenty na oscilátorech. Poslední kapitola 7 obsahuje shrnutí práce a dosažené výsledky.

Kapitola 2

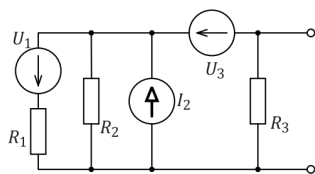
Analogové obvody

Elektrické obvody se podle základního rozdělení dělí na analogové obvody a číslicové obvody. Analogové obvody pracují s fyzikální veličinou (např. napětí, proud), jejíž průběh je spojitý. Mezi typické příklady obvodů lze zařadit zesilovače či oscilátory, které jsou předmětem zkoumání v této práci. Naproti tomu číslicové obvody [10] se charakterizují tím, že elektronické signály nabývají diskrétních hodnot. Obvykle nabývají binárních hodnot a to 0 nebo 1. Každý číslicový obvod je současně i analogovým obvodem, kde hodnoty číslicového obvodu jsou reprezentované určitou úrovní napětí. Z toho plyne, že analogové obvody jsou základními elektronickými obvody a proto se tato práce jimi zabývá.

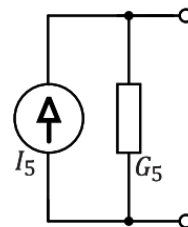
Analogový obvod je tvořen kombinací sériového a paralelního zapojení součástek, jako jsou např. rezistory, kondenzátory, cívky, diody, tranzistory a další. Tato zapojení jsou nejčastěji reprezentována schémata.

Při návrhu a nastavení parametrů analogových obvodů se musí zohlednit parametry jako např. velikost obvodu, počet součástek, cena, provozní podmínky, výsledný výstup obvodu a další. Návrh a analytické řešení obvodů není triviální [8]. Proto se zde využívá zjednodušování obvodu a přepisování do náhradních schémat, které představují ekvivalent původního obvodu, pomocí kterých se daný obvod řeší. Mezi metody, které řeší obvod pomocí zjednodušení patří např. princip superpozice, Theveninův a Nortonův teorém. Současně se pro výpočet obvodu používají i další metody, kterými jsou např. Kirchhoffovy zákony, metoda smyčkových proudů či uzlových napětí nebo i postupné zjednodušování obvodu, pomocí kterého lze obvod zjednodušit.

Tento proces je velmi náročný a také zdlouhavý. Proto je snaha proces automatizovat, alespoň částečně. Jednou z možností by mohl být evoluční návrh s pomocí evoluční opti-



(a) Původní obvod.



(b) Zjednodušený obvod.

Obrázek 2.1: Zjednodušte obvod při použití ekvivalence lineárního zdroje proudu a napětí a ekvivalentních náhrad sériově a paralelně řazených prvků [8].

malizace, kde se využívají evoluční algoritmy. Z tohoto důvodu je cílem této práce ověřit možnosti evolučních algoritmů při evoluční optimalizaci analogových obvodů a toto ověření bude zkoumáno na zesilovačích a oscilátorech. Tématika analogových obvodů, včetně zesilovačů a oscilátoru, je zpracována v publikaci [13], [12] a [6].

2.1 Zesilovače

Jak již název napovídá, úkolem zesilovače je zesílit jeho vstupní signál, tzn. mít na jeho výstupu zesílený signál, přičemž tento signál musí být bez zkreslení a šumu. Zesilovače můžeme rozdělit do mnoho kategorií jako např. podle druhu vstupního signálu, použitých aktivních součástek nebo zapojených tranzistorů. Nicméně nejzákladnější rozdělení je podle pracovních tříd zesilovačů 2.1.2, které charakterizují pracovní vlastnosti. Jedná se o jednu z nejrozšířenějších skupin analogových obvodů, jejichž základní stavební prvek se používá bipolární nebo unipolární tranzistor, případně elektronky. Existuje více druhů zesilovačů, jako jednostupňové, vícestupňové, operační a další. Pro jednoduchost jednostupňového zesilovače s bipolárním tranzistorem byl vybrán pro ověření evoluční optimalizace.

2.1.1 Parametry zesilovačů

Mezi základní parametry zesilovačů patří jeho zesílení,

$$A_U = \frac{U_2}{U_1} > 1[-] \quad (2.1)$$

nebo

$$A_I = \frac{I_2}{I_1} > 1[-] \quad (2.2)$$

kde A_U/A_I je napěťové/proudové zesílení zesilovače, U_2/I_2 je výstupní napětí/proud a U_1/I_1 je vstupní napětí/proud. Dále se definuje jeho šířka pásma, kterou lze vypočítat jako:

$$B = f_2 - f_1 [Hz] \quad (2.3)$$

f_1 je dolní mezní kmitočet a f_2 je horní mezní kmitočet. Mezní kmitočet je kmitočet, při kterém poklesne přenos zesilovače právě o 3 dB [13].

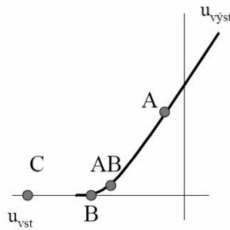
Účinnost zesilovače je dalším důležitým parametrem, který určuje, jaká část příkonu bude přeměněna na užitečný výstupní signál a velikost energetické ztráty:

$$\eta = 100 * \frac{P_2}{P_1 + P_0} [\%] \quad (2.4)$$

P_1 je výkon vstupního signálu, P_2 výkon výstupního signálu a P_0 je výkon dodávaný napájecím zdrojem.

2.1.2 Pracovní třídy zesilovačů

Toto rozdělení zesilovačů se určuje podle nastavení pracovního bodu tranzistoru, který odpovídá určitým pracovním podmínkám. Pracovní bod je bodem VA-charakteristiky, je určený velikostí kolektorového proudu a kolektorovým napětím. Kolektorový proud je ovlivňován bázovým proudem a tedy pomocí bázového proudu se nastavuje pracovní bod. Tohoto se docílí nastavením předřadného odporu či vstupního napěťového děliče.



Obrázek 2.2: Základní pracovní třídy zesilovačů [9].

Třída A

Zesilovače třídy A mají pracovní bod v lineární části charakteristiky, který se obvykle nachází uprostřed výstupní charakteristiky. Zesilovač zpracovává celou periodu vstupního signálu (celý signál). Tyto zesilovače mají velmi malé zkreslení a všeobecné použití. Nicméně zesilovače mají nízkou účinnost, teoretická účinnost se udává kolem $\eta = 50\%$, ovšem praktická účinnost se blíží k $\eta = 25\%$. Toto je způsobeno další nevýhodou, kdy aktivním prvkem protéká proud i bez vstupního signálu.

Třída B

V tomto případě je pracovní bod umístěn v bodě zániku kolektorového proudu. Kvůli tohoto umístění pracovního bodu aktivní prvek zpracovává pouze polovinu periody vstupního signálu. Mezi výhody patří vysoká účinnost, kde teoretická je $\eta = 75\%$ a praktická účinnost se pohybuje kolem $\eta = 50\%$. Nevýhodou je, že tyto zesilovače trpí zkreslením výsledného signálu. Obvykle se tyto zesilovače zapojují ve dvojici paralelně, kde jeden zesilovač zesiluje kladnou polovinu periody vstupního signálu a druhý zesilovač zápornou.

Třída AB

V třídě AB je pracovní bod položen v lineární části charakteristiky, ale blíže k bodu zániku kolektorového proudu (v ohybu charakteristiky). Z tohoto plyne, že při dostatečně velkém vstupním signálu je část jedné poloviny periody nezpracovávána. Jde o jakousi kombinaci třídy A a třídy B a vychází z toho i vlastnosti, kdy účinnost je větší než u třídy A a nižší, než u třídy B. U zkreslení výstupního signálu je to obráceně, kde zkreslení je menší než u třídy B, ale vyšší než u třídy A.

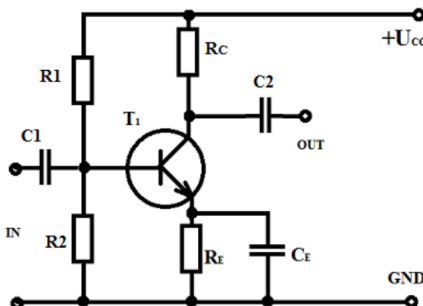
Třída C

Poloha pracovního bodu v třídě C je až v oblasti zániku kolektorového proudu. Tyto zesilovače zpracovávají méně než polovinu periody vstupního signálu. Je zde dosahováno vysoké účinnosti, kdy teoretická se blíží k $\eta = 100\%$ a praktická účinnost je kolem $\eta = 75\%$. Ovšem dochází k velkému zkreslení výstupního signálu.

2.1.3 Jednostupňové zesilovače

Jedná se o nejjednodušší zapojení zesilovače a proto byl vybrán jako testovací pro testy evoluční optimalizace. Lze ho použít jako proudový zesilovač, ale obvykle se zapojuje jako zesilovač napěťový. Tento typ zesilovače má horší vlastnosti než některé jiné typy zesilovačů.

Jeden z hlavních nedostatků základního zapojení je zahřívání tranzistoru při průchodu elektrického proudu. Následkem toho je změna pracovního bodu a tudíž vlastnosti zesilovače, kdy by mohlo dojít k poklesu až ztrátě zesilovacích vlastností, či až k poškození tranzistoru. Proto je nutné pracovní bod stabilizovat, čehož se docílí pomocí tzv. stabilizačních obvodů.



Obrázek 2.3: Jednostupňový tranzistorový zesilovač se společným emitorem. [4]

V konkrétním případě 2.3 se jedná o zapojení se společným emitorem a zesilovač patří pod třídu A. Jako stabilizační obvod je zde zvoleno tzv. můstkové zapojení. Pomocí rezistoru R_1 a R_2 je realizován dělič napětí, pomocí kterého se nastavuje pracovní bod. Při zvýšení teploty tranzistoru se sám začne více otevírat a tím začne proudit větší proud I_B , I_C a I_E . Takto se i zvýší úbytek napětí na R_E , tedy U_{RE} . Tímto musí dojít k úbytku napětí U_{BE} , které je dáno napěťovým děličem. Úbytek U_{BE} vede ke snížení proudu na bázi tranzistoru a tím pádem se zmenší i proud I_C , který byl původně zvýšen. Kondenzátor C_E má za úkol způsobení zkratu pro střídavou složku proudu a tak se neuplatní zpětná vazba stabilizačního obvodu. Kondenzátory C_1 a C_2 filtrují stejnoměrnou složku proudu a rezistor R_C určuje strmost zátěžové přímky.

2.2 Oscilátory

Oscilátory obecně jsou zařízení, které za určitých podmínek generují periodické průběhy fyzikálních veličin. V elektronice jsou oscilátory elektrické obvody, které jsou zdrojem harmonického periodického signálu, kde signál má netlumené kmity. Tzn., že mají stabilní amplitudu kmitů. Existují i zdroje neharmonického periodického signálu, které se nazývají generátory.

Oscilátory jsou charakterizovány amplitudou a tvarem výstupního napětí, frekvencí a vnitřním odporem. Základní vlastnost, kterou musí splňovat, je stabilita kmitočtu generovaného signálu. Jedná se o míru změny frekvence s časem a číselně se udává jako podíl maximální frekvence Δf_0 a nastavenou jmenovitou výstupní frekvenci f_0 za určitý časový interval, tedy zapsáno v rovnici:

$$s = \frac{\Delta f_0}{f_0} \quad (2.5)$$

Mezi nejčastěji používané oscilátory patří tzv. zpětnovazební oscilátory, které obvykle využívají kladnou zpětnou vazbu. Principem těchto oscilátorů je přenesení části energie z výstupu na vstup, kdy při překročení jisté hranice dojde ke vzniku kmitů. Pro vznik oscilace je taky potřeba splnit tyto podmínky:

1. Amplitudová - přenos zpětnovazební smyčky je roven jedné, $\beta A = 1$, kde A je napěťové zesílení signálu a β určuje, jaká část výstupního signálu se vrátí zpět na vstup.

2. Fázová - součet všech fázových posunů musí být roven 2π nebo násobku 2π .

Každý takový oscilátor se skládá ze dvou částí. A to ze zesilovače a řídicího obvodu (tzv. zpětnovazební obvod), který zajišťuje zpětnou vazbu. Oscilátory lze dělit podle různých vlastností a zapojení, jako např. podle průběhu výstupního signálu, podle frekvence výstupního signálu, podle typu aktivního prvku, podle typu zpětnovazebního členu a další. Právě rozdělení podle typu zpětnovazebního členu je důležité z pohledu této práce. Podle toho se dělí oscilátory na RC a LC.

2.2.1 Zpětnovazební LC oscilátory

LC oscilátory jsou jedny z nejpoužívanějších zapojení. Základem je paralelní rezonanční obvod [17], který je tvořen paralelním zapojením cívky a kondenzátoru. Pokud se na obvod připojí zdroj napětí, tak se na cívce vytvoří magnetické pole. Po odpojení zdroje se v cívce indukuje napětí opačné polarity, které nabíjí kondenzátor. Po poklesu napětí na cívce se začne kondenzátor vybíjet a tím opět začne vznikat magnetické pole v cívce. Případně se aplikuje opačná možnost, kdy se od zdroje nabije kondenzátor, který se po odpojení začne vybíjet a tím opět na cívce vznikne magnetické pole a děj se opakuje. Oběma způsoby vytváří tzv. rezonanční kmitočet, který lze vypočítat vztahem:

$$f = \frac{1}{2\pi LC} \quad (2.6)$$

Problémem je, že v reálném obvodu vznikají ztráty energie na odporu vynutí cívky, v jádře cívky, na kondenzátoru a dokonce i v obvodu samotném. Proto se vytvářenému sinusovému průběhu střídavého napětí stále zmenšuje amplituda, vytvářejí se tzv. tlumené kmity. Tato vzniklá ztráta se musí řešit stálým dodáváním energie, tzn. připojením aktivního prvku k obvodu. V případě zpětnovazebních oscilátorů je tímto prvkem zesilovač s kladnou zpětnou vazbou, který společně s rezonančním obvodem tvoří LC oscilátor. Mezi příklady těchto oscilátorů se řadí např. Meissnerovo zapojení, Hartleyův oscilátor nebo Colpittsův oscilátor.

2.2.2 Zpětnovazební RC oscilátory

RC oscilátory jsou vhodné k tvorbě nízkofrekvenčních kmitů a jejich výhodou je, že jsou přeladitelné. RC oscilátor má základ tvořen pomocí zapojení rezistorů a kondenzátorů, které určují frekvenci oscilací. Toto zapojení je sestaveno tak, aby tvořilo pasivní čtyřpól, který společně se zesilovačem s kladnou zpětnou vazbou tvoří RC oscilátor.

Na základě pasivního čtyřpólu se dělí RC oscilátory na oscilátory s posuvem fáze a můstkové oscilátory. Zapojení s posuvem fáze je obvykle tvořeno jednostupňovým zesilovačem, jehož tranzistor je v zapojení se společným emitorem, a kaskádou RC členu. Zesilovač posune fázi o 180° , proto je potřeba, aby kaskáda posunula fázi taky o 180° . Proto je kaskáda obvykle tvořena třemi členy, kde na každý člen připadá posun fáze o 60° . Výpočet frekvence je závislý na použitém typu článků, kde:

$$f = \frac{\sqrt{6}}{2\pi RC} \quad (2.7)$$

v případě integračních článků, pokud jsou použité derivační články, tak se frekvence vypočte jako:

$$f = \frac{1}{2\pi RC\sqrt{6}} \quad (2.8)$$

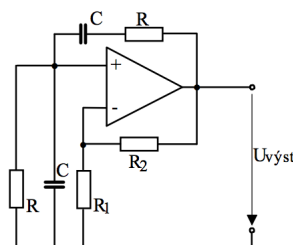
V můstkových oscilátorech je ve zpětné vazbě zapojena kombinace RC členů, které tvoří můstkové zapojení. Tímto se zavádí mezi vstup a výstup zesilovače kladné zpětné vazby, které zajišťují vznik oscilace, a záporné zpětné vazby, které zajišťují stabilní amplitudu kmitu. Díky toho se dosáhne lepší kmitočtové stability a menšího zkreslení signálu. Tyto oscilátory se také snadněji přeladují. Kmitočet je dán selektivní zpětnou vazbou, pomocí selektivních RC článku, které se používají jako pásmová propust. Příkladem je použití Wienova článku, nebo jako pásmová zadrž, kde se používá T-článek.

Oscilátor s Wienovým článkem

Je oscilátor využívající dvě smyčky zpětné vazby a to kladnou a zápornou. Základním prvkem je zde Wienův článek, který je serioparalelní zapojení stejných odporů a kondenzátorů. Využívá principu pásmové propusti kolem kritické frekvence f_0 , kterou lze vypočítat jako:

$$f_0 = \frac{1}{2\pi RC} \quad (2.9)$$

Při této frekvenci má zapojení největší přenos, který je roven $\frac{1}{3}$, čili výstup článku je $V_{out} = \frac{1}{3}V_{in}$. Současně na této frekvenci nedochází k posunu vstupní fáze. Další částí tohoto oscilátoru je zesilovač, kde je vhodné k Wienovému článku použít operační zesilovač. Zesilovač musí mít zvětšení větší nebo rovnou 3 a zesilovač musí být neinvertující, protože při frekvenci f_0 nedochází k posuvu fáze. Toto vše lze zajistit dvoustupňovým zesilovačem, kdy každý tranzistor zesilovače posune fázi o 180° . Kladná zpětná vazba zajišťuje splnění fázové podmínky pro vznik oscilací, kdežto záporná zpětná vazba se stará o stabilizaci amplitudy a nastavení zesílení zesilovače.



Obrázek 2.4: Příklad RC oscilátoru s Wienovým článkem. Převzato a upraveno z [5].

Na obrázku 2.4 lze vidět základní RC oscilátor s Wienovým článkem. Rezistory R spolu s kondenzátory C tvoří můstkové zapojení, tedy kladnou zpětnou vazbu. Rezistory R_1 a R_2 určují zesílení zesilovače a tvoří zápornou zpětnou vazbu. Tento oscilátor by bylo potřeba doplnit obvodem pro stabilizaci amplitudy, toho by se dalo docílit přidáním žárovky za rezistor R_1 , nebo přidáním termistoru za rezistor R_2 .

Tento obvod byl vybrán k experimentování jakožto zástupce oscilátoru. Důvodem je si vyzkoušet experiment na složitějším obvodu, oscilátor s Wienovým článkem není o nejjednodušší zapojení oscilátoru, ale současně jde o zajímavé zapojení, využívající dvě smyčky zpětné vazby .

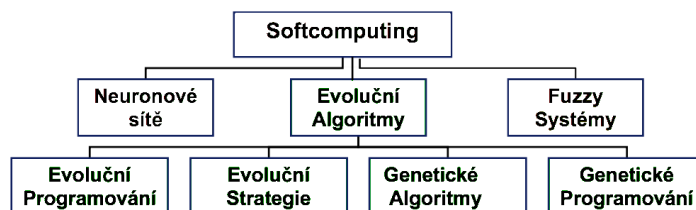
Kapitola 3

Evoluční algoritmy

Evoluční algoritmy jsou jednou ze skupin strojového učení a jsou založené na biologických principech, konkrétně se inspiřují Darwinovým přirozeným výběrem. Jde tedy o matematické postupy využívající modely evolučních procesů v přírodě.

