



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ALGORITMICKÉ OBCHODOVÁNÍ NA BURZE S VYU-  
ŽITÍM GENETICKÝCH ALGORITMŮ**

ALGORITHMIC TRADING USING GENETIC ALGORITHMS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. KAREL ČERVÍČEK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. IGOR SZÓKE, Ph.D.**

BRNO 2019

## Zadání diplomové práce



21848

Student: **Červíček Karel, Bc.**  
Program: Informační technologie    Obor: Počítačová grafika a multimédia  
Název: **Algoritmické obchodování na burze s využitím genetických algoritmů**  
**Algorithmic Trading Using Genetic Algorithms**  
Kategorie: Umělá inteligence

### Zadání:

1. Seznamte se se základy genetických algoritmů, strojového učení a intraday obchodování na burze.
2. Prostudujte dostupné techniky technické analýzy.
3. Definujte vhodná data pro trénování a testování metod pro algoritmické obchodování.
4. Navrhněte a implementujte systém pro generování a ohodnocení obchodních strategií.
5. Otestujte navržený systém na zvolených datech a systém vylepšete.
6. Zhodnoťte výsledky a navrhněte směry dalšího vývoje.
7. Vytvořte A2 plakátek a cca 30 vteřinové video prezentující výsledky Vaší práce.

### Literatura:

- Dle pokynů vedoucího

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Szóke Igor, Ing., Ph.D.**  
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.  
Datum zadání: 1. listopadu 2018  
Datum odevzdání: 22. května 2019  
Datum schválení: 10. května 2019

## Abstrakt

Obchodování na burze ve spojení s automatizací je široce probírané téma. Snahou této práce je posoudit využití optimalizačních metod a prostředků strojového učení pro efektivní a obecné zpracování finančních časových řad. Je navržen a otestován systém, který zpracuje signál a generuje optimální strategii.

## Abstract

Automatization is highly used in stock trading. The thesis tries to exploit optimization principles and machine learning. Developed and tested stock trading system processes financial time series and generates optimal strategy.

## Klíčová slova

Strojové učení, Forex, Genetické algoritmy, Gramatická evoluce, Obchodování, Burza, Robot, Automat

## Keywords

Machine learning, genetic algorithm, Forex trading, trading system

## Citace

ČERVÍČEK, Karel. *Algoritmické obchodování na burze s využitím genetických algoritmů*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szóke, Ph.D.

# Algoritmické obchodování na burze s využitím genetických algoritmů

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Igora Szókeho, Ph.D. Další informace mi poskytl Prof. Ing. Lukáš Sekanina, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Karel Červíček  
15. května 2019

## Poděkování

Chtěl bych poděkovat vedoucímu za podporu při nelehké implementaci systému a celkovým vedením, jak při práci postupovat. Také děkuji profesoru Sekaninovi za konzultace ohledně genetických algoritmů.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Obchodování . . . . .	3
1.2	Hlavní cíl . . . . .	4
1.3	Struktura práce . . . . .	5
<b>2</b>	<b>Forex</b>	<b>6</b>
2.1	Fundamentální analýza . . . . .	8
2.2	Technická analýza . . . . .	8
2.3	Závěr . . . . .	11
<b>3</b>	<b>Prostředky, techniky a poznatky pro návrh systému</b>	<b>14</b>
3.1	Generování popisu signálu na základě gramatiky . . . . .	14
3.2	Genetické algoritmy . . . . .	15
3.2.1	Gramatická evoluce . . . . .	17
3.2.2	Aplikace genetických algoritmů . . . . .	17
3.3	Neuronové sítě . . . . .	17
3.3.1	Trénování neuronové sítě . . . . .	18
3.3.2	Zpětná propagace chyby . . . . .	19
3.3.3	Promíchání dat . . . . .	19
3.3.4	Normalizace dat . . . . .	19
3.3.5	Redukce dimenze . . . . .	20
3.4	Finanční kalkul . . . . .	20
3.4.1	Reprezentace cenové změny . . . . .	20
3.4.2	Vlastnosti rozložení pravděpodobnosti finančních dat . . . . .	21
3.4.3	Lineární modely pro popis časových řad . . . . .	23
3.4.4	Autoregressive conditional heteroskedasticity . . . . .	24
3.5	Studium Forexu z hlediska praxe . . . . .	25
3.6	Závěr . . . . .	25
<b>4</b>	<b>Návrh systému</b>	<b>27</b>
4.1	Základní demonstrace . . . . .	28
4.2	Architektura prvního návrhu systému . . . . .	31
4.3	Generátor vět . . . . .	32
4.4	Překladač . . . . .	33
4.5	Optimalizační modul . . . . .	37
4.6	Evaluační modul . . . . .	39
4.7	Modul dopředné neuronové sítě . . . . .	40