Tyto matematické postupy jsou postavené na společném principu, kdy se pracuje s množinou možných řešení určitého problému, místo toho, aby se pracovalo s konkrétním řešením problému. Tato množina se postupně vylepšuje pomocí nových řešení, která vznikají výběrem původních řešení a následně jejich kombinací mezi sebou, mutací a eliminací horších řešení.

Na obr. 3.1 lze vidět, že EA se vedle neuronových sítí a fuzzy systému řadí do oboru inteligentních výpočtů. Na obrázku lze také vidět rozdělení EA, kde genetické algoritmy a evoluční strategie jsou starší typy EA a evoluční programování s genetickým programováním jsou novější typy EA. V této práci budou rozebrány pouze genetické algoritmy a evoluční strategie.



Obrázek 3.1: Zařazení a rozdělení EA. [16]

Tyto algoritmy jsou velmi robustní a vhodné pro problémy, kde dochází k prohledávání velkého prostoru možných řešení, u kterých tradiční metody řešící problém selhávají, kde nelze vykonat matematickou analýzu problému nebo je mnoho extrémů, kritérií a omezujících podmínek.

Používají se tedy při problémech, kde analytické či heuristické metody nejsou dostačující, zejména v NP úplných problémech. Využití tedy naleznou v různých oblastech, jako např. numerických a kombinatorických optimalizacích, plánování a řízení, inženýrských návrzích, dolování dat, strojovém učení, umělé inteligenci atd. Mezi konkrétní příklady použití lze uvést nalezení nejkratší Hamiltonovské cesty v grafu, úloha obchodního cestujícího a mnohé další. Tématika evolučních algoritmů, včetně genetického algoritmu a evoluční strategie, je rozebrána v publikacích [11], [7] a [16].

3.1 Ch. Darwin, Darwinovská evoluce a neodarwinismus

Charles Darwin (1809 - 1882) byl britským přírodovědcem a zakladatelem evoluční biologie. Jeho největším dílem byla kniha *O vzniku druhů cestou přírodního výběru aneb zachování zvýhodněných odrůd v boji o život* [18], která byla vydaná v roce 1859. Toto dílo pojednává o Darwinově teorii, podle které vzniká mezi jedinci konkurence a boj o přežití, pokud nastane situace, kdy vznikne větší počet jedinců, než je možné v daném prostředí uživit. V rámci dané populace přežijí silnější jedinci (s lepšími dědičnými vlastnostmi) a budou se dále rozmnožovat, kdežto slabší jedinci zaniknou. Toto zajistí přežití nejsilnějších jedinců a tím postupné přizpůsobení a rozvoj biologického druhu na dané prostředí. Darwinova teorie se skládá ze tří prvků:

1. Přirozený výběr - proces, kde silnější jedinci (jedinci s větší pravděpodobností na přežití) jsou s větší pravděpodobností vybíráni k reprodukci, než ostatní jedinci.
2. Náhodný genetický drift - náhodné události jedinců, které ovlivňují populaci. Jde například o náhodnou smrt silného jedince, který se nezúčastnil produkčního procesu, nebo náhodnou mutaci genetického materiálu. Náhodný genetický drift nejvíce ovlivňuje malé populace.
3. Reprodukční proces - jde o proces, kdy se z rodičů vytvářejí potomci. Nový potomek zdědí svou genetickou informaci od svých rodičů. Proces probíhá tak, že z genetické informace dvou jedinců se náhodně vyberou části chromozómů, z kterých je vytvořena genetická informace nového jedince.

Darwinova teorie nedokáže vysvětlit mechanismus proměnlivosti. Gregor Mendel (1822 - 1884) [18], byl přírodovědec a zakladatel genetiky, který doplnil tuto chybějící část v Darwinově teorii. Spojením Darwinovy teorie a Mendelových základů genetiky vznikla moderní syntéza neboli Neodarwinismus [1], který je v současné době hlavní paradigma evoluční teorie. Podle Neodarwinismu probíhá evoluce na základě přirozené selekce náhodných genetických mutací.

3.2 Základní části EA

Všechny EA mají podobné části, které jsou základem každého evolučního algoritmu. Jednou z nejdůležitějších částí je reprezentace jedinců. Tj. způsob zakódování jedinců. U této složky jsou dva základní pojmy, a to **fenotyp** a **genotyp**. Genotypem rozumíme způsob kódování řešení, kdežto fenotyp je reprezentace genotypu. Např. genotyp je binární číslo, které kóduje dekadické číslo a fenotypem je dekadické číslo, jenž je hodnotou tohoto genotypu (genotyp=1000, fenotyp=8). Necht je v dalším příkladu genotyp genetická informace, fenotyp je potom jedinec, který vznikl podle této genetické informace.

Dalším základním stavebním kamenem EA je tzv. fitness funkce ohodnocující jedince, tedy říká, na kolik je jedinec kvalitní řešení. Jedinci se sdružují v populaci, tedy jde o nějaké seskupení jedinců, se kterým EA pracuje a tudíž je nedílnou součástí každého EA. K populaci se váže další část a to selekce rodičovských jedinců, kde dochází k výběru jedinců z populace. Jedinci zvolení pomocí selekce budou pomocí variačních operátorů křížení a mutace tvořit nové jedince (tzv. potomky). Na konci se pomocí obnovy populace vytvoří nová populace ze starých a nových jedinců.

Jednotlivé složky jsou odlišné pro různé typy evolučních algoritmů. Některé specifikace komponentů budou popsány níže u genetických algoritmů a evolučních strategií.

3.3 Průběh EA

Před začátkem vykonávání EA je potřebné zvolit vhodný genotyp jedinců. Na začátku se vytvoří počáteční populace tvořená náhodně vygenerovanými jedinci. Všechny tyto jedince je nutné ohodnotit fitness funkcí, která určuje kvalitu jedince. Od hodnoty fitness funkce se určuje pravděpodobnost vstupu do reprodukčního procesu, kde jedinci s lepší fitness hodnotou mají větší pravděpodobnost, ale i slabší jedinci s menší pravděpodobností mají možnost se do procesu dostat. Po vyhodnocení fitness probíhá selekce jedinců určených k reprodukci. Tito jedinci vstupují do reprodukce, kde pomocí variačních operátorů dochází ke křížení nebo mutacím, což dodává populaci rozmanitost. Z původní populace se vytvoří nová v kombinaci s novými jedinci.

Nejdůležitější prvek je reprodukční proces, který je založen na selekci rodičovských jedinců a variačních operátorech. Vhodně zvolený reprodukční proces vede k lepším výsledkům populací, ohodnocení jejich jedinců a tedy k rychlejšímu nalezení řešení.

```
begin
t := 0; // nastavení počátečního času
initpopulation P(t); // inicializace, náhodné generování populací jedinců
evaluate P(t); // ohodnocení všech počátečních jedinců populace
while not finished do // test na ukončovací kritérium (čas, vhodnost, atd.)
    t := t + 1; // inkrementace čísla populace
    P' := selectpar P(t); // selekce rodičů
    recombine P'(t); // křížení vybraných rodičů
    mutate P'(t); // náhodná mutace potomků
    evaluate P'(t); // ohodnocení nově vzniklých potomků
    P := survive P,P'(t); //obnova populace
od
end
```

Listing 3.1: Pseudokód obecného schématu EA. [16]

K průběhu EA se taky vážou ukončující podmínky, kdy evoluční algoritmus lze ukončit. Např. při nalezení potomka dostačující s fitness funkcí, dosáhnutí maximálního počtu generací, vypršení maximální doby běhu EA a další. Toto je výpis pouze základních podmínek, ale i z nich je patrné, že EA nemusí najít nejlepší výsledek, ba dokonce nemusí dojít k žádnému výsledku.

3.4 Genetické algoritmy

Genetické algoritmy jsou nejpoužívanějším typem evolučních algoritmů. Nejvíce se tyto algoritmy uplatňují v optimalizačních úlohách, při hledání globálního extrému jedno a více-dimenzionálních funkcí.

3.4.1 Genotyp a kódování

V genetických algoritmech se jedinci označují jako chromozomy. Chromozom se dělí na geny, které jsou v chromozomu lineárně uspořádány. Gen nabývá různých hodnot, které se nazývají alely. Zvláštním typem chromozomu jsou Messy-chromozomy, které mají proměnlivou délku a jejich geny jsou definované indexem a hodnotou. Pokud I je množina indexů a N množina hodnot, které může M-chromozom, pak je gen prvkem z množiny $Q = I \times N$, tedy dvojice (i, n) , kde $i \in I$ a $n \in N$.

Nejčastějším typem kódování genotypu v GA je binární kódování. V tomto kódování nabývá gen hodnotu z množiny 0,1. Jde o velmi jednoduché a nenáročné kódování. Problém tohoto kódování je, že při malé změně fenotypu jedince chceme pouze malou změnou genotypu a naopak, toto ale u binárního kódu nemusí platit, pokud se budou hodnoty měnit na vysoké pozici v chromozomu. Tento problém se dá řešit pomocí Grayova kódu.

Chromozom A	1	0	1	0
Chromozom B	1	0	1	1

Obrázek 3.2: Ukázka binárního kódování.

Dalším typem kódování může být permutační kódování. Toto kódování je tvořeno řadou čísel, kde každé číslo představuje pozici v pořadí. Je vhodné například pro řešení problému obchodního cestujícího.

Chromozom A	3	1	2	4
Chromozom B	2	4	1	3

Obrázek 3.3: Ukázka permutačního kódování.

Existují i další druhy kódování jako používající pro reprezentaci celočíselná čísla, reálná čísla, strom, matici, pole, vektor či jiné struktury.

3.4.2 Selektce

Selektce je proces výběrů chromozomů, které se budou účastnit reprodukčního procesu. Větší pravděpodobnost výběru mají chromozomy s lepší hodnotou fitness funkce. Nieméně je nutné, aby možnost dostali i slabší jedinci a byli vybíraní, aby nedošlo k lokální konvergenci.

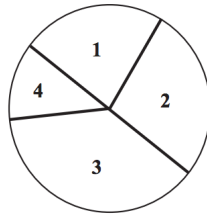
Proporcionální selektce (ruletová selektce)

První algoritmus pro výběr chromozomů, který dává největší šanci nejlepšímu chromozomu. Pravděpodobnost, že bude jedinec vybrán, je dána vzorcem:

$$p_i = \frac{f_i}{\sum_{j=0}^N f_j} \quad (3.1)$$

Principiálně ruletová selektce funguje na bázi točení rulety. Ruleta se rozdělí na dílky, které představují určité chromozomy. Velikosti dílku je úměrná fitness hodnotě daného chromozomu. Ruletou se zatočí a vybere se chromozom, který náleží dílku, na kterém se ruleta zastavila.

Problémem této metody je, že může docházet k vybírání stále stejných chromozomů. K tomu dojde, pokud jsou v populaci chromozomy s vysokou fitness hodnotou (stačí jeden takový chromozom), tak zaberou téměř celou ruletu a ostatní chromozomy mají velmi malou šanci.



Obrázek 3.4: Ukázka rulety pro 4 chromozomy. [15]

Pořadová selekce

Tato selekce je založená na ruletové selekci, jenž zamezuje problému, kterým ruletová selekce trpí. Základem je vzestupné seřazení chromozomů podle jejich fitness hodnoty. Pravděpodobnost výběru se nepočítá přímo z fitness hodnoty chromozomů, ale počítá se z pořadí chromozomů. Podle typu uspořádání se počítá pravděpodobnost výběru chromozomů, kde uspořádání může být lineární nebo exponenciální.

Turnajová selekce

Tato selekce také zamezuje problému s konvergencí. Slabší jedinci mají větší šanci, než v ruletové selekci a chromozomy se nemusí řadit jako u pořadové selekce. Princip této selekce spočívá v souboji chromozomů. Na začátku se vybere N chromozomů z populace, které mezi sebou soupeří. Vítězem a tedy i vybraným chromozomem se stává chromozom, který má nejlepší hodnotu fitness.

Z pravidla se vybírají 2 jedinci k souboji, tedy $N = 2$. Tento výběr chromozomů k boji probíhá náhodně.

Elitismus

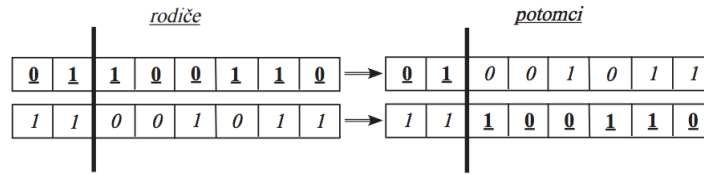
Nejde úplně o selekční metodu, protože může jít pouze o doplnění ostatních selekčních technik. Elitismus znamená výběr N nejlepších jedinců z populace.

3.4.3 Křížení

Základní variační operátor pro GA, který tvoří nové chromozomy. Konkrétně ze dvou rodičů tvoří dva potomky, které nesou geny od obou rodičů.

Jednobodové křížení

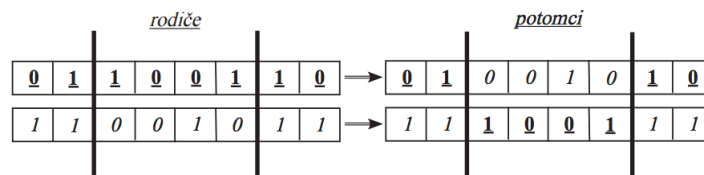
Nejjednodušší způsob křížení, kdy se náhodně vybere jeden bod/gen, od kterého se zbylé geny chromozomů rodičů navzájem prohodí, a tím vzniknou potomci. Potomek má tedy první část chromozomu od jednoho rodiče a druhou část od druhého rodiče, jako to je znázorněno na obr. 3.5.



Obrázek 3.5: Jednobodové křížení. [15]

Vícebodové křížení

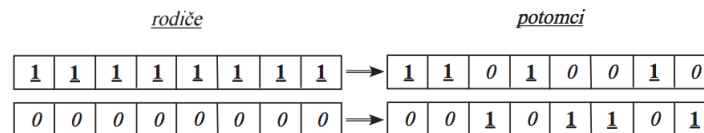
Principiálně se jedná o stejné křížení jako jednobodové křížení, ale na chromozomu se náhodně vybere více bodů křížení, které chromozom rozdělí na části, které se mezi sebou navzájem prohazují. Tento proces je vyobrazen na obr. 3.6.



Obrázek 3.6: Vícebodové křížení. [15]

Uniformní křížení

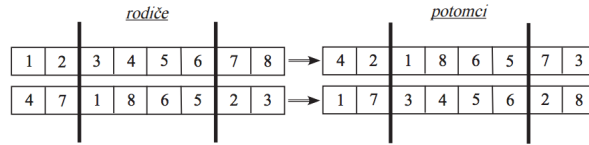
Je to křížení, při kterém se náhodně vyberou geny (jejich pozice), které se mezi sebou prohodí, jako tomu je na obr. 3.7.



Obrázek 3.7: Uniformní křížení. [15]

Permutační křížení

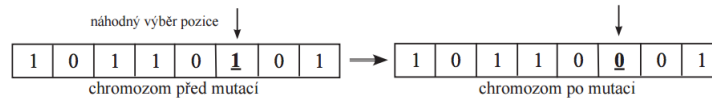
Používá se pro permutační kódování genotypu. Zabraňuje kolizím alel v chromozomu, protože každá jedna alela se v chromozomu nemůže vyskytovat vícekrát než jednou. Princip je podobný jako u jednobodového křížení, jen se chromozomy rozdělí na 3 části pomocí dvou dělících bodů. Prostřední část se vzájemně prohodí a krajní části se doplňují tak, aby nevznikla kolize. Tzn., že se postupně prochází krajní části rodiče, a pokud alela ještě není v potomkovi, tak se tam nastaví. Pokud tam již hodnota je, tak se prohodí s druhým rodičem. Křížení je znázorněno na obr. 3.8.



Obrázek 3.8: Permutační křížení. [15]

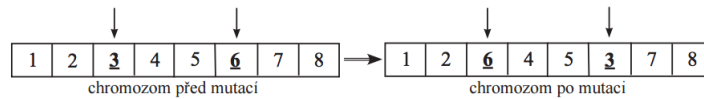
3.4.4 Mutace

Mutace je variační operátor s malou pravděpodobností výskytu. V principu funguje tak, že se náhodně vyberou některé geny, kterým se změní alela. Např. v bitovém kódování se bit alely invertuje.



Obrázek 3.9: Mutace chromozomu s binárním kódováním. [15]

Existují i speciálnější mutace, například pro permutační kódování, kde nelze měnit hodnotu alely. Zde se mutace provádí výběrem dvou genů, které se navzájem prohodí. Tímto se zabrání kolizí alel.



Obrázek 3.10: Mutace chromozomu s permutačním kódováním. [15]

Mutace může mít zanedbatelný vliv, ale taky může mít zničující dopad na chromozom. Pravděpodobnost výskytu mutace musí být vhodně nastavená, protože pokud je nízká, tak neovlivní evoluci, naopak pokud je vysoká, způsobuje nestabilitu evoluce.

3.4.5 Obnova populace

Tato operace určuje dynamiku prohledávání stavového prostoru. Existují dva základní přístupy obnovy populace:

1. Generativní GA s úplnou obnovou
2. Částečná obnova

V případě generativního GA s úplnou náhradou dochází k tzv. vymírání rodičů. Tato obnova nahradí starou populaci zcela novou populací, která je tvořena z nových potomků.

Při částečné obnově dochází k výběru jednoho z potomků a tento potomek nahradí nejslabšího jedince ze staré populace. Tímto se vytváří nová populace, kde postupně a pomalu populace vylepšuje. Pro volbu tohoto jedince lze zvolit techniky jako ruletová selekce, pořadová selekce, turnajová selekce nebo elitismus.

Tyto dva přístupy se dají kombinovat, kdy se vybere 20% až 50% z potomků, kteří se dosadí do staré populace a tím vznikne nová populace.

3.5 Evoluční strategie

Evoluční strategie je více než 30. let starým typem evolučních algoritmů, který se zaměřuje na inženýrské úlohy. ES je určena hlavně pro práci s desetinnými čísly a je vhodná například pro strojní nebo stavební inženýry. Reprezentace/kódování jedinců se dělá jako kódování pomocí reálných čísel. Selektce rodičů pro reprodukci je náhodná selektce s uniformní rozložením. U křížení jsou dvě možnosti a to diskrétní nebo aritmetické křížení. Nicméně křížení v ES je používáno s nízkou četností, či v některých případech vůbec. Naopak mutace je hlavním variačním operátorem v ES, kde se používá Gaussovy distribuce pravděpodobnosti. Obnovit populaci lze pomocí plus strategie ($\mu + \lambda$) nebo čárková strategie (μ, λ). V ES lze použít některé speciální postupy, jako např. auto adaptace vlastní mutace.

Tato charakteristika je současně i odlišnost od GA, jednotlivé techniky budou v této kapitole popsány.