<b>5</b>	<b>Testování a vyhodnocení problémů</b>	<b>42</b>
5.1	Test základního principu optimalizace . . . . .	42
5.2	Testy fitness funkce trailing stop . . . . .	43
5.3	Test finální optimalizace . . . . .	45
5.4	Ohodnocení podle nejlepšího sloupce bool . . . . .	47
5.5	Nasazení strategie na testovacích datech . . . . .	48
5.6	Nasazení neuronové sítě . . . . .	48
5.6.1	Vyhodnocení testů pro návrh finálního systému . . . . .	50
<b>6</b>	<b>Finální implementace</b>	<b>52</b>
6.1	Úvod . . . . .	52
6.2	Vytvoření jedince v populaci . . . . .	54
6.3	Překlad a evaluace . . . . .	56
6.4	Optimalizace strategií . . . . .	59
6.5	Závěr finální implementace . . . . .	61
6.6	Testování . . . . .	62
6.6.1	Test 1 . . . . .	63
6.6.2	Test 2 . . . . .	64
6.6.3	Test 3 . . . . .	65
6.6.4	Test 4 . . . . .	65
6.6.5	Přehled zisků na více párech . . . . .	66
6.6.6	Test nutnosti adaptace . . . . .	66
<b>7</b>	<b>Závěr</b>	<b>67</b>
7.1	Návrh dalšího vývoje . . . . .	67
	<b>Literatura</b>	<b>69</b>
<b>A</b>	<b>Obsah příloženého paměťového média</b>	<b>72</b>
<b>B</b>	<b>Manuál</b>	<b>73</b>
<b>C</b>	<b>Konfigurační soubor</b>	<b>74</b>
<b>D</b>	<b>Testy Metatrader</b>	<b>75</b>

# Kapitola 1

## Úvod

V úvodu je definován účel, cíl a popis, ale také zdůvodnění, proč práce vznikla. Vysvětluje hlubší souvislosti návrhu postupu, návrhu architektury a výběru zkoumané oblasti, kterou je obchodování na Forexu. Otevírá témata, která přímo ovlivňují celkový návrh. Pokud čtenář není obeznámen s obchodováním na burze, pak by nemusel zcela pochopit některé kroky v návrhu či jistá omezení.

### 1.1 Obchodování

Díky moderním informačním technologiím se principy obchodování na burze mění již několik let. Z fyzicky řešených objednávek se přešlo na elektronické příkazy. Dříve měl broker na burze obchodníka, který objednávku vyřizoval. Dnes jsou veškeré objednávky vyřizovány elektronicky. Broker přijímá objednávky svých klientů prostřednictvím platformy. Obchodování na trzích je dnes prováděno, jak lidmi, tak automaticky.

Vysokofrekvenčnímu obchodování se od devadesátých let začalo věnovat několik firem [3]. Také dnes existuje mnoho obchodníků, kteří trh analyzují fundamentálně nebo technicky za pomoci podpůrných programů a objednávky zadávají samostatně. Zadání příkazu obchodníkem nebo obchodním programem je ovlivněno vývojem ceny. Cena je tvořena nabídkou a poptávkou. Pokud je poptávka po dané komoditě či jiné hodnotné entitě vyšší než nabídka, pak její cena roste. Stejně tak, pokud je nabídka vyšší než poptávka, pak cena dané komodity klesá. Za tímto jednoduchým principem stojí komplexní systém ekonomiky celého světa.

Trh je ovlivňován mnoha faktory, jako jsou počasí, války, zprávy ze světa, zprávy z ekonomiky, ale také objemné objednávky bank, obchodní algoritmy a mnoho dalších. Vysokofrekvenční obchodování je založeno na algoritmech, které jsou schopny v rámci sekund provést obchod například na základě různých cen z několika burz nebo na základě stochastického modelování. Pro odvětví vysokofrekvenčního obchodování je nutné mít co nejrychlejší přístup k aktuálním informacím a také co nejkratší dobu přístupu přímo na servery burzy. Plnit tyto podmínky není snadné a je k tomu zapotřebí relativně velký kapitál. Tato práce se zaměřuje na intradenní obchodování prostřednictvím brokera 2. Výhodou je možnost obchodování s malým kapitálem a nízko nákladový přístup na burzu. I s tímto přístupem je možné zavést automatické obchodování nebo zavést automatického asistenta, který dohlíží na trh a upozorňuje na obchodní příležitosti v rámci minut, hodin a dnů. Většina brokerů poskytuje programové rozhraní pro nasazení obchodního systému.

## 1.2 Hlavní cíl

Text se zabývá vytvořením systému, který bude zpracovávat a analyzovat časové řady za účelem automatického sestavení řešení, jež pro časovou řadu hledáme. Řešený problém bude ohodnocen funkcí, kterou zadavatel určí. Problém nalezení optimální analýzy časové řady bude řešen maximalizací nebo minimalizací funkce pro ohodnocení. I přesto, že primárně se práce zabývá obchodováním na Forexu, je snahou vytvořit systém obecný a případně použitelný i pro jiné druhy signálu.

Vzhledem k tomu, jak se obchodování na burzách vyvíjelo, je dnes pro obchodníka takřka nemožné zpracovat všechna data, která trh potenciálně ovlivňují. Obchodník se snaží na základě aktuálních informací a historického vývoje trhu určit, zda uvažovanou entitu koupí nebo zda je čas na prodej. Obchodník využívá fundamentální [2.1](#) a technickou analýzu [2.2](#). Motivací této práce je automatizace strojového zpracování dat. Fundamentální data jsou většinou v textovém formátu, který je také možné zpracovávat strojově [\[21\]](#). Získávání dat z textu a zpracování přirozeného jazyka je samostatná obsáhlá disciplína informatiky.

Vedle zpracování fundamentálních dat je možné zakládat obchodní strategie na analýze historických dat. Z historického vývoje ceny je možné statisticky ohodnotit, zda obchod uskutečnit [2.2](#). Pro ziskový obchod je nutné správně vyhodnotit mnoho dat, která mohou i nemusí být relevantní vzhledem k obchodnímu záměru. To otevírá otázku, jaké přístupy zvolit. Historická data jsou popsána matematickými modely z oblasti finančního kalkulu. Na jejich základě je možné pochopit charakter burzovních dat a vyvodit přístupy pro jejich zpracování. Tato tematika je také popsána v mnoha knihách zaměřených na technickou analýzu [\[10\]](#). Většina obchodních systémů bývá parametrizována, a tak je také možné brát v úvahu optimalizační techniky. Obchodní systémy tvoří množinu ziskových a ztrátových obchodů, které je možné klasifikovat pomocí prostředků strojového učení [\[1\]](#). Existují práce, které se touto problematikou zabývají. Jak bylo zmíněno, například práce [\[15\]](#) je kolekcí metod pro zpracování fundamentálních dat za účelem predikce kurzu. Forexový kalendář je jedním z kanálů, kde jsou aktuální fundamentální data. Práce [\[19\]](#) se zabývá zpracováním právě takovýchto dat. Některé práce se také zabývají optimalizací technické analýzy pomocí genetických algoritmů. Například v článku [\[17\]](#) je optimalizován obchodní systém založený na pěti parametrech, které určují vstup do pozice, a pěti parametrech pro jejich ukončení.