3.5.1 Genotyp a kódování

Evoluční strategie na rozdíl od genetického algoritmu, který má více možných reprezentací, se v této technice používá kódování pomocí reálných čísel, tedy $(x_1, x_2, x_3, \dots, x_n)$, kde $x_i \in R$.

Chromozom A	-3.04	1	2.22	-9.24
Chromozom B	4.6	-2.2	-1.14	3.82

Obrázek 3.11: Ukázka kódování v evoluční strategii.

3.5.2 Křížení

Jako u genetických algoritmů existuje více druhů křížení, která jsou specifická pro daný typ problému. Oproti GA zde neplatí, že z křížení vždy vzniknout 2 potomci.

Diskrétní křížení

V tomto křížení se vytvoří jedinec, jehož některé geny se převezmou od jednoho rodiče a zbylá část se vezme od druhého rodiče. Jinými slovy se pro každý gen náhodně rozhodne, od kterého z rodičů se gen převezme. Z mnoha existujících křížení lze vybrat jako příklady diskrétní křížení a křížení průměrem.

-3,91	8,84	-3,98	-2,51	-5,76	-1,40	-3,33	8,26	rodič_1
▼			▼		▼	▼		
-3,91	8,46	-1,33	-2,51	6,66	-1,40	-3,33	-0,40	potomek
	▲	▲		▲			▲	
6,91	8,46	-1,33	-5,66	6,66	-0,75	9,13	-0,40	rodič_2

Obrázek 3.12: Diskrétní křížení. [11]

Křížení průměrem

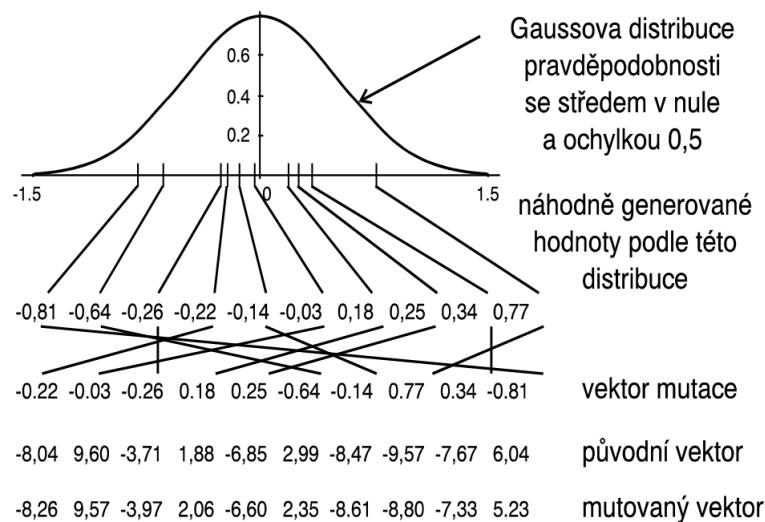
Křížení je založeno na principu počítání aritmetického průměru, kde alela pro gen nového jedince je vypočítaná jako aritmetický průměr alel rodičů na daném genu.

-3,91 8,84 -3,98 -2,51 -5,76 -1,40 -3,33 8,26	rodič_1
6,91 8,46 -1,33 -5,66 6,66 -0,75 9,13 -0,40	rodič_2
1,50 8,65 -2,67 -4,09 0,45 -1,08 2,90 3,93	potomek
	= $\frac{\text{rodič}_1 + \text{rodič}_2}{2}$

Obrázek 3.13: Křížení průměrem. [11]

3.5.3 Mutace

Na rozdíl od GA je v ES mutace hlavním variačním operátorem, který se aplikuje na všechny geny daného chromozomu. K účelům mutace se využívá Gaussovy distribuce pravděpodobnosti změny s nulovou střední hodnotou a směrodatnou odchylkou σ . Mnohdy se používá normované rozdělení formalizované zápisem $N(0, 1)$.



Obrázek 3.14: Mutace pomocí Gaussovy distribuce pravděpodobnosti. [16]

Podle distribuce se vytvoří vektor náhodně vygenerovaných hodnot. Tento vektor se transformuje na vektor mutace tak, že se přehodí jednotlivé položky vektoru mezi sebou (hodnoty se promíchají). Tento vektor mutace se přičte k alelám jednotlivých genů rodičovského chromozomu, a tím vznikne nový chromozom.

3.5.4 Obnova populace

Evoluční strategie má pro obnovu populace dvě strategie:

1. Plus strategie $(\mu + \lambda)$
2. Čárková strategie (μ, λ)

V plus strategii $(\mu + \lambda)$ (velikost rodičovské populace + velikost populace potomků) dochází k obnově tak, že N rodičů vygeneruje M potomků, tito rodiče a potomci se spojí do jedné množiny a pomocí elitismu se vybere N jedinců, kteří budou tvořit novou populaci. V původní implementaci se jednalo o strategii 1+1, tzn. že populace byla tvořena jedním jedincem, a pokud potomek měl lepší fitness ohodnocení, tak nahradil rodiče v populaci. V opačném případě rodič zůstal v populaci a potomek se nepoužil.

V čárkové strategii (μ, λ) tvoří novou populaci pouze vytvoření potomci. Rodiče tedy mohou být nahrazeni i jedinci se slabším fitness ohodnocením. Technika umožňuje opustit lokální minima a používá se v obtížných optimalizačních úlohách.

3.5.5 Autoevoluce řídicích parametrů

Autoevoluce [7] je jedna z technik, kdy je chromozom tvořen cílovými i řídicími parametry. Všechny tyto parametry podléhají evoluci, a tak evoluce ovlivňuje řízení jednotlivých chromozomů. Konkrétně jde o uchování a ovlivňování mutačního kroku, čímž se umožní, aby mutační krok byl variabilní v průběhu evoluce.

Nejjednodušší případ je, pokud pro všechny geny v chromozomu je stejný mutační krok, tedy jediný mutační krok. Teda máme vektor $v = ((x_1, \dots, x_n), \sigma)$, kde (x_1, \dots, x_n) jsou jednotlivé geny (hodnoty) a σ je mutační krok. V každém iteraci evoluční strategie se při mutaci vypočte nový mutační krok, který se v mutaci použije. Je to popsáno vztahy:

$$\sigma(t+1) = \sigma(t)e^{\tau r} \quad (3.2)$$

$$x_i(t+1) = x_i(t) + \sigma(t+1)r_i, i = 1, \dots, n \quad (3.3)$$

kde n je počet genů v chromozomu, r a r_i jsou náhodné veličiny, získané pomocí Gaussovy distribuce pravděpodobnosti $N(0, 1)$. Hodnota τ je zadána uživatelem, který představuje parametr učení a ovlivňuje míru autoevoluce. Obvykle se volí hodnota dané vztahem $\tau \propto \frac{1}{\sqrt{n}}$. Pro správnou funkčnost je potřeba dodržet pořadí výpočtu, tedy jako první mutovat σ a poté x_i .

Složitějším případem je situace, kdy každý gen v chromozomu má vlastní mutační krok. Je to zcela analogové k případu stejného mutačního kroku pro všechny geny v chromozomu. V tomto případě existuje vektor $v = ((x_1, \dots, x_n), (\sigma_1, \dots, \sigma_n))$, kde (x_1, \dots, x_n) jsou jednotlivé geny (hodnoty) a $(\sigma_1, \dots, \sigma_n)$ jsou mutační kroky pro jednotlivé geny. Zde jsou vztahy popsány jako:

$$\sigma_i(t+1) = \sigma_i(t)e^{\tau' r' + \tau r} \quad (3.4)$$

$$x_i(t+1) = x_i(t) + \sigma_i(t+1)r_i, i = 1, \dots, n \quad (3.5)$$

kde r_i , r a r_i jsou opět náhodnými veličinami získané s Gassovou distribucí pravděpodobnosti $N(0, 1)$. Hodnoty τ a τ' se určují podle vztahů $\tau' \propto \frac{1}{\sqrt{n}}$ a $\tau \propto \frac{1}{\sqrt{2\sqrt{n}}}$.

Kapitola 4

Evoluční optimalizace analogových obvodů

Cílem této práce je ověřit možnosti evolučních algoritmů v optimalizaci analogových obvodů, kvůli snahy automatizovat proces návrhu a nastavení obvodu. Evoluční optimalizace analogových obvodů by mohla být jedna z možností, jak obvody nastavit. Evoluční optimalizace by mohla proti konvenčním řešením (případně i jiným přístupům) umožnit prohledat větší prostor možných řešení a navíc by mohla dosahovat zajímavých výsledků.

Při pevně dané struktuře/zapojení obvodu je snaha pomocí evolučních algoritmů nastavit parametry tohoto obvodu (jeho součástek) tak, aby se dosáhlo co nejlepšího výsledku vzhledem k referenčnímu řešení, které bylo spočteno konvenčním způsobem. Při tomto procesu nedochází ke změně zapojení prvků v obvodu. Konkrétně se budou ověřovat genetické algoritmy a evoluční strategie na analogových obvodech zesilovačů a oscilátorů.

K tomuto ověření je nutné na začátku navrhnout zadávání obvodů jakožto vstup do evolučních algoritmů. Poté je potřeba navrhnout kódování chromozomu, fitness funkce ohodnocující kandidátní řešení a implementovat samotné evoluční algoritmy.

4.1 Zadávání obvodů a jejich simulace

Pro simulaci obvodů byl zvolen program *Ngspice circuit simulator* [2], který je open source simulátor obvodů. Je vystavěn na balíčcích Spice3, Cider a Xspice, tedy na jednom z nejznámějších simulátorů a balíčcích, které umožňují ještě větší přesnost. Ngspice je použit v programech jako KiCAD, Blackboard, WeSpice, eSim a dalších.

Tato volba ovlivnila zadávání vstupního obvodu do evolučních algoritmů a i evoluční algoritmy samotné.

4.1.1 Zadávání obvodů

Program Ngspice má interaktivní prostředí, ve kterém se pomocí příkazů dá obvod nastavit a spustit simulaci. Kromě této možnosti má i možnost zadávání obvodu v podobě textového souboru, tzv. netlistu. Z tohoto textového zadávání vychází vstup pro evoluční algoritmy. Jde téměř o shodné formáty souborů s drobnými modifikacemi:

1. Je zrušena možnost zadání hodnoty prvku, které podléhají evoluci. V tomto řešení to nedává smysl, tuto hodnotu nastavuje EA.

2. Je přidáno ohraničení oblasti pro definici prvků, které budou nastavované pomocí evoluce. Pokud existují prvky, které nemají být ovlivněné evolucí, tak se definují mimo tuto oblast.
3. Byla přidána možnost u prvku podléhající evoluci obvodu definovat rozsah hodnot, kterých může prvek nabývat.
4. Byla prvků přidána možnost vazby, tzn. že pokud evoluce změní hodnotu dané součástce, tak ji změní i součástce, se kterou je propojená. Tyto dvě součástky tedy vždy mají stejné hodnoty.

```

* Components *
V1 4 0 9
Ce 0 5 0.000009
Cout 1 out 0.000000220 100-220.5
Cin in 3 56 000000220
Rc 1 4 1000
Rload 0 out 22000 2000-22000
Re 0 5 1000 1-2000 Rg
Rg 5 2 40 1-2000 Re
R2 0 3 33000 4-8
R1 3 4 47000
* End components *

Vin in 0 sin(0 .1 2k)
Q1 1 3 2 bc547c
.model bc547c NPN (BF=730 NE=1.4 ISE=29.5F IKF=80M IS=60F VAF=25 ikr=12m BR=10 NC=2
VAR=10 RB=280 RE=1 RC=40 VJE=.48 tr=.3u tf=.5n cje=12p vje=.48 mje=.5 cjc=6p vjc=.7
mjc=.33 isc=47.6p kf=2f)
.tran 1u 1.5m
.end

```

Listing 4.1: Ukázku vstupní definice obvodu.

Na ukázce 4.1 lze vidět ohraničení součástek obvodu, které podléhají evoluci. Toto vymezení je označeno pomocí značek ** Components ** a ** End components **. Komponenty obvodu mohou nabývat 6 hodnot, kde první hodnota je jméno, další 2 hodnoty definují zapojení do obvodu, čtvrtá hodnota je nastavená hodnota prvku, která se použije pro výpočet referenčního řešení. Hned za ní se nachází volitelné rozmezí udávající rozsah hodnot a jako poslední je možnost vazby na jinou součástku, v ukázce jde vidět vzájemnou vazbu rezistorů R_e a R_g .

4.1.2 Simulace obvodů

Problémem v simulaci obvodů je nutnost dávkového zpracování. Vzhledem k velkému množství simulace je potřeba zajistit co největší rychlost simulace.

Ngspice nabízí více možností, jak zadaný obvod spustit [14]. První možností je přes interaktivní prostředí pomocí zadání příkazu, ale toto řešení je pro dávkové spouštění nevhodné.

Další možností je spuštění programu s parametry, pomocí kterých se nastaví vstupní soubor s obvodem. Program výsledek zapíše opět do souboru. Toto řešení je již použitelné, ale vzniká zde velká reže na zpracování. Je nutné vytvořit soubor s obvodem, spustit program a výsledek přečíst ze souboru. Při tak velkém dávkovém volání, jako je v evolučních algoritmech, by řešení bylo značně pomalé.

Poslední rozhraní Ngspice je vystavené API se sdílenou knihovnou pro různé programovací či skriptovací jazyky. Pomocí tohoto API je možné nastavit obvod, spouštět příkazy jako v předchozích možnostech a získat výsledek simulace. Odpadá zde režie s tvorbou, zpracováním souborů a spouštěním externího programu. Jedná se o nejrychlejší a implementačně nejsnazší způsob pro dávkové zpracování. Proto se v této diplomové práci využívá tohoto rozhraní.

4.1.3 Detekce nevhodných řešení

V průběhu vykonávání EA může nastat, že EA najde kandidátní řešení, jehož obvod nelze simulovat v programu Ngspice. Mohou nastat dva chybové případy:

1. Simulace nevrátí žádný výsledek.
2. Simulace běží déle, než je nastaven práh maximální doby běhu, který je nastaven na 5 sekund.

Oba případy se v EA detekují. Takto chybový jedinec je vyloučen z populace a samotné evoluce. Místo něj je do populace zařazen jedinec, jenž vzniká jako úplně nový, tedy jako při procesu inicializace populace. Pokud by selhal i tento náhradní jedinec, tak se proces vyloučení jedince a tvorby nového jedince opakuje, dokud se nenažde jedinec, jehož obvod se dá korektně simulovat.

4.2 Kódování chromozomu

Kódování tohoto problému je velmi jednoduché. Chromozom představuje možné řešení obvodu a jednotlivé geny chromozomu představují prvky obvodu, které se nastavují. Alela genu obsahuje informace:

1. Nastavená **hodnota** prvku obvodu, jenž je reprezentována reálným číslem.
2. Informaci o **zapojení prvku** do obvodů, tato informace je klíčová k následné simulaci obvodu.
3. Nepovinnou informací je **rozsah hodnot**. Některým prvkům obvodu nemá smysl nastavovat hodnoty z celého definičního oboru reálných čísel. Proto tato informace slouží jako drobná heuristika, ať se hodnota při počáteční inicializaci a při mutaci nastavuje ze smysluplného oboru hodnot.
4. Nepovinnou informaci o vazbě na jiný prvek/součástku.

První hodnota je nejvíc důležitá pro EA, jde o hodnotu se kterou přímo EA pracuje a upravuje. Zbylé prvky genu slouží pro samotnou simulaci obvodu nebo jako pomocná informace pro EA k zlepšení nalezení výsledku.

4.3 Fitness funkce

Fitness funkce porovnává signál referenčního obvodu se signálem obvodu kandidátního řešení, který byl získán simulací tohoto řešení. Jde tedy o porovnání podobnosti signálu, nicméně neporovnávají se celé délky výstupních signálu obou řešení, ale pouze se porovnává jistý výřez z obou signálů. Výřez odpovídá jisté časové délce. V ideálním případě fitness funkce nepočítá pouze s podobností s referenčním signálem, ale počítá s možností, že signál kandidátního řešení má lepší vlastnosti, než referenční signál. V této situaci se podobnost signálů může zhoršovat.

4.3.1 Fitness funkce s Pearsonovým korelačním koeficientem

Pearsonův korelační koeficient [19] určuje vztah/závislost mezi dvěma veličinami, tedy znázorňuje statistickou závislost dvou veličin. Pearsonův korelační koeficient udává míru korelace a nabývá hodnot $r \in (-1, 1)$. Přímou závislost veličin (tedy shodnost signálu) označuje $r = 1$. Pokud je $r = 0$ tak jsou veličiny zcela nezávislé, neexistuje mezi nimi žádný vztah. Zcela nepřímou závislost představuje $r = -1$. Pokud je hodnota $r \geq 0.85$, tak je řešení označeno za přijatelné. Vzorec pro výpočet koeficientu:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}, \text{ kde} \quad (4.1)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ a } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (4.2)$$

Výsledný vzorec použitý v implementaci dostaneme dosazením vzorců 4.2 do vzorce 4.1:

$$r = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n \sum x_i^2 - (\sum x_i)^2} \sqrt{n \sum y_i^2 - (\sum y_i)^2}} \quad (4.3)$$

Tato fitness funkce je snadná na implementaci. Mezi další výhody patří snadné vypořádání s nepřímou závislostí signálů, která by se v některých případech mohla hodit. Nicméně v této práci se tato vlastnost ignoruje, požadavkem je mít neinvertovaný výsledný signál. V závislosti na požadavky výsledného signálu může být velkou nevýhodou, že s touto fitness funkcí nelze získat lepší řešení, než je to referenční. Tzn., že nejlepší možné řešení je získání referenčního signálu.

Ve fitness funkci není koeficient počítán z celého signálu najednou, ale používá se tzv. plovoucí okno. Dojde k rozdělení signálu do oken o pevné délce padesáti hodnot, nebo zbytek hodnot konce signálu, které by celé okno nevyplnily. Pro každé okno se spočítá koeficient. Výsledný koeficient se spočítá jako průměr koeficientů oken.

Problémem fitness funkce je, že měří skutečně pouze závislost dvou veličin, tzn. že v tomto použití velmi dobře hlídá shodu frekvencí referenčního signálu a signálu kandidátního řešení, ale nehlídá shodu jejich amplitud. Z tohoto důvodu byl zaveden příznak, který říká, jestli kandidátní řešení je falešně pozitivní. Tato falešná pozitivnost se určuje na základě procentuálního porovnání maximální amplitudy obou signálů. Pokud maximální amplituda signálu kandidátního řešení nedosahuje alespoň 85% z maximální amplitudy referenčního signálu, tak je řešení označeno za falešně pozitivní. Nicméně i tato detekce nemusí vždy fungovat. Porovnávají se max. amplitudy z každého signálu, přičemž nezáleží, jestli jde o amplitudu kladnou či zápornou. Toto znamená, že se u obou signálů ignoruje kontrola kladné nebo záporné části (záleží na výběru maximální amplitudy) daného signálu, nebo

dojde porovnání kladné amplitudy jednoho signálu a záporné amplitudy druhého signálu. Navíc se neporovnávají všechny půl periody obou signálů, ale z každého signálu se vybere půl perioda, která má největší amplitudu, zbytek signálu se ignoruje. Toto vše může vést k špatné detekci.