Podstatným zdrojem informací je kniha [\[5\]](#) řešící predikci časových řad na základě gramatik. V této knize jsou popsány možnosti generování jazyka pro zpracování časové řady. Kniha využívá i možnosti strojového učení, konkrétně support vector machines. Na podobném principu je založen článek [\[2\]](#), který využívá gramatickou evoluci popsanou i v této práci. Zmíněné strojové učení je také často skloňovaným nástrojem a bylo vydáno několik článků a knih, které toto téma popisují. Zajímavé je například využití dopředné neuronové sítě, které je popsáno autorem Robertem J. Eydenem [\[8\]](#). Posuzuje vliv objemu dat na výsledky predikce pomocí neuronové sítě. Vstupem jsou různorodá data, jako například indexy, ekonomické statistiky, zprávy či technické indikátory. V závěru jsou posouzeny výsledky založené na různých datových zdrojích v různém objemu. Posouzení ukázalo, že 20 a 63 zdrojů nemělo úměrně rozdílné výsledky. Robustnost a objem dat neměly přímý vliv na lepší predikci. Tento výsledek opět pokládá otázku, jak určit, která data jsou relevantní pro správnou generalizaci problému. Dopředná neuronová síť se jeví jako vhodný nástroj pro zpracování finančních dat. Například konkrétně článek [\[27\]](#) používá tento návrh pro predikci.

Genetické algoritmy mohou být využity i pro nastavení parametrů neuronové sítě, optimalizaci architektury nebo optimalizaci zpracování dat. V tomto článku [\[6\]](#) využili au-



toři právě genetické algoritmy pro optimální nastavení parametrů, dále analýzu hlavních komponent za účelem snížení dimenzionality, kterou vyřešili odstraněním nerelevantních dat. Zmíněné informace spolu s knihou [4] stanovily směr této práce. Jejím cílem je vhodně navrhnout architekturu obchodního systému, která bude efektivně určovat relevanci dat pro technickou analýzu. Určovat ziskové strategie a rozhodování pozic bude potenciálně podpořeno neuronovou sítí. Vzhledem k možným variacím, jak historická data vyhodnocovat a stavět možné obchodní strategie, je nutné efektivně generovat množinu výpočtů, obecně optimalizovat potenciálně výdělečné strategie a určit, kdy a jak bude využito strojové učení. Vzhledem k mnoha faktorům a možným přístupům, jak tento problém řešit, by měla architektura umožňovat eventuální rozšíření. Výsledný systém bude testován na měnovém páru a bude zaměřen na získání obchodních strategií. Avšak plánovaná obecnost by měla systém postavit do pozice, ve které bude stačit jeho minimální rozšíření pro zpracování i jiných dat.

### 1.3 Struktura práce

Práce testuje a předpokládá pouze forexová data. Výběr studovaných materiálů odpovídá cíli primárně vytvořit systém, který bude přínosem pro obchodování na Forexu. V kapitole 2 uvádím, jak celý proces obchodování funguje. Popisuji rizika, která obchodování přináší spolu s nástrahami, které přináší obchodování přes brokera. Další kapitoly jsou teoretickým podkladem k samotnému systému. Kapitola 3.3 uvádí princip primárně dopředné neuronové sítě. Studium neuronové sítě jsem do práce zanesl pro její **případné** využití ve smyslu testu robustnosti a obecnosti systému z pohledu rozšiřování funkcionality a využívání dalších metod optimalizace a strojového učení. Kapitola 3.2 popisuje, jak genetické algoritmy fungují. Každá teoretická část je zakončena shrnutím vzhledem k cíli práce. Kapitola 3.4 pouze okrajově popisuje některé modely pro stochastický popis finančních dat.

Vzhledem k tomu, že tato práce by měla mít především praktický dopad, je největší pozornost věnována kapitolám 4 a 6. Do práce jsem se rozhodl zahrnout první návrh systému. Na jeho základech je postaven systém finální 6, ve kterém jsou opravené nepřesnosti, chyby a hlavně optimalizační nedostatky, které jsem zjistil při první implementaci. Čtenář obeznámený s genetickými algoritmy, gramatikami, umělými neuronovými sítěmi a především s principy Forexu může cílit přímo na tyto kapitoly.

## Kapitola 2

# Forex

Volba finančních trhů byla provedena na základě charakteristiky dat a také obecných vlastností. Forex je zkratkou pro Foreign Exchange. Jde o trh pro směnu měnových párů. Pro nás je i vzhledem k této práci zajímavé, že jde o nejlikvidnější a největší trh na světě, na kterém participují a ve kterém směňují aktiva a deriváty banky, fondy a jiné finanční instituce.

Příkladem samotným je postavení našeho systému v roli investora, kdy jeho úkolem bude spekulace o budoucím pohybu různých měn za účelem zisku. Trh je tedy ovlivňován ekonomikami celého světa a odráží jejich charakter. Obchodníci a brokeri jako tvůrci trhu zveřejňují svoji nabídku a poptávku, která musí být na obou stranách potvrzena, neboť trh nemá centrální burzovní knihu.

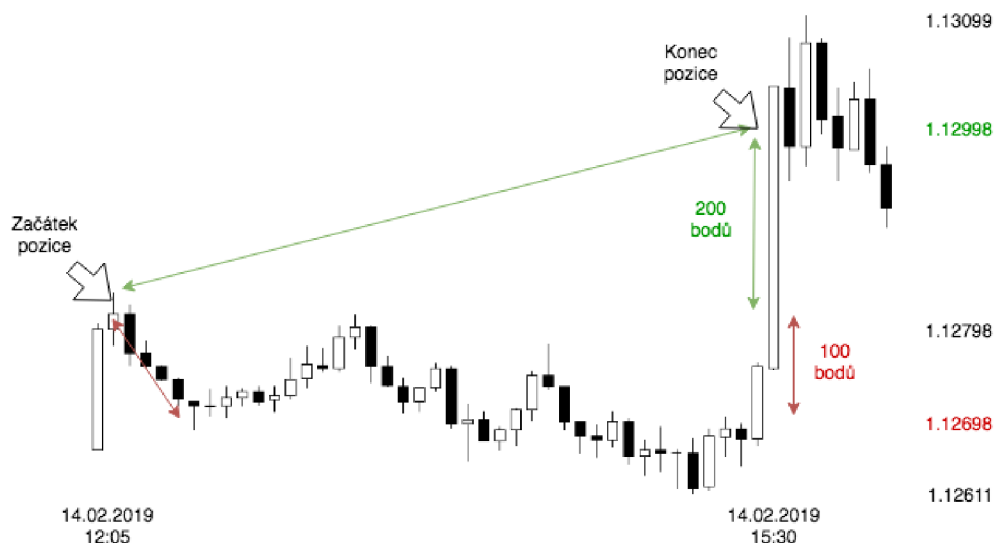
Je zajímavé, co stojí za vznikem Forexu. Vývoj vzniku přesně vysvětluje, na čem je cena dané měny založena. Dříve byla měna daného státu podložena zlatem. Kurz mezi měnami byl pak určen poměrem zlata, které daný stát měl. Tomuto finančním principu se říká zlatý standard. Z důvodů tlaků na ekonomiku během první světové války byl tento standard zrušen, neboť zlatý standard ekonomiku brzdil a státy byly nuceny peníze tisknout bez podložení. Po válce došlo k usnesení, že hlavní podkladovou měnou se stane americký dolar, který byl vyrovnán zlatem. Postupem času byly nároky na rezervy zlata neúnosné. Byl zaveden systém plovoucích měnových kurzů, který je dodnes aktuální. Měna je tak podložena důvěrou držitelů a její cena je určena nabídkou a poptávkou [14].

Forex je dostupný i pro obchodníky s malým kapitálem a bez přímého přístupu na burzu. Forex broker ve smyslu finančních trhů je společnost, která poskytuje přístup na Forex. Zajišťuje software a server pro správu obchodů svých klientů a také **finanční páku**. **Finanční páka** dovoluje obchodovat s malým kapitálem a dosahovat velkých zisků resp. ztrát. Broker podkládá kapitál obchodníka v násobku finanční páky. Pokud obchodník využívá páku 1:100, pak je jeho kapitál stonásobně větší pro otevírání obchodních pozic. Pro otevření pozice je nutné disponovat alespoň takovým kapitálem, který stačí pro složení jistiny. Jistina slouží jako krytí brokera z hlediska úvěrového rizika. Pokud obchodník neodhadne situaci a ponechá běžet pozice ve ztrátě tak velké, že se vyčerpá jeho volná jistina, pak bude vyzván k doplnění prostředků na účet, jinak jsou jeho pozice automaticky uzavřeny. Tento krok je ochranou proti zadlužení. Dalšími skutečnostmi, které ovlivňují výsledný profit obchodování, je **spread** neboli rozdíl mezi nákupní a prodejní cenou. Rozdíl je proměnlivý, ale průměr je nutné brát v potaz. Další důležitou hodnotou je swap. **Swap** je rozdíl úrokových sazeb mezi obchodovanými měnami. Při zadávání pozice je nutné zvážit několik možností, které otevření přinese. Pokud pozici otevřeme, pak spekulujeme o cenové hladině v budoucím čase a stanovíme hodnotu ceny, na které pozici uzavřeme, takzvané **take profit**. Také stanovíme rizika, která jsme schopni podstoupit. Hladina ceny,