Označení falešně pozitivní nijak neovlivňuje číselnou hodnotu fitness funkce, má tedy pouze informační charakter. Používá se například při výběru nejlepšího jedince nebo pro určení, zda je řešení přijatelné.

4.3.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem

Tato fitness funkce řeší problém nehlídání amplitudy u fitness funkce s Pearsonovým korelačním koeficientem. Hodnota fitness funkce se skládá ze tří složek. První složkou je výpočet Pearsonova korelačního koeficientu, který je shodný s výpočtem ve fitness funkci s Pearsonovým korelačním koeficientem, také se počítá podle vztahu 4.3 a využívá se princip plovoucího okna. Tato složka hlídá podobnost frekvence signálu kandidátního řešení a referenčního signálu. Nabývá hodnoty z intervalu $r \in (-1, 1)$, kde nejhorší hodnota je $r = -1$ a nejlepší hodnota je $r = 1$.

Další složkou je hodnota porovnávací podobnost maximální kladné amplitudy obou signálů. Základem je výpočet procentuální velikosti max. amplitudy signálu kandidátního řešení k max. amplitudě referenčního signálu, tedy $ratio = \frac{A_K}{A_R}$. Tato $ratio$ hodnota může být větší než 100%, pokud amplituda signálu kandidátního řešení je větší než amplituda referenčního signálu. Výsledná hodnota složky je rovna procentuální velikosti, tedy hodnotě $ratio$, pokud platí, že $ratio \leq 100\%$. Pokud je $ratio >> 100\%$, tak se výsledná hodnota spočítá jako $ratio = 100\% - overlap$, kde $overlap$ je spočten vztahem $ratio - 100\%$. Toto omezení je pro zachování principu Pearsonova korelačního koeficientu, tedy k porovnávání podobnosti, je tato procentuální velikost upravována. Výsledná $ratio$ hodnota nabývá hodnot z intervalu $ratio \in (0, 1)$, kde nejhorší hodnota je $r = 0$ a nejlepší hodnota je $r = 1$.

Poslední složkou je hodnota porovnávací podobnost maximální záporné amplitudy obou signálů. Výpočet je naprosto shodný s výpočtem druhé složky, tedy s hodnotou porovnávací maximální kladné amplitudy obou signálů. Jediný rozdíl je, že se zde porovnávají maximální záporné amplitudy.

Oproti fitness funkce s Pearsonovým korelačním koeficientem 4.3.1 a její detekci falešného pozitivního řešení je zde silnější kontrola podobnosti amplitudy. Je to dáno integrací této detekce do hodnoty fitness funkce. Dalším důvodem je separátní kontrola max. kladné amplitudy i max. záporné amplitudy. Nicméně ani tato metoda nemusí být plně spolehlivá, protože pro kladnou i zápornou část signálu se vybere jedna půl perioda, která má maximální amplitudu. A obě tyto půl periody se porovnávají s půl periodami referenčního signálu, které se vyberou stejným způsobem. Tímto se v kladné i záporné části ignoruje zbytek signálu, čímž může dojít k falešně dobré detekci.

Výsledná hodnota fitness funkce je rovna součtu všech tří složek, které jsou popsány výše. Hodnota fitness funkce nabývá hodnotu z intervalu $r \in (-1, 3)$, kde nejhorší hodnota je $r = -1$ a nejlepší hodnota je $r = 3$. Řešení je přijatelné, pokud je hodnota $r \geq 2.7$, aby hodnoty kontrolující amplitudy nepřebily hodnotu kontrolující frekvenci, tak je práh přijatelnosti nastaven tak vysoko.

4.3.3 Poziční fitness funkce

Jde o návrh fitness funkce, která má za cíl umožnit nalézt lepší signál, než je referenční signál. V konkrétním případě by měl umožnit nalézt větší amplitudu signálu kandidátního řešení, než je amplituda referenčního signálu. Je navržena pouze na obvody zesilovačů a oscilátorů, které generují harmonický sinusový signál.

Princip této fitness funkce je hlídání podobností frekvence signálů a jejich amplitud. Rozdílem oproti ostatním fitness funkcím je povolení dosáhnout větší amplitudy kandidátního řešení, než je amplituda referenčního signálu. Výpočty podobností se neprovádí na celém signálu najednou, ale signál se rozděluje na půl periody, na kterých výpočty probíhají. K tomuto je zapotřebí nalézt nulové body signálu, které vymezují půl periody. Mezi body, které vymezují půl periody, patří i začátek a konec signálu, i přesto, že se nenachází v nulovém bodě. Půl periody se postupně na kandidátním i referenčním řešení prochází a porovnávají se mezi sebou, kde se půl periody porovnávají podle pořadí. Tedy vezmou se první půl periody z obou signálů a porovnají mezi sebou, pak se druhé půl periody, třetí půl periody, atd., dokud se neprojdou všechny půl periody na jednom či druhém signálu. Pokud jeden signál má více půl period než ten druhý, tak se tyto půl periody, které jsou navíc, ignorují, protože by se neměly s čím porovnat. Samotná fitness funkce se skládá ze dvou částí a to mírou podobností frekvence signálů a mírou podobností amplitud signálů.

Míra podobností frekvence signálů hlídá, aby frekvence signálu kandidátního řešení byla přijatelná vzhledem k referenčnímu signálu. Vypočte se procentuální velikost šířky dané půl periody signálu k velikosti šířky půl periody referenčního signálu, tzv. $ratio_f$ hodnota. Tato hodnota se vypočte jako $ratio_f = \frac{f_K}{f_R}$, kde f_K je šířka půl periody u kandidátního řešení a f_R je šířka půl periody u referenčního řešení. Pokud platí podmínka $ratio_f > 100\%$, tak se tato hodnota upraví vztahem $ratio_f = 100\% - (ratio_f - 100\%)$. To proto, že se požaduje co nejpřesnější frekvence, proto se za přesah penalizuje. Výsledná hodnota míry podobnosti frekvence, tzv. $ratio_{fsum}$, je průměrem $ratio_f$ hodnot všech počítaných půl period a nabývá hodnot z intervalu $ratio_{fsum} \in (0, 1)$.

Další částí je míra podobnost amplitud, která hlídá minimální požadavky na výslednou amplitudu signálu kandidátního řešení, ale dovoluje dosáhnout i větších amplitud, než je tomu u referenčního signálu. Princip výpočtu je podobný jako u míry podobností frekvence, kdy je základem procentuální velikost amplitudy dané půl periody signálu k velikosti amplitudy půl periody referenčního signálu, tzv. $ratio_A$ hodnota. Výpočet je dán vztahem $ratio_A = \frac{A_K}{A_R}$, kde A_K je velikost amplitudy v případě kandidátního řešení a A_R je tedy velikost amplitudy u referenčního řešení. Na rozdíl od míry podobností frekvence zde není žádné omezení na přesah amplitud, takže $ratio_A$ hodnota se již nijak neupravuje a může platit $ratio_A > 100\%$. Podobně jako v předchozím případě je výsledná hodnota míry podobnosti amplitud, tzv. $ratio_{Asum}$, průměrem $ratio_A$ hodnot všech počítaných půl period, ale $ratio_A$ hodnota náleží intervalu $ratio_{Asum} \in (0, \infty)$.

Jako i předchozí fitness funkce, i zde není ohodnocení podobnosti amplitud vždy funkční. Zde se porovnávají amplitudy pro každou půl periodu a poté se tyto výsledky průměrují. Pokud některé půl periody budou mít velké ohodnocení amplitudy a zbylé půl periody budou mít velmi nízké ohodnocení amplitudy, tak se díky průměru tento rozdíl ztratí a hodnota míry podobnosti bude ukazovat na příznivý výsledek, i když tomu tak není.

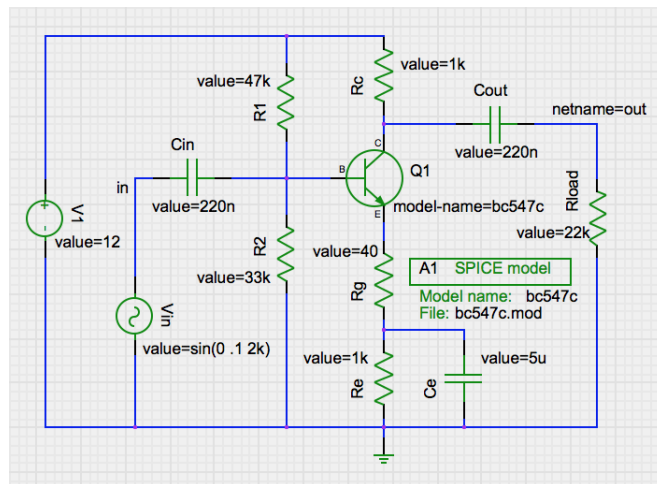
Kandidátní řešení je označeno za přijatelné, pokud platí podmínka $ratio_{fsum} \geq 0.7$ a současně $ratio_{Asum} \geq 0.8$. Výsledná hodnota fitness funkce, tzv. $fitness$ hodnota, se kterou pracuje EA, se počítá z těchto dvou částí. Prvně se nastaví $fitness$ hodnota jako $fitness = ratio_{fsum}$. Pokud pro $fitness$ hodnotu platí, že $fitness \geq 0.1$, tak se hodnota

dále upravuje, v opačném případě výpočet končí. Pokud v úpravě hodnoty platí podmínka $ratio_{f_{sum}} \geq 0.7$ nebo $ratio_{A_{sum}} \leq 1.0$, tak se nastaví *fitness* hodnota jako $fitness = fitness + ratio_{A_{sum}}$. V opačném případě, pokud podmínka není splněna, tak se *fitness* hodnota vypočte vztahem $fitness = fitness + 1.0$. Výsledná *fitness* hodnota je v rozmezí intervalu $fitness \in \langle 0, \infty \rangle$.

Kapitola 5

Experiment - jednostupňový zesilovač

V této kapitole budou popsány výsledky experimentů prováděných na obvodu jednostupňového zesilovače 2.1.3. Konkrétní zapojení, včetně nastavených hodnot součástek, které se používají jako referenční řešení, lze vidět na schématu 5.1.



Obrázek 5.1: Zapojení jednostupňového zesilovače včetně nastavených hodnot součástek obvodu.

Z tohoto zapojení podléhají evoluci jen součástky C_e , R_c , R_{load} , R_e , R_g , R_1 a R_2 . Tzn. že evoluce nastavuje hodnoty těchto prvků a hledá co nejlepší výsledek. Zbytek prvků obvodů, tedy C_{out} , C_{in} , V_{in} , V_1 a Q_1 , nepodléhají evoluci a tedy jejich hodnoty jsou stejné pro všechna kandidátní řešení. Porovnává se výstupní signál napětí na výstupu zesilovače, který je v schématu označen jako out.

Pro zrychlení nalezení řešení evoluce jsou hodnoty součástek, pouze těch jenž podléhají evoluci, omezeny intervalem hodnot, kde $C_e \epsilon < 1 \cdot 10^{-12}, 5 \cdot 10^9 > F$, $R_c \epsilon < 1, 5 \cdot 10^5 > \Omega$, $R_{load} \epsilon < 1, 5 \cdot 10^5 > \Omega$, $R_e \epsilon < 1, 2 \cdot 10^3 > \Omega$, $R_g \epsilon < 1, 5 \cdot 10^5 > \Omega$, $R_1 \epsilon < 1, 5 \cdot 10^5 > \Omega$, $R_2 \epsilon < 1, 5 \cdot 10^5 > \Omega$. Těmito intervaly je zabezpečen dostatečně velký prohledávací prostor, bez toho, aniž by evoluce dostala příliš velkou znalost jak nastavit obvod.

5.1 Testovací sestavy

Pro urychlení vyhodnocení práce byly jednotlivé experimenty prováděny na dvou různých testovacích sestavách:

1. PC sestava s čtyř-jádrovým procesorem Intel(R) Core(TM) i7-6700K CPU @ 4.0 GHz, 16 GB RAM a SSD diskem.
2. PC sestava s čtyř-jádrovým procesorem Intel(R) Core(TM) i7-4850HQ CPU @ 2.30GHz, 16 GB RAM a SSD diskem.

5.2 Genetický algoritmus

V této části jsou zapsány všechny experimenty genetického algoritmu na jednostupňovém zesilovači. Požadavkem na výsledný signál je mít co nejvěrnější frekvenci s porovnáním frekvence referenčního signálu. Dalším požadavkem je mít přijatelné zesílení výstupního signálu, ideálně ať je co největší, tzn. mít co největší amplitudu výstupního signálu.

Po provedení drobných testů se vyvodilo nastavení genetického algoritmu. Ve všech testech je velikost populace 10 jedinců. V GA se jedinci pro reprodukci vybírají pomocí turnajové selekce. Pro reprodukci jsou zde užity oba variační operátory, tedy křížení a mutaci. Pro křížení jedinců se používá jednobodové křížení s náhodným výběrem bodu. Mutace mění alely dvou genů, které jsou náhodně vybrané z chromozomu, náhodným výběrem z intervalu hodnot určených pro daný gen/součástku. K obnově populace se používají pouze noví potomci, tzn. že je použita varianta generativního GA s úplnou obnovou. Pro testy všech fitness funkcí je na nastaven počet generací genetického algoritmu na 7000.

5.2.1 Fitness funkce s Pearsonovým korelačním koeficientem

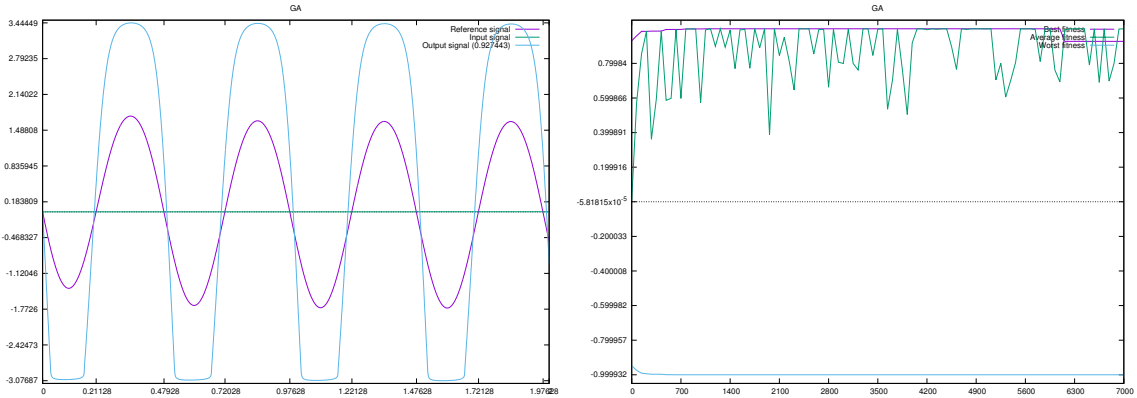
V této sekci jsou zaznamenány výsledky experimentů genetického algoritmu používající fitness funkce s Pearsonovým korelačním koeficientem. Tato fitness funkce má zvláštnost oproti jiným fitness funkcím, při použití GA vykazuje lepší výsledky, pokud je pravděpodobnost výskytu mutace větší než pravděpodobnost výskytu křížení. Proto je v této variantě nastavená pravděpodobnost výskytu mutace na 80%, kdežto pravděpodobnost pro křížení pouze 20%. V tabulce 5.2 lze vidět statistické údaje průběhu experimentu testovaných na testovací soustavě č. 1.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
70	17	6	24.28436	8.57086	43

Obrázek 5.2: Statistické údaje experimentů kombinace: jednostupňový zesilovač, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Ze statických experimentů lze vyčíst, úspěšnost nálezu řešení je 24%. Došlo i k nalezení falešně pozitivních řešení, výskyt těchto řešení činí 8.5% z experimentu. Na obrázku 5.3a je ukázáno správné výsledné řešení, při kterém se vykazuje i velké zesílení výsledného signálu oproti referenčního signálu. Záporné půl periody mají na vrcholu trochu zkosený průběh, kdežto kladné půl periody ho mají zaoblený. Avšak toto nevádí z pohledu požadavku na výstup. Na obr. 5.3b je zachycen průběh hodnoty fitness funkce v evoluci daného testu. Na

tomto průběhu lze vidět zvláštnost, hodnota fitness funkce nejlepšího řešení někde kolem 6000 generací poklesne. Je to způsobeno tím, že do generace 6000 byl nejlepší jedinec falešně pozitivní. Poté se našel jedinec, který nebyl falešně pozitivní a nahradil dosavadního nejlepšího jedince, i přesto, že jeho hodnota fitness funkce je vyšší.

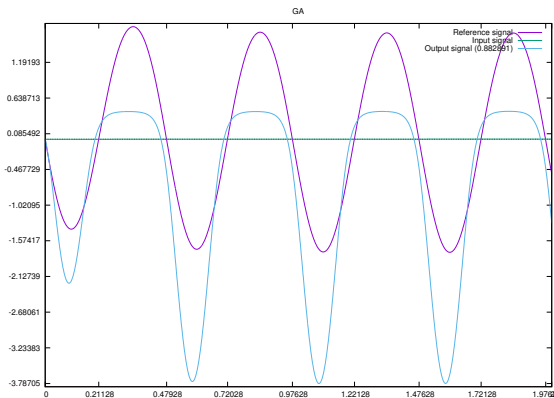


(a) Výstupní signál kandidátního řešení.

(b) Průběh hodnoty fitness funkce.

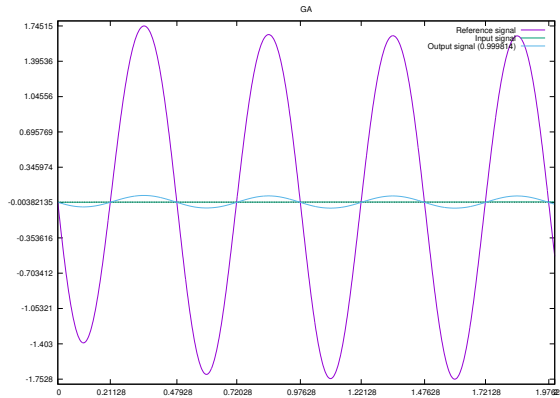
Obrázek 5.3: Ukázka úspěšného řešení v kombinaci: jednostupňový zesilovač, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Příklad falešně pozitivního řešení je vyobrazeno na obr. 5.4, kde lze vidět, že zesílení záporné části signálu je dobré, nicméně amplituda kladné části signálu je velmi malá a nevyhovuje z pohledu požadavků na výstupní signál obvodu.



Obrázek 5.4: Ukázka falešně dobrého řešení v kombinaci: jednostupňový zesilovač, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Mezi špatná řešení lze zařadit řešení, jehož výsledný signál je ukázán na obr. 5.5. Výstupní signál má velmi dobrou podobnost frekvence s referenčním signálem, ale jeho amplituda je příliš malá, tudíž na zesilovači nedochází k velkému zesílení vstupního signálu.



Obrázek 5.5: Ukázka falešně špatného řešení v kombinaci: jednostupňový zesilovač, GA, fitness funkce s Pearsonovým korelačním koeficientem.