kteřá určuje zavřeni pozice v pŕípadě, že se hladina ceny pohybuje směrem proti otevřené pozici, se nazývá **stop loss**. Obchodník má generalizovaně dva pŕístupy, jak trh analyzovat 2.1, 2.2. Pokud je v práci zmíněna obchodní pozice, je tím myšlena jedna z dvou variant **Long** a **Short**. Pŕíklad otevřeni pozice je uveden v technické analýze 2.2. V pŕípadě **Long** obchodník spekuluje na růst ceny. Pokud po uzavřeni kontraktu dojde k růstu ceny, pak obchodník nabývá zisku. Pokud dojde k poklesu, pak nabývá ztrát. V pŕípadě **Short** obchodník spekuluje na pokles ceny. Pokud po uzavřeni kontraktu dojde k poklesu ceny, pak obchodník nabývá zisku. Pokud dojde k vzestupu, pak nabývá ztrát. V práci také budou využívány pojmy **Tick**, **Bod**, **Spread**, **Lot**. Tyto i výše popsané pojmy jsou shrnuty v tabulce 2.1.

Otevřeni pozice	Zadání pŕíkazu na burzu o nákupu nebo prodeji neboli spekulaci na růst nebo pokles ceny. Na obrázku 2.1 je pozice otevřena v čase 12:05.
Take profit	Určení ceny, na které pozici ukončíme se ziskem. Na obrázku 2.1 je zelenou barvou vyznačena cena na hladině 1.12998, kde je nastaven take profit
Stop loss	Určení ceny, na které pozici ukončíme ve ztrátě. Červenou barvou je na obrázku 2.1 naopak vyznačena pŕíkladem zvolená hladina 1.12698 jako ukončení ve ztrátě.
Tick	Pŕijmutí aktualizované ceny z Forexu. Časová řada ticků je nejmenší časové rozlišení, které lze získat. Souvislost ticku a grafické reprezentace ceny je podrobněji popsána v kapitole 2.2
Bod	Pohyb na burze je udáván v bodech. Jde o nejmenší možné vyjádření vertikálního pohybu. Napŕíklad pokud bychom chtěli na otevřené pozici z obrázku 2.1 vydělat 200 bodů, tak nastavíme take profit na hladinu 1.12998. Z rozdílu ceny, kde byla pozice otevřena, a hladiny take profitu vidíme, že jde o rozdíl na pátém desetinném místě. Pŕíkladové ceny z obrázku 2.1 jsou vybrány z historie páru EURUSD. Pokud bude v textu udáván jakýkoliv zisk, pak bude uveden v bodech. Bodový zisk je pak možné pŕepočítat na zisk podle objemu otevřené pozice <sup>1</sup> .
Spread	Je náklad spojený s otevřeni pozice. Rozdíl mezi nákupní a prodejní cenou udává velikost spreadu.
Lot	Je standartní jednotkou, která udává objem obchodované pozice. Jeden <i>Lot</i> představuje 100 000 jednotek základní měny. Pro představu napŕíklad pozice na obrázku 2.1 skončí ve ztrátě 100 bodů. Pokud bychom pozici otevřeli s objemem 1 lot, tak by ztráta v dolarech činila 100 dolarů.

Tabulka 2.1: Tabulka shrnuje a vysvětluje klíčové pojmy Forexu. Pojmy zahrnují techniky zadávání obchodní pozice. Definují jednotky a poplatky, které jsou na Forexu nedílnou součástí obchodování.