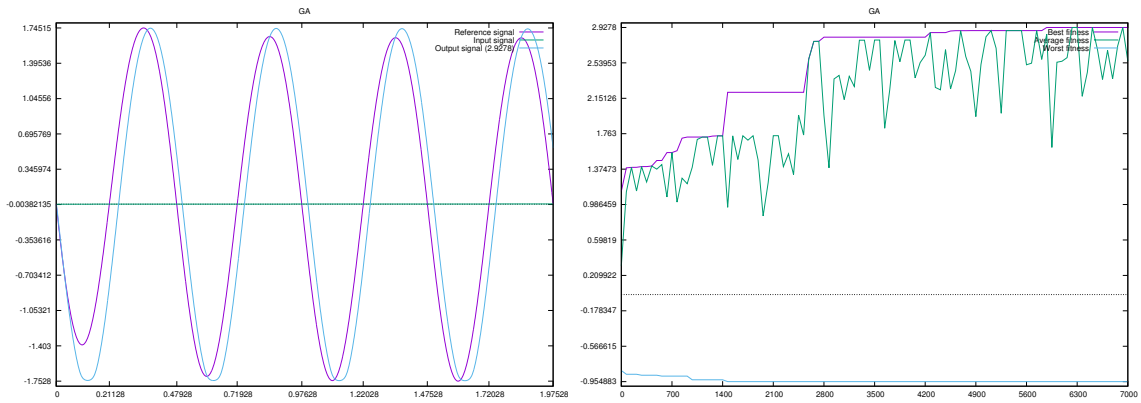
5.2.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem

Zde jsou popsány výsledky experimentů genetického algoritmu s použitím vylepšené fitness funkce s Pearsonovým korelačním koeficientem. Oproti fitness funkce s Pearsonovým korelačním koeficientem je zachováno standardní nastavení pravděpodobností variační operátorů genetického algoritmu. V případě křížení, jakožto hlavního variačního operátoru, je pravděpodobnost výskytu 80%. Mutace je méně významným variačním operátorem a její pravděpodobnost výskytu činí 20%. Experiment probíhal na testovací soustavě č. 2 a tabulka 5.6 prozrazuje statistické výsledky experimentu.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
70	56	0	79.99878	0.0	81

Obrázek 5.6: Statistické údaje experimentů kombinace: jednostupňový zesilovač, GA, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Z výsledků je vidět, že vylepšená fitness funkce v kombinaci s genetickým algoritmem je velmi úspěšná při hledání řešení pro jednostupňový zesilovač. Tato úspěšnost je téměř 80%, přitom se neukázaly žádná falešná řešení. Toto plyne z faktu, že fitness funkce hlídá maximální kladnou i zápornou amplitudu, proto nedovolí velký rozdíl mezi velikostí kladné a záporné amplitudy. Na obrázku 5.7a lze vidět příklad dobrého řešení tohoto experimentu, kde řešení má hodnotu fitness funkce 2.92.



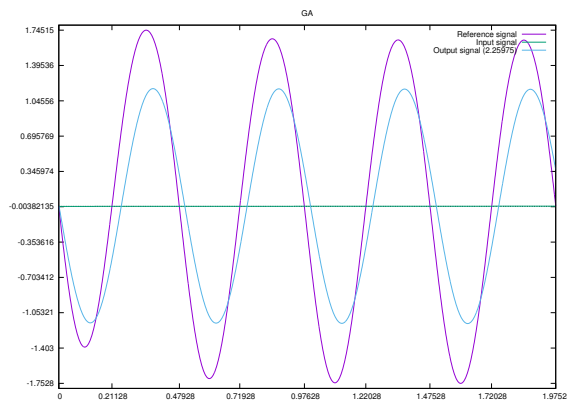
(a) Výstupní signál kandidátního řešení.

(b) Průběh hodnoty fitness funkce.

Obrázek 5.7: Ukázka úspěšného řešení v kombinaci: jednostupňový zesilovač, GA, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Z grafu 5.7a je viditelné, že výsledný signál je zanedbatelně posunutý oproti referenčnímu signálu. Toto ale výsledku nevádí. Dále lze z grafu zjistit, že výsledný signál má nepatrně větší amplitudu než referenční signál. Větší amplituda je z pohledu fitness funkce špatná a kandidátní řešení je za to penalizováno, nicméně z pohledu požadavků na výstup je větší amplituda žádaná. Průběh hodnoty fitness funkce evoluce lze vidět na obr. 5.7b.

Naopak lze i ukázat příklad nejlepšího řešení evoluce, které ale nebylo úspěšné. Tento příklad lze vidět na obr. 5.8. Z ukázky je viditelné, že signál je trochu posunutý a má menší amplitudu. Nicméně signál není špatný a pokud by byly na obvod kladeny jiné požadavky, tak by mohl být tento výsledek dostačující.



Obrázek 5.8: Ukázka špatného řešení v kombinaci: jednostupňový zesilovač, GA, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

5.2.3 Poziční fitness funkce

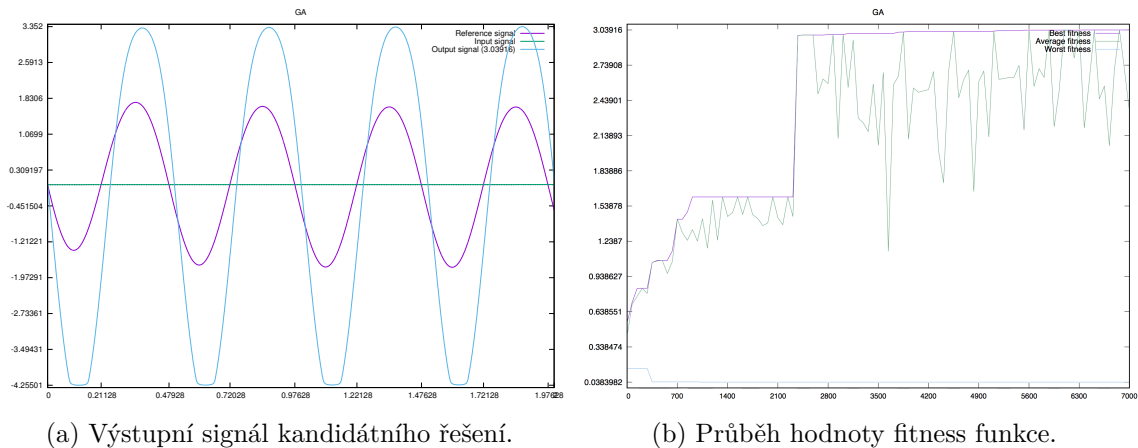
V této části jsou uvedeny výsledky experimentů genetického algoritmu s použitím poziční fitness funkce. Stejně jako u vylepšené fitness funkce s Pearsonovým korelačním koeficientem je zde zachováno standardní nastavení pravděpodobností variační operátorů genetického algoritmu a taktéž je pravděpodobnost výskytu křížení 80% a pro mutaci je pravděpodobnost

výskytu 20%. Experiment byl testován na testovací soustavě č. 1 a tabulka 5.9 jsou zachyceny statistické výsledky experimentu.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
70	34	8	48.57025	11.4273	62

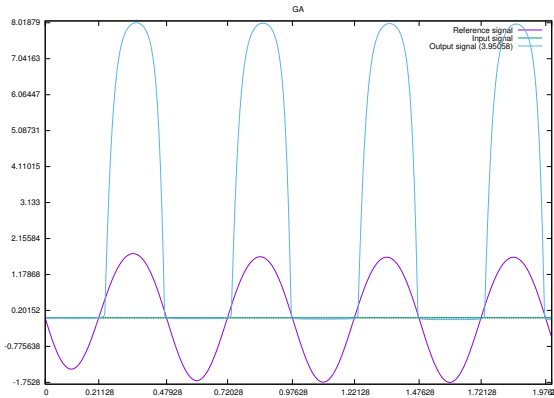
Obrázek 5.9: Statistické údaje experimentů kombinace: jednostupňový zesilovač, GA, poziční fitness funkce.

Výsledky vykazují také vysokou úspěšnost, která je kolem 48%. Podobně jako je tomu u vylepšené fitness funkce s Pearsonovým korelačním koeficientem, tak je zde přísná kontrola amplitud a frekvence. Navíc se zde kontroluje velikost amplitudy a šířka každé půl periody. Na obrázku 5.10a lze vidět úspěšné řešení tohoto experimentu. Opět lze pozorovat drobný posun signálu. Ale získalo se řešení s větší amplitudou, než je amplituda referenčního řešení. Tímto se našlo lepší řešení, než je referenční. Průběh hodnoty fitness funkce je zobrazen na obrázku 5.10b, kde lze vidět výrazné zlepšení před 2800 generací.



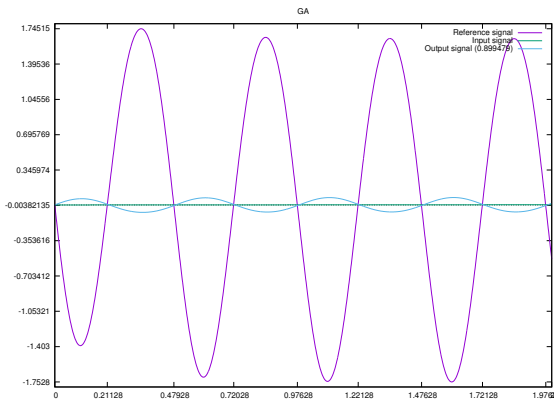
Obrázek 5.10: Ukázka úspěšného řešení v kombinaci: jednostupňový zesilovač, GA, poziční fitness funkce.

Nicméně výsledky experimentu ukázaly kolem 11.5% falešně úspěšných řešení. Jedno z těchto řešení je vyobrazeno na obr. 5.11. K této situaci nastalo, protože podobnost frekvencí je v pořádku, je zde pouze drobná odchylka. Problém nastává v porovnání amplitud. Jak bylo popsáno v sekci 4.3.3, tak výpočet amplitudy se počítá pro každou půl periodu zvlášť a pak se výsledky průměrují do jedné hodnoty. Výpočty pro kladné půl periody vykazují vysoké hodnoty, protože amplitudy těchto půl period jsou mnohem větší, než je tomu u referenčního signálu. Kdežto výsledky pro záporné půl periody jsou velmi nízké, protože hodnoty amplitud těchto půl period se pohybují kolem 0. Vzhledem k tomu, že konečný výsledek je průměr amplitud, tak se tento rozdíl ztratí.



Obrázek 5.11: Ukázka falešně dobrého řešení v kombinaci: jednostupňový zesilovač, GA, poziční fitness funkce.

Mimo falešně pozitivních výsledků zde našel genetický algoritmus řešení, která nevyhovují požadavkům na výstup. Jedno takové řešení je vyobrazeno na obr. 5.12. Zde je signál kandidátního řešení dokonce invertován vůči referenčnímu signálu, což je nežádoucí vlastnost.



Obrázek 5.12: Ukázka špatného řešení v kombinaci: jednostupňový zesilovač, GA, poziční fitness funkce.

5.3 Evoluční strategie

Zde jsou popsány všechny experimenty evoluční strategie na jednostupňovém zesilovači. Požadavky na výstupní signál výsledného obvodu jsou stejné jako v případě genetického algoritmu. Tedy mít co nejpodobnější frekvenci s referenčním signálem a současně mít alespoň přijatelnou amplitudu signálu vzhledem k referenčnímu signálu, ideálně mít amplitudu větší než je tomu u referenčního řešení.

Tak jako u genetického algoritmu je ve všech testech velikost populace 10 jedinců. Na rozdíl od GA se zde nepoužívá variační operátor křížení, používá se pouze mutace jako hlavní variační operátor. Tímto odpadá výběr rodičů, protože k mutaci je vybrán každý jedinec z populace. Mutace používá Gaussovou distribuci pravděpodobnosti a také využívá autoevoluce, kde každý gen má vlastní mutační krok. Počáteční mutační krok se spočítá

jako $\sigma = (max - min) * 0.07$, kde max představuje maximální hodnotu, kterou gen/součástka může nabývat. Naopak min je minimální hodnota, která genu může být nastavena. Vzhledem k tomu, že se používá jen mutace a to vždy, tak navíc odpadá jakékoliv nastavování pravděpodobností výskytu variačních operátorů. Stejně jako u GA je i zde je nastaveno 7000 generací pro evoluční strategie u všech testů. Ve všech případech fitness funkce je použité stejné nastavení evolučního algoritmu.

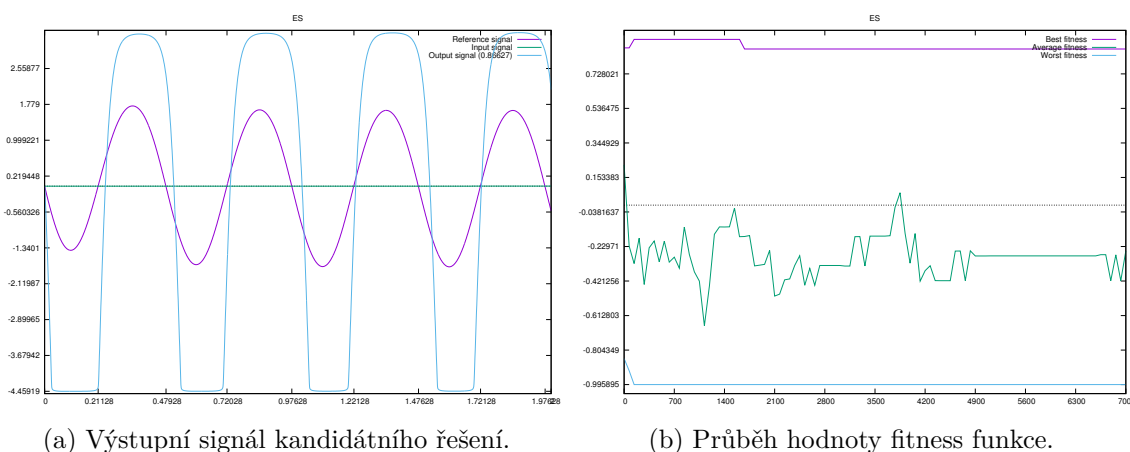
5.3.1 Fitness funkce s Pearsonovým korelačním koeficientem

Výsledky experimentů evoluční strategie používající fitness funkce s Pearsonovým korelačním koeficientem jsou zaznamenány v této části dokumentu. Testy probíhaly na testovací soustavě 2 a jejich statistické výsledky jsou zaznamenány v tabulce 5.13.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
70	8	32	11.4273	45.7138	137

Obrázek 5.13: Statistické údaje experimentů kombinace: jednostupňový zesilovač, GA, fitness funkce s Pearsonovým korelačním koeficientem.

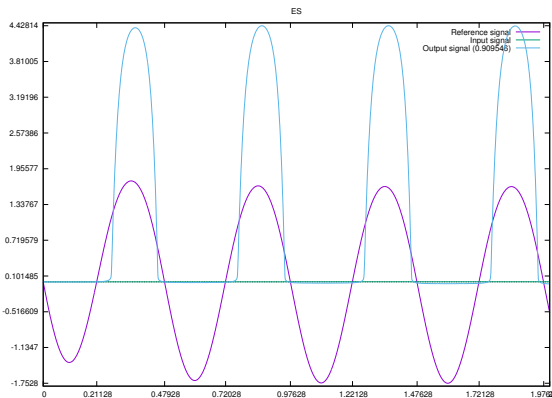
Testy ukázaly, že se v tomto experimentu našlo 40 řešení vhodných ze 70, ale 32 z těchto řešení je falešně pozitivních. Procentuálně je 45% ze všech výsledků falešně pozitivních, takže skutečná úspěšnost ES s touto fitness funkcí je kolem 11%. Toto poukazuje na nízkou účinnost ES s testovanou fitness funkcí a je potřeba si dávat pozor na falešně pozitivní výsledky. Příklad úspěšného řešení je zobrazen na obr. 5.14a, s průběhem hodnoty fitness funkce v evoluční strategii na obrázku 5.14b. Výsledný signál má mnohem větší amplitudu než signál referenční, tedy v zesilovači dochází k většímu zesílení vstupního signálu, než je tomu u referenčního zesilovače. Podle průběhu hodnoty fitness funkce bylo řešení nalezeno před dosažením počtu generací 2100, tudíž jde o velmi brzký nález řešení.



Obrázek 5.14: Ukázka úspěšného řešení v kombinaci: jednostupňový zesilovač, ES, fitness funkce s Pearsonovým korelačním koeficientem.

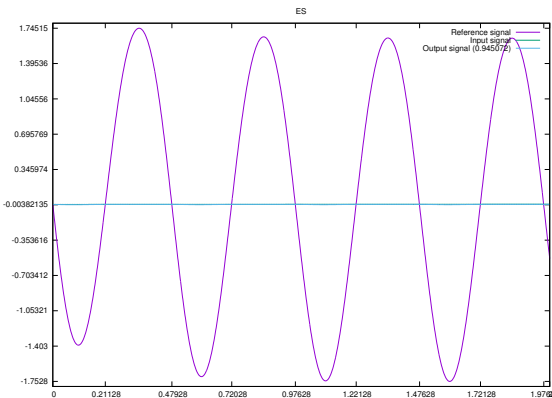
Vzhledem k velkému počtu falešných řešení je na obr. 5.15 ukázán příklad takového řešení. Podobný případu už bylo dosaženo genetickým algoritmem s poziční fitness funkcí.

Jde o případ s dobrou frekvencí a dobrou kladnou polovinou signálu. Nicméně amplituda záporné části signálu je téměř nulová.



Obrázek 5.15: Ukázka falešně dobrého řešení v kombinaci: jednostupňový zesilovač, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Příklad špatného řešení je ukázán na obr. 5.16. Výstupní signál je téměř nulový, tedy v zesilovači výsledného řešení téměř vůbec nedochází k zesílení.



Obrázek 5.16: Ukázka falešně špatného řešení v kombinaci: jednostupňový zesilovač, ES, fitness funkce s Pearsonovým korelačním koeficientem.

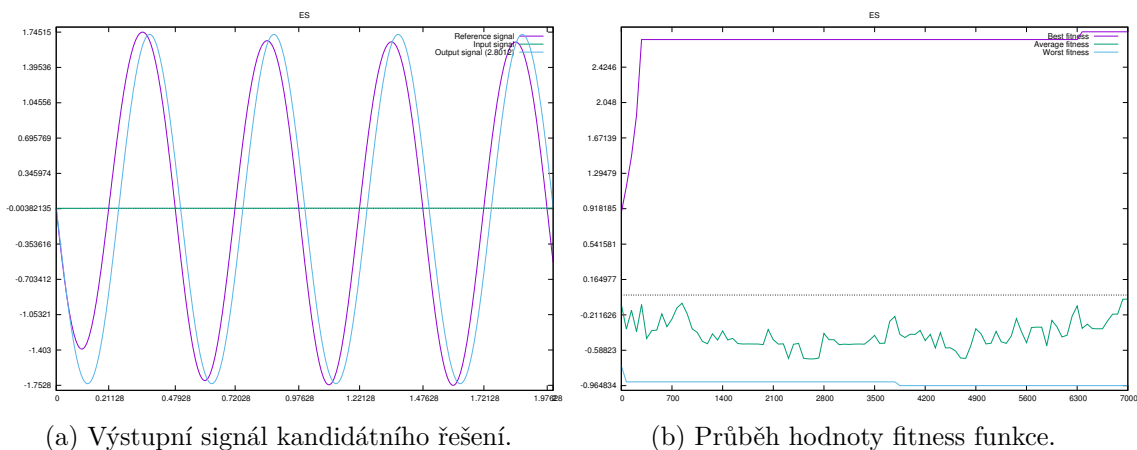
5.3.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem

Zde se nachází popis výsledků experimentů evoluční za použití vylepšené fitness funkce s Pearsonovým korelačním koeficientem. Experiment probíhal na testovací soustavě č. 2, tabulka 5.17 ukazuje statistické výsledky experimentu.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
70	19	0	27.14233	0.0	92

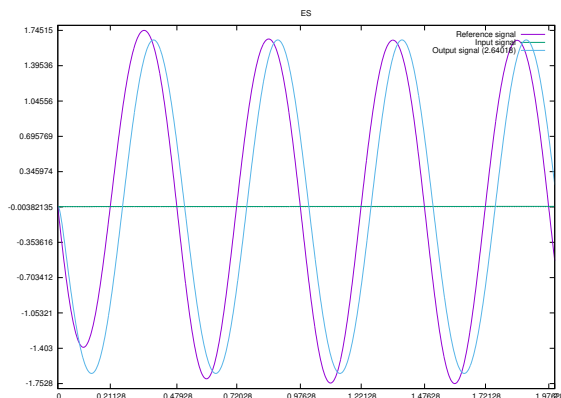
Obrázek 5.17: Statistické údaje experimentů kombinace: jednostupňový zesilovač, ES, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Podle výsledků je úspěšnost kolem 27%, což nejde o velmi dobrý výsledek. Nicméně ani zde fitness funkci nevykazuje žádné falešné úspěšné řešení. Ukázkou správného řešení lze vidět na obrázku 5.18a. Z grafu vyplývá, že výsledný signál je posunutý oproti referenčnímu signálu, ale v přijatelné míře vzhledem k požadavkům na výsledné řešení. Výsledná amplituda signálu je téměř shodná s amplitudou referenčního signálu, i když je o něco větší. Opět platí, že z pohledu fitness funkce jde o horší výsledek amplitudy, ale z pohledu požadavků na řešení je o dobrý výsledek. Průběh hodnoty fitness funkce evoluce u tohoto lze vidět na obr. 5.18b.



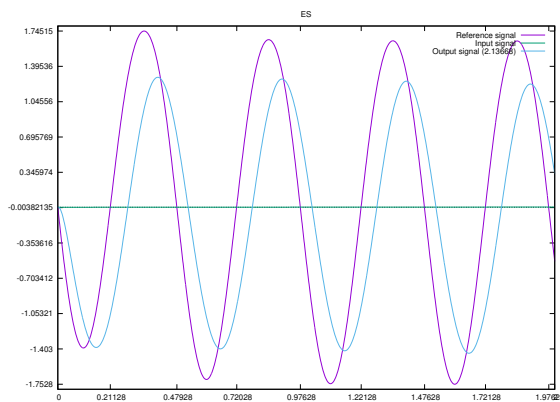
Obrázek 5.18: Ukázka úspěšného řešení v kombinaci: jednostupňový zesilovač, ES, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Ze statistiky není čitelné, že se našlo mnoho řešení, která se blížila k hranici přijatelnosti. I přes to, že kandidátní řešení tuto hranici nedosáhla, tak šlo o kvalitní výsledky, jak je možno vidět na obrázku 5.19. Toto řešení by se mohlo v pořádku použít pro nastavení výsledného obvodu. I toto ukazuje přísné nastavené požadavky na hodnotu fitness funkce, aby byla zajištěná kvalitní frekvence a amplituda výstupního signálu.



Obrázek 5.19: Ukázka nesprávného řešení z pohledu fitness funkce, ale dobrého řešení z pohledu požadavků na výstup. Řešení v kombinaci: jednostupňový zesilovač, ES, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Evoluční strategie společně s touto fitness funkcí našla i špatná řešení jako je na obr. 5.20. Toto řešení má už relativně hodně posunutý signál, navíc má menší amplitudu na hranici přijatelnosti. Oproti jiným nalezeným kandidátním řešením není tento výstup dobrý.



Obrázek 5.20: Ukázka špatného řešení v kombinaci: jednostupňový zesilovač, ES, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

5.3.3 Poziční fitness funkce

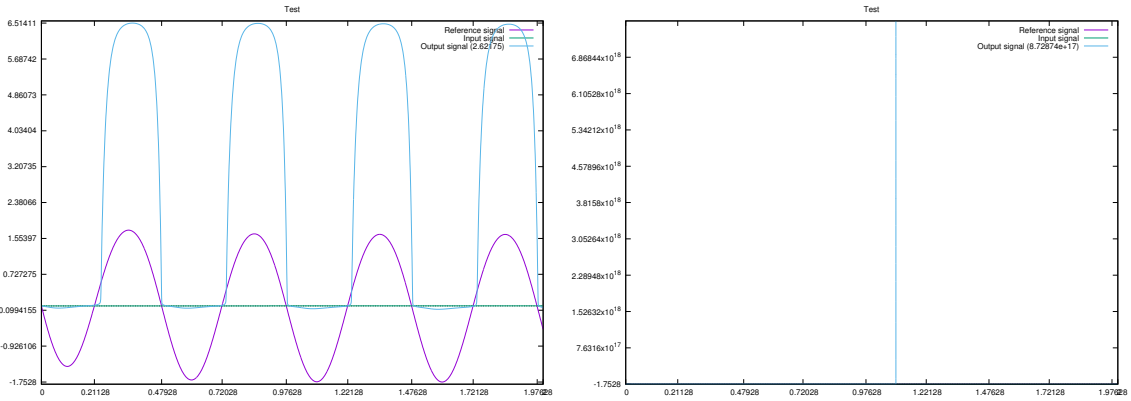
Výsledky experimentů evoluční strategie s použitím poziční fitness funkce jsou uvedené v této sekci. Experiment byl testován na testovací soustavě č. 1 a statické výsledky experimentu jsou zachyceny v tabulce 5.21.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
70	0	61	0.0	87.14142	62

Obrázek 5.21: Statistické údaje experimentů kombinace: jednostupňový zesilovač, ES, poziční fitness funkce.

Tato fitness funkce má 0% úspěšnost, jak je zapsáno ve statických výsledcích. Důvodem je velký počet falešně pozitivních výsledků, jenž se pohybuje kolem 87% ze všech testů. Na ukázce falešně pozitivního řešení 5.22b lze trochu vidět, co je toho příčinou. I když to z grafu není vidět, na základě testování se zjistilo, že podobnost frekvence řešení s referenčním řešením je přijatelná, ale došlo k porušení amplitudové podmínky, kdy je amplituda nepřiměřeně velká. Tento jev je způsoben neomezením maximální amplitudy touto fitness funkcí. Vzhledem k tomu, že frekvence je korektní, tak fitness funkce přijme tuto amplitudu a řešení získá vysokou fitness hodnotu.

Na obrázku 5.22a jde o případ, kdy kvůli výpočtu ohodnocení amplitud, který se počítá jako průměr amplitud všech půl period, došlo k ignorování velikosti amplitud záporných půl period.



(a) Výstupní signál kandidátního řešení.

(b) Výstupní signál kandidátního řešení.

Obrázek 5.22: Ukázka falešně pozitivních řešení v kombinaci: jednostupňový zesilovač, ES, poziční fitness funkce.

5.3.4 Shrnutí a diskuze

V této kapitole jsou popsány experimenty s jednostupňovým zesilovačem, kde se vyzkoušely 2 typy evoluce a 3 typy fitness funkcí, které se testovali pro každý typ evoluce. Výsledky se můžou porovnávat různým způsobem. Prvním možným způsobem je porovnání fitness funkcí v rámci daného typu evoluce.

Pro genetický algoritmus se testovaly 3 druhy fitness funkcí a to: fitness funkce s Pearsonovým korelačním koeficientem, vylepšená fitness funkce s Pearsonovým korelačním koeficientem a poziční fitness funkce. Pro fitness funkci s Pearsonovým korelačním koeficientem z výsledku plyne menší úspěšnost než u ostatních fitness funkcí, pohybuje se kolem 24%. Ale tato fitness funkce velmi dobře kontroluje frekvenci výstupního signálu a dovoluje nalézt i zajímavá řešení ve smyslu zesílení signálu, čili velikosti amplitudy signálu oproti signálu referenčnímu. Tato fitness funkce je vhodná pro použití, kde je nutné mít co nejlepší frekvenci, je snaha dosáhnout lepšího řešení, než je referenční řešení, ale nevadí, že se řešení nenajde, případně, že se řešení hledá vícekrát. Poté tu je vylepšená fitness funkce s Pearsonovým korelačním koeficientem, která hlídá frekvenci téměř tak dobře jako fitness funkce s Pearsonovým korelačním koeficientem. Navíc na rozdíl od ní kontroluje amplitudu. Fitness funkce měla vysokou úspěšnost, která se pohybovala kolem 80%. Velkou výhodou je, že fitness funkce v testech nevykázala žádné falešně pozitivní výsledky. Nevýhoda této fitness funkce je, že nedovoluje získat lepší řešení než je referenční řešení, resp. umožňuje získat zanedbatelné zlepšení. Z toho plyne, že tato fitness funkce se hodí tam, kde je nutné najít výsledek s největší pravděpodobností, kde nevadí, případně kde je žádané, že se nenajde lepší než referenční řešení. Navíc se hodí k případům, kdy nedochází ke kontrole osobou, protože fitness funkce nevykazuje falešně pozitivní řešení. Poslední fitness funkce je poziční fitness funkce, která nehlídá frekvenci signálu tak dobře jako předchozí dvě fitness funkce. Odlišnost této fitness funkce je v tom, že nehlídá maximální amplitudu a tedy více umožňuje najít řešení, které mají větší zesílení než referenční signály. Ukázalo se, že v tomto případě má fitness funkce úspěšnost kolem 48%, což není špatný výsledek. Fitness funkce je tedy vhodná pro případy, kde je žádané najít signál s lepším zesílením.

V evoluční strategii se porovnávaly stejné fitness funkce. Všechny fitness funkce mají stejné vlastnosti, jak tomu je v případě genetického algoritmu, viz. výše. Fitness funkce s Pearsonovým korelačním koeficientem vykazovala vysoký počet falešně pozitivních řešení

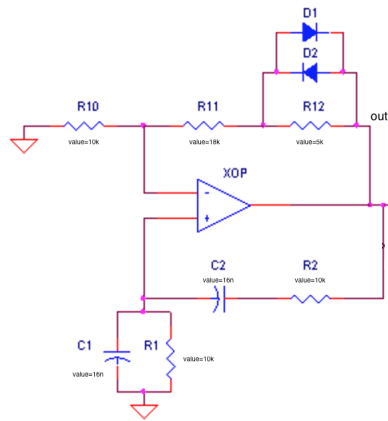
a tak má v těchto testech úspěšnost kolem 11% a řadí se tak jako druhá nejspolehlivější funkce. V případě použití evoluční strategie je vhodná k případům, kde je velmi žádané dosáhnout lepšího zesílení výsledného obvodu než je zesílení referenčního obvodu. Nicméně nesmí vadit malá úspěšnost nálezů řešení a je nutná kontrola falešně pozitivních řešení. Vylepšená fitness funkce s Pearsonovým korelačním koeficientem opět měla nejvyšší úspěšnost, která se ale pohybuje pouze kolem 27%. Opět nevykázala žádné falešné řešení. Proto je vhodná tam, kde je potřeba najít výsledek s co největší pravděpodobností i přesto, že se nenajde lepší řešení než to referenční. Opět je vhodná tam, kde není k dispozici kontrola falešně pozitivních řešení. Poziční fitness funkce měla 0% úspěšnost. Nekontrolování maximální amplitudy na rozdíl od předchozího případu se zde ukazuje jako nevýhoda. Řešení detekuje korektní podobnost frekvence s referenční frekvencí, ale nekontroluje max. hodnotu amplitudy, a proto toto řešení je přijato jako správné. Stává se to při porušení amplitudové podmínky. Tato fitness funkce ve spojení evoluční strategií není pro tento experiment použitelná.

Z porovnání testů mezi genetickým algoritmem a evoluční strategií vzešel zajímavý výsledek. Evoluční strategie je určena pro práci s reálnými čísly, ale ve všech testovaných fitness funkcích byl genetický algoritmus úspěšnější. Z toho plyne, že s každou z uvedených fitness funkcí je lepší použít genetický algoritmus při evoluční optimalizaci jednostupňového zesilovače.

Kapitola 6

Experiment - oscilátor s Wienovým článkem

V této kapitole budou popsány výsledky experimentů prováděných na obvodu oscilátoru s Wienovým článkem 2.2.2. Konkrétní zapojení, včetně nastavených hodnot součástek, které se používají jako referenční řešení, lze vidět na schématu 6.1.



Obrázek 6.1: Zapojení oscilátoru s Wienovým článkem, včetně nastavených hodnot součástek obvodu [3].

Jako v případě genetického algoritmu, i zde podléhají evoluci jen určité součástky obvodu a jsou to C_1 , C_2 , R_1 , R_2 , R_{10} , R_{11} a R_{12} . Tedy pouze těmto prvkům obvodu evoluce nastavuje hodnoty a snaží se mezi nimi najít řešení. Další součástky, jako XOP , D_1 a D_2 , nepodléhají evoluci a jejich hodnoty jsou neměnné pro všechna řešení. Porovnává se výstupní signál napětí oscilátoru, který je v schématu zaznačen jako out. V tomto zapojení je jedná zvláštnost, hodnoty C_1 a C_2 by se měly rovnat. To samé platí pro R_1 , R_2 . Kvůli tohoto byla v evolučním algoritmu implementována vazba součástek, která zajistí, aby hodnoty provázaných součástek byly stejné.

Stejně jako u GA i zde jsou pro urychlení nalezení řešení evoluce hodnoty součástek, omezeny intervalem hodnot, kde $C_1 \epsilon < 1 \cdot 10^{-12}, 1 \cdot 10^{-6} > F$, $C_2 \epsilon < 1 \cdot 10^{-12}, 1 \cdot 10^{-6} > F$, $R_1 \epsilon < 1,5 \cdot 10^5 > \Omega$, $R_2 \epsilon < 1,5 \cdot 10^5 > \Omega$, $R_{10} \epsilon < 1,5 \cdot 10^5 > \Omega$, $R_{11} \epsilon < 1,5 \cdot 10^5 > \Omega$, $R_{12} \epsilon < 1,5 \cdot 10^5 > \Omega$. Tímto nastavením je prohledáván dostatečně velký prohledávací prostor, aniž by evoluce měla příliš velkou znalost jak nastavit obvod. Kvůli kondenzátor

C_e u jednostupňového zesilovače by se mohlo zdát, že v tomto experimentu se prohledává menší prostor pro nalezení řešení. Ale není tomu tak, protože kondenzátor C_e nemá velký vliv na výsledek obvodu, tudíž jeho nastavení nemá velkou důležitost. Kdežto u oscilátoru s Wienovým článkem jsou kondenzátory C_1 a C_2 velmi důležité součástky obvodu a silně ovlivňují výstup oscilátoru.

Testování experimentu oscilátoru s Wienovým článkem probíhá na stejné testovací sestavě 5.1, jako tomu bylo i u genetického algoritmu. Počet spuštěných testů pro každou variantu se ve srovnání s GA zmenšil na 40. Je to kvůli větší náročnosti simulace obvodu a tudíž i větší časové náročnosti na vyhledání výsledku.

6.1 Genetický algoritmus

V této části jsou uvedeny experimenty genetického algoritmu na oscilátoru s Wienovým článkem. Cílem je dosažení výsledku, kdy kmity signálu kandidátního řešení byly co nejvěrnější kmitům referenčního signálu. Tzn. mít co nejpodobnější frekvenci signálu a také i amplitudu. Přijatelné řešení tedy je, pokud frekvence a amplitudy splňují limity odchylek od referenčního řešení. Přijatelné také je, pokud kandidátní řešení bude mít větší oscilace signálu, než referenční řešení, tedy jeho amplitudy budou větší.

Pro všechny testy genetického algoritmu je nastavena velikost populace na 10 jedinců. Zbylé nastavení GA je podobné jako v případě testování GA na jednostupňovém zesilovači. Tedy jedinci pro reprodukci se volí pomocí turnajové selekce. Reprodukce jedinců používá oba variační operátory, tedy křížení a mutaci. Křížení jedinců se vykonává pomocí jednobodové křížení s náhodným výběrem bodu. Mutace opět mění alely dvou genů, které jsou vybrané náhodným způsobem z chromozomu. Hodnoty se změni náhodným výběrem z rozsahu hodnot určených pro daný gen/součástku. Používá se přístup generativního GA s úplnou obnovou pro obnovu populace, tedy do nové populace jsou vybráni pouze noví jedinci, tzv. potomci. Počet generací evoluce pro všechny testy GA je roven 10000.

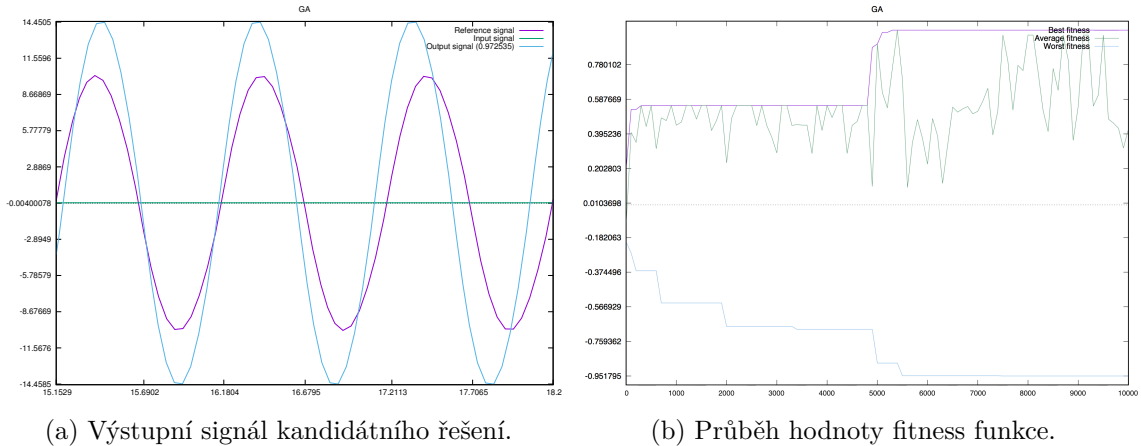
6.1.1 Fitness funkce s Pearsonovým korelačním koeficientem

V této části jsou uvedeny výsledky experimentů genetického algoritmu používající fitness funkce s Pearsonovým korelačním koeficientem. Tak jako v GA u jednostupňového zesilovače, tak i zde platí zvláštnost této fitness funkce, kdy se dosahuje lepší výsledky při použití větší pravděpodobnosti výskytu mutace než je pravděpodobnost výskytu křížení. Nastavení je opět 80% pravděpodobnosti výskytu mutace, 20% pravděpodobnosti pro křížení. V tabulce 6.2 jsou uvedeny statistické údaje experimentu, který se vykonával na testovací soustavě č. 1.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
40	5	9	12.5	22.49908	66

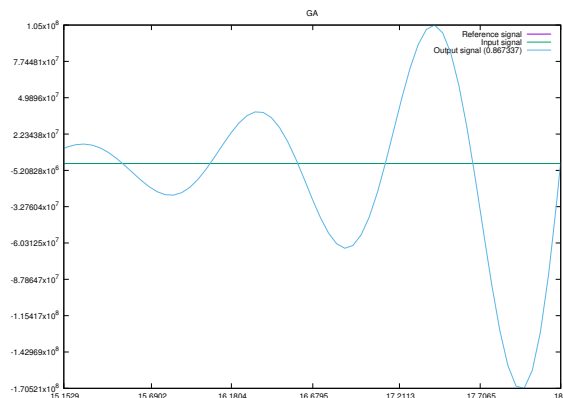
Obrázek 6.2: Statistické údaje experimentů kombinace: oscilátor s Wienovým článkem, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Ze statických experimentů lze vyčíst, že reálná úspěšnost nálezů řešení je 12.5%. Je to způsobeno faktem, že 9 ze 14 nálezů bylo falešně pozitivních, tedy kolem 22% z všech testů bylo falešně pozitivních. Tato metoda má tedy tendenci hledat falešná řešení, pokud nějaké najde. Na obrázku 6.3a lze vidět jedno z úspěšných řešení. Toto řešení má o něco menší šířku půl periody, ale je to přijatelné. Navíc má trochu větší amplitudu, které je stabilní. Na obrázku 6.3b je možnost vidět průběh fitness funkce během evoluce.



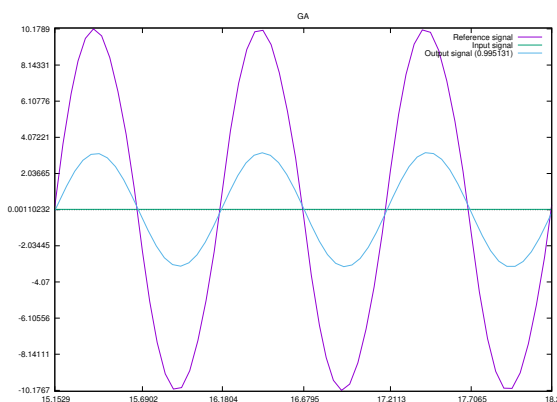
Obrázek 6.3: Ukázka úspěšného řešení v kombinaci: oscilátor s Wienovým článkem, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Jedno z falešně pozitivních řešení je vyobrazeno na obr. 6.4, kde je vidět příliš velkých amplitud jednotlivých půl period. Dokonce referenční řešení je tak malé, že není na grafu ani vidět. Navíc jde vidět, že signál nemá stabilní amplitudu, amplituda představující napětí stále roste. Toto vše může naznačovat nedodržení amplitudové podmínky oscilátoru, tedy jde o chybné řešení.



Obrázek 6.4: Ukázka falešně dobrého řešení v kombinaci: oscilátor s Wienovým článkem, GA, fitness funkce s Pearsonovým korelačním koeficientem.

Jako příklad špatného kandidátního řešení je zobrazeno obr. 6.5. Frekvence výstupního signálu je téměř perfektní, ale výsledné napětí na výstupu tohoto řešení je příliš malé a proto je nevhodné. Ovšem při jiných požadavcích na výstup by mohlo být toto řešení výborné.



Obrázek 6.5: Ukázka falešně špatného řešení v kombinaci: oscilátor s Wienovým článkem, GA, fitness funkce s Pearsonovým korelačním koeficientem.

6.1.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem

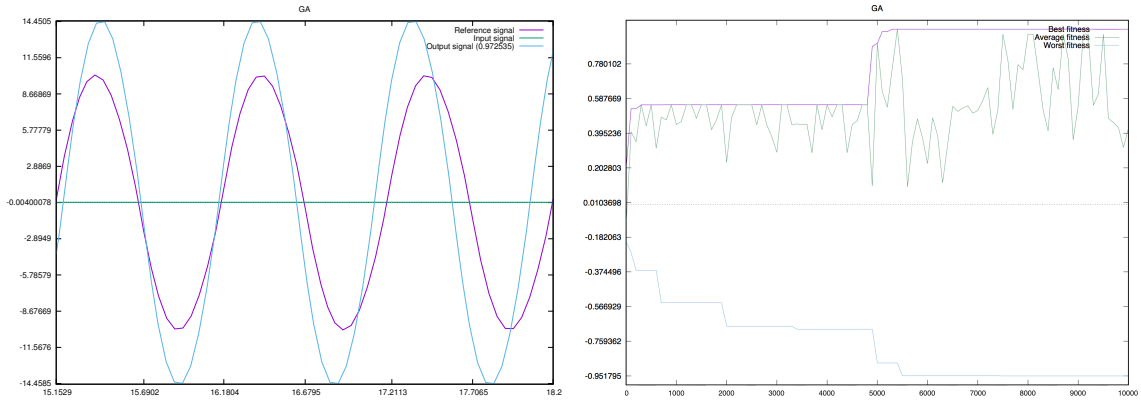
V této podkapitole jsou uvedeny výsledky experimentů genetického algoritmu s použitím vylepšené fitness funkce s Pearsonovým korelačním koeficientem. Na rozdíl od fitness funkce s Pearsonovým korelačním koeficientem je pravděpodobnost výskytu křížení 80% a mutace 20%. Jde o standardní nastavení pravděpodobností výskytu pro variační operátory, kde křížení je hlavním operátorem.

Testování probíhalo na testovací soustavě č. 2 a jeho statistické výsledky jsou zapsány v tabulce 6.6.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
40	28	0	69.9997	0.0	118

Obrázek 6.6: Statistické údaje experimentů kombinace: oscilátor s Wienovým článkem, GA, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Z výsledku lze vyčíst, že tato fitness funkce ve spojení s GA byla velice úspěšná, kde se reálná úspěšnost pohybuje kolem 70%. A to bez nalezení falešně pozitivních řešení. Na ukázce 6.7a lze vidět, že výstupní signál nalezeného řešení je téměř shodný s referenčním signálem. Z pohledu fitness funkce i požadavků na výstup jde téměř o dokonalé řešení. Přitom hodnoty součástek nalezeného řešení a referenčního řešení se nerovnají. Hodnoty součástek tohoto výsledku jsou $C_1 = 8.73 \cdot 10^{-8}F$, $C_2 = 8.73 \cdot 10^{-8}F$, $R_1 = 1833.67\Omega$, $R_2 = 1833.67\Omega$, $R_{10} = 91908.8\Omega$, $R_{11} = 167936\Omega$ a $R_{12} = 29804\Omega$. Na obrázku 6.7b je zobrazen průběh hodnoty fitness funkce v genetickém algoritmu.

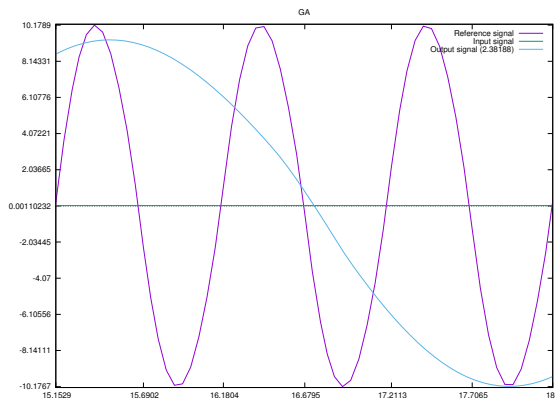


(a) Výstupní signál kandidátního řešení.

(b) Průběh hodnoty fitness funkce.

Obrázek 6.7: Ukázka úspěšného řešení v kombinaci: oscilátor s Wienovým článkem, GA, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

Příklad špatného nalezeného řešení je zobrazen na obr. 6.8. Amplituda signálu je v pořádku, ale frekvence je příliš malá oproti referenčnímu signálu. Lze vidět velkou šířku půl periody.



Obrázek 6.8: Ukázka falešně špatného řešení v kombinaci: oscilátor s Wienovým článkem, GA, vylepšená fitness funkce s Pearsonovým korelačním koeficientem.

6.1.3 Poziční fitness funkce

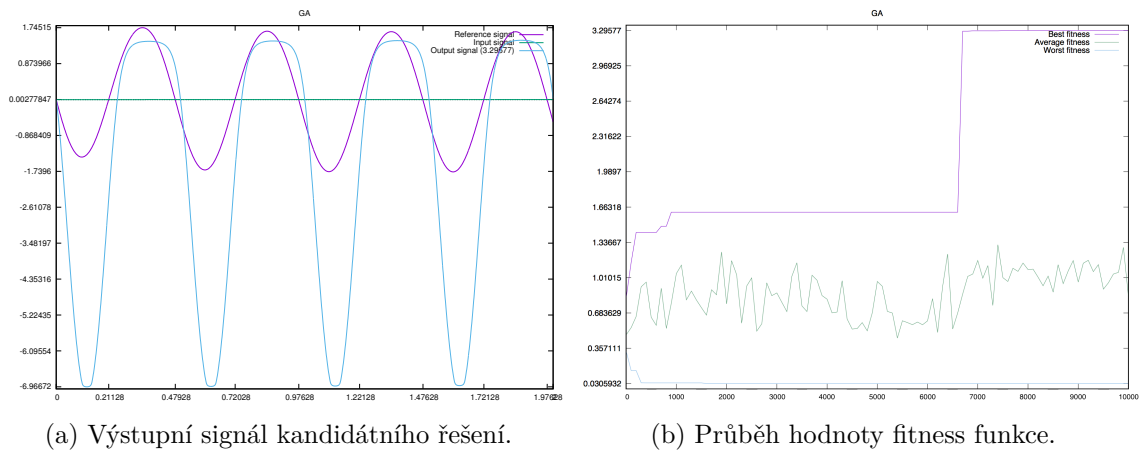
V podkapitole jsou shrnuty výsledky experimentů genetického algoritmu s poziční fitness funkcí. Stejně jako u vylepšené fitness funkce s Pearsonovým korelačním koeficientem je pravděpodobnost výskytu křížení 80% a mutace 20% a tím je opět zachováno standardní nastavení pravděpodobností výskytu pro variační operátory.

Testování probíhalo na testovací soustavě č. 1 a jeho statistické hodnoty jsou zapsány v tabulce 6.9.

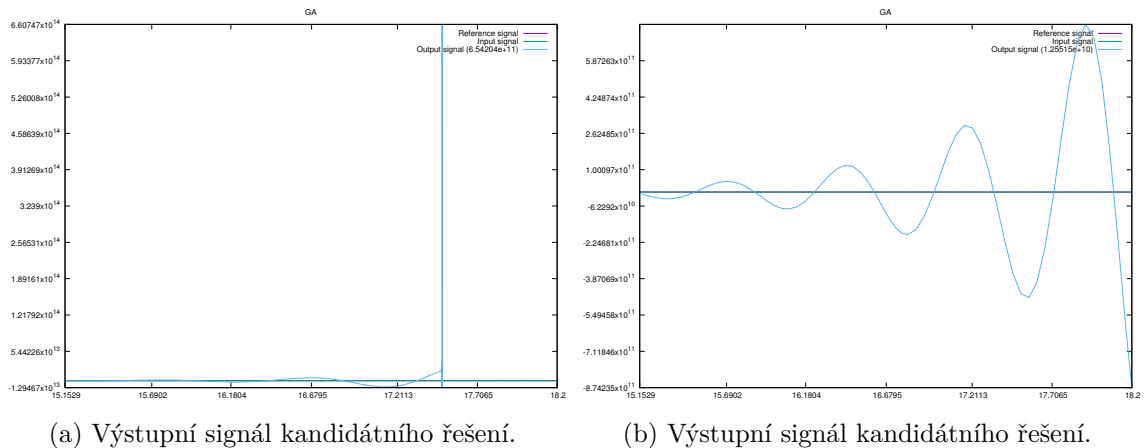
Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
40	23	6	57.4997	14.99939	41

Obrázek 6.9: Statistické údaje experimentů kombinace: oscilátor s Wienovým článkem, GA, poziční fitness funkce

Ze statických výsledku vyšlo, že tato fitness má 57% úspěšnost. Našli se i falešně pozitivních výsledků, které činí kolem 15% ze všech testů. Na obrázku 6.10a je ukázka správného řešení, kde amplituda záporných půl period je výrazně větší.



Obrázek 6.10: Ukázka úspěšného řešení v kombinaci: oscilátor s Wienovým článkem, GA, poziční fitness funkce.



Obrázek 6.11: Ukázka falešně pozitivních řešení v kombinaci: oscilátor s Wienovým článkem, GA, poziční fitness funkce.

Z ukázek falešně pozitivních výsledků na obrázcích 6.11a a 6.11b lze vidět, proč se našli falešně pozitivní řešení. Jde o stejný jev jako v případě testování ES na jednostupňovém zesilovači. Frekvence obou řešení je přijatelná, ale není splněna amplitudová podmínka. Amplitudy řešení jsou na první pohled nesmyslně vysoké. Jev je způsoben neomezením

maximální amplitudy touto fitness funkcí, frekvence je přijatelná a tudíž se i přijme vysoká hodnota z porovnání amplitud.

6.2 Evoluční strategie

V podkapitole jsou uvedeny všechny experimenty evoluční strategie na oscilátoru s Wienovým článkem. Cíl je shodná jako u genetického algoritmu. Tedy nalezení výsledku, kde signál kandidátního řešení bude co nejvěrnější referenčnímu signálu. Tzn. mít co nejpodobnější frekvenci signálu a také i amplitudy vzhledem k referenčnímu řešení. Řešení se považuje za přijatelné, pokud frekvence a amplitudy splňují odchylky od referenčního řešení, kdy je možné, aby kandidátní řešení mělo větší amplitudy signálu, než referenční řešení.

Opět se pro všechny testy používá velikost populace 10 jedinců. Nastavení je shodné s nastavením ES v experimentu jednostupňového zesilovače. Tedy používání pouze mutace jakožto variační operátor, který se aplikuje na všechny jedince. Z toho důvodu není potřeba výběru rodičů, mutování jsou všichni jedinci z populace. Mutace opět používá Gaussovou distribuci pravděpodobnosti společně autoevolucí, kde geny mají vlastní mutační krok. Výpočet počátečního mutačního kroku je dán vztahem $\sigma = (max - min) * 0.07$, kde max je maximální hodnotu, kterou gen/součástka může nabývat. Proměnná min je minimální hodnota, kterou lze genu nastavit. Stejně jako u GA je i zde 10000 generací v evoluční strategii pro všechny testy. Ve všech případech fitness funkcí je použité stejné nastavení evolučního algoritmu.

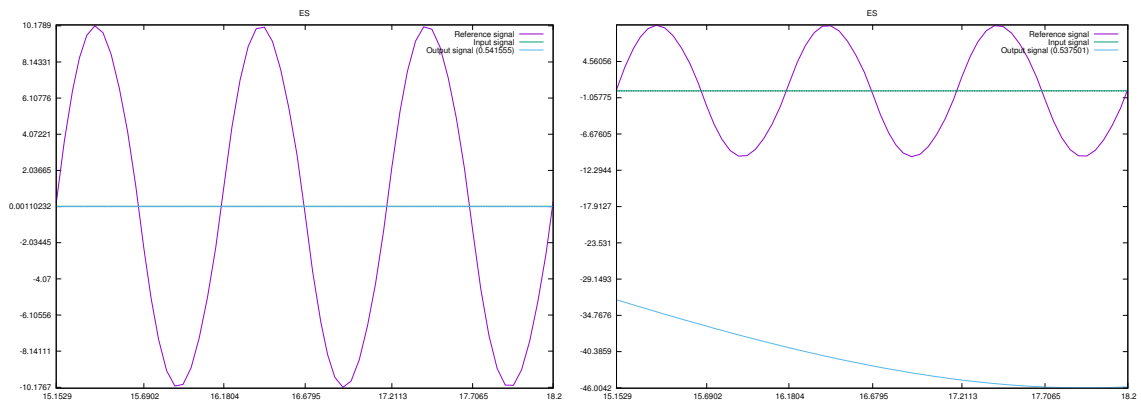
6.2.1 Fitness funkce s Pearsonovým korelačním koeficientem

V této části jsou uvedeny výsledky experimentů evoluční strategie s fitness funkcí s Pearsonovým korelačním koeficientem. V tabulce 6.12 jsou uvedeny statistické hodnoty experimentu, který probíhal na testovací soustavě č. 1.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
40	0	0	0.0	0.0	41

Obrázek 6.12: Statistické údaje experimentů kombinace: oscilátor s Wienovým článkem, ES, fitness funkce s Pearsonovým korelačním koeficientem.

Ze statických výsledků plyne, že tato fitness funkce má 0% úspěšnost. Výsledky testů měli velkou tendenci sklouzávat k řešení ukázaném na obr. 6.13a, kdy signál je téměř konstantní, obvykle nulový. Druhým případem kterým se výsledky ubíraly je zobrazen na příkladě 6.13b, kde délka periody příliš velká.



(a) Výstupní signál kandidátního řešení.

(b) Výstupní signál kandidátního řešení.

Obrázek 6.13: Ukázka neúspěšných řešení v kombinaci: oscilátor s Wienovým článkem, ES, fitness funkce s Pearsonovým korelačním koeficientem.

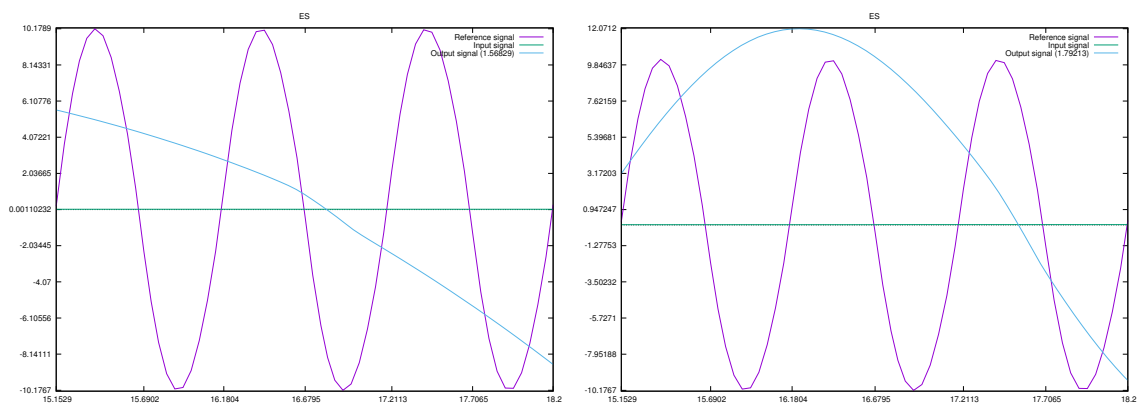
6.2.2 Vylepšená fitness funkce s Pearsonovým korelačním koeficientem

V kapitole jsou popsány výsledky experimentů evoluční strategie s použitím fitness funkce s Pearsonovým korelačním koeficientem. V tabulce 6.14 jsou uvedeny statistické údaje experimentu, který probíhal na testovací soustavě č. 1.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
40	0	0	0.0	0.0	56

Obrázek 6.14: Statistické údaje experimentů kombinace: oscilátor s Wienovým článkem, ES, fitness funkce s Pearsonovým korelačním koeficientem.

Jako v předchozím experimentu, i zde má fitness funkce 0% úspěšnost. Na obrázcích obr. 6.15a a 6.15b, jsou zobrazeny nejčastěji dosažené výsledky kandidátních řešení.



(a) Výstupní signál kandidátního řešení.

(b) Výstupní signál kandidátního řešení.

Obrázek 6.15: Ukázka neúspěšných řešení v kombinaci: oscilátor s Wienovým článkem, ES, fitness funkce s Pearsonovým korelačním koeficientem.

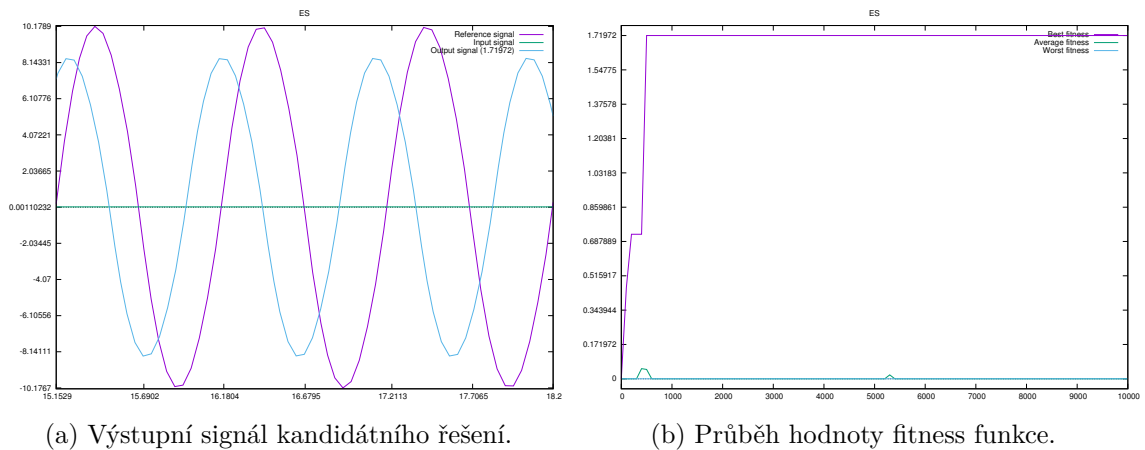
6.2.3 Poziční fitness funkce

Výsledky experimentů genetického algoritmu s využitím poziční fitness funkce jsou popsány v této kapitole. Statistické údaje experimentu prováděného na testovací soustavě č. 1 jsou zapsány v tabulce 6.16.

Počet testů	Počet úspěšných řešení	Počet falešně pozitivní řešení	Úspěšnost [%]	Falešná úspěšnost [%]	Průměrný čas běhu evoluce [m]
40	1	1	2.49939	2.49939	66

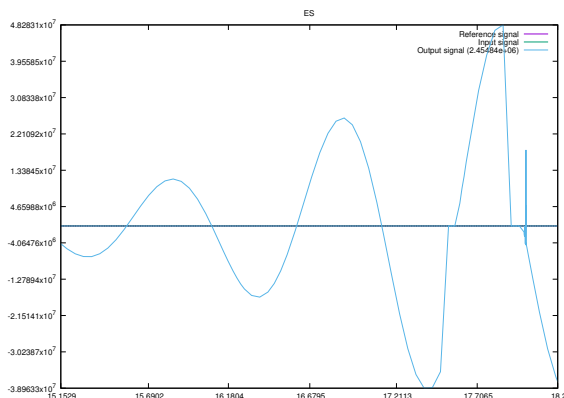
Obrázek 6.16: Statistické údaje experimentů kombinace: oscilátor s Wienovým článkem, ES, poziční fitness funkce.

Hodnoty v tabulce statistických údajů ukazují, že na rozdíl od experimentů testujících ostatní fitness funkce se dospělo k nalezeným řešením. Úspěšné řešení je pouze jedno a tak úspěšnost se pohybuje kolem 2.5%. Našlo se i jedno falešně pozitivní řešení a tak celkové zastoupení falešných řešení v testech je také 2.5%. Úspěšné řešení je ukázáno na obrázku 6.17a a průběh fitness funkce při hledání tohoto řešení na obr. 6.17b. Lze vidět, že výsledný signál je posunutý a kratší délku periody. I amplituda je menší než u referenčního signálu. Signál nalezeného řešení není ideální, ale je přijatelný. Na průběhu fitness funkce lze vidět, že řešení se našlo již po uplynutí 1000 generací, tedy se našlo brzo po začátku evoluce.



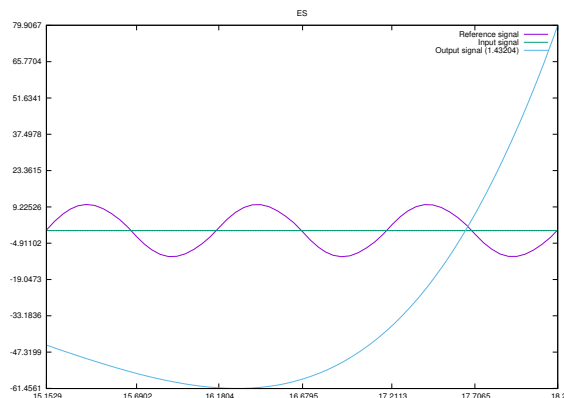
Obrázek 6.17: Ukázka úspěšného řešení v kombinaci: oscilátor s Wienovým článkem, ES, poziční fitness funkce.

Ve výsledcích se také objevil jeden falešně pozitivní výsledek, který je zobrazem na obr. 6.18. Na řešení lze opět pozorovat porušení amplitudové podmínky.



Obrázek 6.18: Ukázka falešně dobrého řešení v kombinaci: oscilátor s Wienovým článkem, ES, poziční fitness funkce.

Příklad špatného kandidátního řešení je ukázán na obr. 6.19, na kterém jde vidět příliš velkou délku periody a amplitudu.



Obrázek 6.19: Ukázka falešně špatného řešení v kombinaci: oscilátor s Wienovým článkem, ES, poziční fitness funkce.

6.2.4 Shrnutí a diskuze

V kapitole jsou popsány experimenty s oscilátorem s Wienovým článkem, kde se jako při experimentu s jednostupňovým zesilovačem vyzkoušely 2 typy evoluce a 3 typy fitness funkcí, jež se testovaly pro každý typ evoluce. Jedná se o stejné typy evoluce, fitness funkce i s jejich vlastnostmi, jako tomu je u experimentu jednostupňového zesilovače.

V případě genetického algoritmu se testovaly 3 druhy fitness funkcí, mezi kterými jsou fitness funkce s Pearsonovým korelačním koeficientem, vylepšená fitness funkce s Pearsonovým korelačním koeficientem a poziční fitness funkce. Fitness funkci s Pearsonovým korelačním koeficientem je podle výsledku ta nejméně úspěšnou fitness funkcí. Její úspěšnost se pohybuje kolem 12.5% a navíc se našlo kolem 22% falešně pozitivních výsledků. Fitness funkce opět dobře hlídá podobnost frekvence výstupního signálu s referenčním signálem. Dovoluje najít i signály vykazující větší amplitudu, než je tomu u referenčního řešení. U oscilátoru tato vlastnost může být žádaná, ale také i ne, v tomto experimentu je to přijatelné.

Tato fitness funkce se hodí k použití, kde je nutné mít co nejlepší frekvenci a současně kde je snaha dosáhnout lepšího řešení, než je referenční řešení. Ale nesmí vadit její účinnost a kontrola falešně pozitivních řešení. Dále tu je vylepšená fitness funkce s Pearsonovým korelačním koeficientem, která také velmi dobře kontroluje frekvenci výsledného řešení. Fitness funkce hlídá i velikost amplitudy. Opět se ukázala nejvyšší úspěšnost této fitness funkce, která je kolem 70%. Tato fitness funkce v testech nevykázala žádné falešně pozitivní výsledky, což je velkou výhodou. Vzhledem ke kontrole amplitudy fitness funkce nedovoluje získat lepšího řešení než je to referenční, resp. umožňuje získat zanedbatelné zlepšení. Nicméně tato nevýhoda může být v případě oscilátorů zanedbatelná, záleží na požadavcích pro daný výstup. Fitness funkce se hodí hlavně k případům, kde není možnost kontroly falešně pozitivních řešení, je důležité dosáhnout co největší účinnosti, i za cenu toho, že nelze získat lepší řešení než je referenční. Poziční fitness funkce je poslední testovanou fitness funkcí. Úspěšnost fitness funkce se řadí na druhé místo při svých 57%. Došlo ale také k nalezení 15% falešně pozitivních výsledků. Fitness funkce nemá tak dobrou kontrolu frekvence signálu, jako je tomu u předchozích dvou fitness funkcí. Ale tato fitness funkce nehlídá maximální amplitudu a tedy více umožňuje nalézt řešení, které mají větší amplitudu než referenční signály. Ukázalo se, že v tomto případě má fitness funkce úspěšnost kolem 48%, což není špatný výsledek. Fitness funkce je vhodná pro případy, kdy je požadavkem dosáhnout větší amplitudy signálu řešení než je amplituda referenčního signálu.

V evoluční strategii došlo k velmi negativním výsledkům. V případě fitness funkce s Pearsonovým korelačním koeficientem a vylepšené fitness funkce s Pearsonovým korelačním koeficientem se nedosáhlo žádných výsledků, tedy jejich úspěšnost byla 0%. Tudíž jsou tyto fitness funkce při použití evoluční strategie pro tento experiment nepoužitelné. Jediná poziční fitness funkce našla jeden správný výsledek, takže její úspěšnost činí 2.5%. Vzhledem k tomuto výsledku se i tato fitness ukázala být v tomto případě nepoužitelná.

Tak jako v případě experimentu s jednostupňovým zesilovačem se i zde projevilo, že genetický algoritmus je lepší, než evoluční strategie a je tedy vhodnější ho použít ve všech případech.

Kapitola 7

Závěr

V rámci diplomové práce bylo cílem se seznámit a nastudovat si vybrané typy analogových obvodů a možnosti jejich simulace systémem SPICE. Dalším cílem bylo se seznámit s problematikou optimalizace pomocí evolučních algoritmů. Na základě těchto znalostí bylo úkolem navrhnout systém pro optimalizaci zvolených analogových obvodů pomocí evoluce. V řešení systému pro optimalizaci je potřeba se zabývat různými fitness funkcemi. Na tomto systému bylo cílem ověřit, na kolik je evoluce schopná optimalizovat již navržený obvod.

Z analogových obvodů byly k optimalizaci vybrány zesilovače a oscilátory. K ověření možnosti optimalizovat analogové obvody pomocí evolučních algoritmů byly vybrány genetické algoritmy a evoluční strategie.

V rámci diplomové práce byl implementován systém pro testování možnosti optimalizace analogových obvodů pomocí evoluce. Tento systém podporuje genetické algoritmy a evoluční strategie. Tyto 2 druhy evolučních algoritmů lze použít s 3 různými fitness funkcemi. Na tomto systému byla provedena sada testů ukazujících možnosti evoluční optimalizace vybraného zesilovače a oscilátoru. Výsledky těchto testů byly zaznamenány v této práci. Dále je zde popsána teorie o analogových obvodech a evolučních algoritmech.

V rámci vyhodnocení testů bylo zjištěno, že evoluční optimalizace zesilovačů a oscilátorů pomocí genetického algoritmu nebo evoluční strategie je možné. Podle požadavků na výsledné řešení lze pomocí evoluce dosáhnout až 80% úspěchu nalezení řešení v případě zesilovače a 70% úspěchu v případě oscilátoru. S menším procentuálním úspěchem lze i nalézt řešení, která jsou lepší, než je referenční řešení vypočteno konvenční metodou. Zajímavým zjištěním po vyhodnocení testů bylo, že i přesto, že jsou evoluční strategie určené na práci s desetinnými čísly, tak genetický algoritmus byl ve všech případech úspěšnější.

Při dalším vývoji je zde prostor pro zlepšení fitness funkcí, zejména odstranění jejich nedostatků, aby se eliminovaly nebo alespoň snížily nálezy falešně pozitivních řešení. Dále je možnost přidat implementaci pro různé možnosti selekce, více druhů křížení a mutací, více druhů obnovy populace, atd. a ty možnosti otestovat. V neposlední řadě je tu možnost vyzkoušet evoluční optimalizaci společně s evolučním návrhem analogových obvodů a ověřit možnost této automatizace.

Literatura

- [1] Neodarwinismus. DNA.
URL <http://www.prf.jcu.cz/zmb/menu/neodarwinismus-dna.html>
- [2] Ngspice.
URL <http://ngspice.sourceforge.net/presentation.html>
- [3] Wien Bridge Oscillator.
URL <http://www.ecircuitcenter.com/Circuits/opwien/opwien.htm>
- [4] Bešta, M.: Zesilovače.
URL
<http://www.mbest.cz/wp-content/uploads/2013/01/T3.1-Zesilova%C4%8De.pdf>
- [5] Biolek, D.; Hájek, K.; Krtička, A.: *Analogové elektronické obvody*. Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně, 2007.
- [6] Biolek, D.; Hájek, K.; Krtička, A.; aj.: *ELEKTRONICKÉ OBVODY I*. Brno: Fakulta vojenských technologií, UNIVERZITA OBRANY, 2006.
- [7] Brabazon, A.; O'Neill, M.; McGarraghy, S.: *Natural Computing Algorithms*. Natural Computing Series, Springer Berlin Heidelberg, 2015, ISBN 9783662436318.
URL <https://books.google.cz/books?id=wAi0CgAAQBAJ>
- [8] Černík, M.: *Elektrické obvody: teorie a příklady*. Technická univerzita v Liberci, 2014.
URL <https://dspace.tul.cz/bitstream/handle/15240/7152/Elektrick%C3%A9%20obvody%20teorie%20a%20p%C5%99%C3%ADklady.pdf?sequence=3>
- [9] Dostál, T.: *Analogové elektronické obvody*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2004.
- [10] Jan, K.: *Elektronika III*. BEN - technická literatura, druhé vydání, 2003, ISBN 80-7300-182-9, 112 s.
- [11] Kvasnička, V.; Pospíchal, J.; Tiňo, P.: *Evoluční algoritmy*. Vydavatelství STU Bratislava, 2000, ISBN 80-227-1377-5, 215 s.
- [12] Malina, V.: *Poznáváme elektroniku I*. KOPP, třetí vydání, 2003, ISBN 80-7232-039-4, 222 s.
- [13] Morris, J.: *Analogue Electronics*. Arnold, 1999, ISBN 9780340719251.
URL <https://books.google.cz/books?id=z3c0AAAACAAJ>

- [14] Nenzi, P.; Vogt, H.: *Ngspice Users Manual*. 2014.
URL <http://ngspice.sourceforge.net/docs/ngspice26-manual.pdf>
- [15] Popelka, J.: *Evoluční výpočetní techniky*. Bakalářská práce, Vysoké učení technické v Brně, Brno, 2011.
- [16] Schwarz, J.; Sekanina, L.: *Aplikované evoluční algoritmy EVO - Studijní opora*. 2006.
- [17] Serafín, Č.: LC oscilátor.
URL
<http://www.kteiv.upol.cz/uploads/soubory/serafin/frvs12/LC-oscilator.pdf>
- [18] Všetěčka, J.: *Opodstatněnost evoluční teorie jako primární teorie v rámci vzdělávání české populace*. Diplomová práce, Univerzita Palackého v Olomouci, Olomouc, 2012.
- [19] Zou, K. H.; Tuncali, K.; Silverman, S. G.: Correlation and Simple Linear Regression. *Radiology*, ročník 227, č. 3, 2003: s. 617–628, doi:10.1148/radiol.2273011499, PMID: 12773666, <http://dx.doi.org/10.1148/radiol.2273011499>.
URL <http://dx.doi.org/10.1148/radiol.2273011499>

Přílohy

Seznam příloh

A	Obsah CD	56
B	Návod k použití systému	57
B.1	Instalace prerekvizit	57
B.2	Instalace sytému	57
B.3	Spuštění testu	57
B.4	Parametry systému	58

Příloha A

Obsah CD

- `/bin` Složka obsahující spustitelný systém a testovací skript.
- `/circuits` Netlisty testovaných obvodů.
- `/doc` Složka obsahující tento dokument a jeho zdrojové kódy.
- `/lib` Knihovny Ngspice nutné k běhu systému.
- `/ngspice` Zdrojové soubory programu Ngspice.
- `/output` Složka určena pro výstupy systému.
- `/src` Zdrojové kódy systému.
- `/CMakeLists.txt` Vstupní soubor pro program cmake.
- `/README.txt` Popis k použití systému.

Příloha B

Návod k použití systému

Systém pro evoluční optimalizaci analogových obvodů je podporován na operačních systémech Linux a macOS. Pro řešení systému jsou použité standardní knihovny a knihovna Ngspice, jejíž zdrojové soubory jsou přiložené k této práci. Systém byl testován na PC s operačním systémem Linux Mint 18.1 a PC s operačním systémem macOS 10.12.4.

B.1 Instalace prerekvizit

Před instalací samotného systému je potřeba mít nainstalovanou Boost knihovny a program Gnuplot. Poté je potřeba zkompileovat knihovnu Ngspice tak, aby se nacházela ve složce. Toho se docílí vykonáním následujících kroků:

1. přemístit se do složky ngspice
2. spustit příkaz `./configure --disable-debug --with-ngshared --prefix='pwd'/../lib/`
3. spustit příkaz `make`
4. spustit příkaz `make install`

.

B.2 Instalace systému

1. přemístit se do hlavní složky
2. spustit příkaz
3. spustit příkaz `make`

Nyní se system nachází ve složce bin pod jménem Thesis.

B.3 Spuštění testu

Po přemístění do složky bin lze pustit test dvěma způsoby:

1. příkazem `./Thesis` a doplněním patřičných parametrů, které jsou popsány níže [B.4](#)
2. příkazem `python3 test_script.py` se spustí sada testů, jejichž výstup se bude nacházet ve složce output

B.4 Parametry systému

1. n - název testu
2. i - cesta k netlistu testovaného obvodu
3. o - výstupní složka, kde se ukládají výsledky systému, hodnota musí končit zpětným lomítkem /
4. e - nastavuje typ evolučního algoritmu, nabývá 2 hodnot
 - (a) GA - genetický algoritmus
 - (b) ES - evoluční strategii
5. f - nastavuje typ fitness funkce, nabývá tří hodnot
 - (a) pearson - fitness funkce s Pearsonovým korelačním koeficientem
 - (b) better_pearson - vylepšená fitness funkce s Pearsonovým korelačním koeficientem
 - (c) position - poziční fitness funkce
6. g - počet generací pro evoluční algoritmus

V případě použití GA lze nastavit:

1. m - pravděpodobnost výskytu mutace, hodnota z intervalu $\langle 0,1 \rangle$
2. r - pravděpodobnost výskytu křížení, hodnota z intervalu $\langle 0,1 \rangle$