Obrázek 2.1: Obrázek ukazuje černobílý svíčkový graf, kde vyplněná černá svíčka značí pokles a bílá nevyplněná značí růst. Vytváření svíček je blíže popsán v kapitole 2.2. Na obrázku je graficky znázorněno několik pojmů. Jde o příklad otevření pozice s plánovaným take profitem a stop lossem. Pokud bychom pozici otevřeli s nastaveným stop lossem na pokles o 100 bodů na hladině 1.12698, pak by pozice skončila ve ztrátě -100 dolarů, pokud bychom pozici otevřeli s objemem 1 lot. Pokud by však stop loss zůstal nenastavený a nastavili bychom pouze hladinu pro take profit, tak by pozice byla ukončena v zisku 200 dolarů na hladině 1.12998. Pokud bychom započítali rozdíl mezi nákupní a prodejní cenou 20 bodů, pak by zisk činil 180 dolarů.

## 2.1 Fundamentální analýza

Obchodník hodnotí události a data, které trh přímo i nepřímo ovlivňují. Jde nejen o makroekonomické informace týkající se daného ekonomického sektoru, ale také politické situace. Obchodník sleduje ekonomickou situaci ostatních sektorů a hodnotí jejich případný vliv na samotný trh. V rámci Forexu je nutné sledovat ekonomiky států, kterých se týká obchodovaný pár.

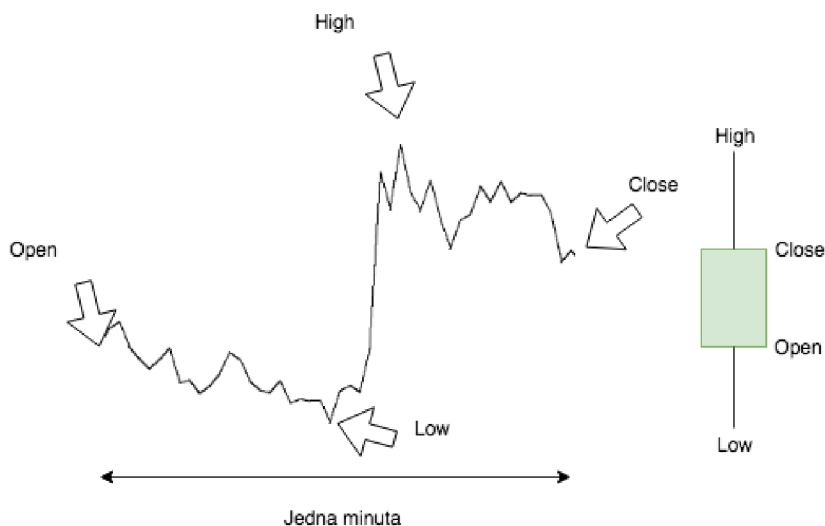
Vhodným příkladem obecné fundamentální analýzy je sledování otevírání světových burz. V těchto časech je možné uvažovat o vstupu do pozice. Také existují forexové kalendáře, které seskupují ekonomická data relativně k měnám [13].

## 2.2 Technická analýza

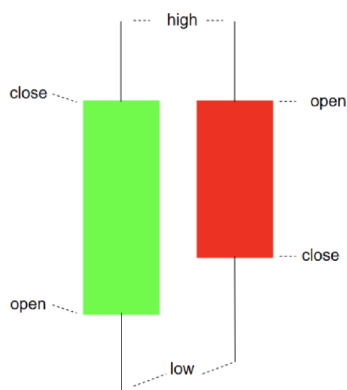
Obchodník pracuje s daty přímo odvozenými z historické a aktuální ceny. Jde tedy o data tvořená přímo trhem. Je zřejmno mnoho technických indikátorů, které transformují události na trhu a indikují možné situace. Popis a výčet technických indikátorů použitých v této práci je uveden ve výsledné architektuře 4.

Pouze na základě technické analýzy je možné vytvořit obchodní strategie. Pro představu, jak může vypadat a být využita, uvedu příklad. Samotné zobrazení dat je základem technické analýzy. Nejčastěji používané jsou dva typy grafů zobrazení uzavírací ceny a graf svíčkový. Historická kurzovní data můžeme zobrazit v různém časovém rozlišení. V případě

grafu s uzavírací cenou je zobrazena pouze cena v daném časovém intervalu. Svíčkový graf zobrazuje i informaci o ceně v mezičase. Svíčku tvoří čtyři hodnoty *open*, *high*, *low*, *close*. Na obrázku 2.3 jsou vyznačené dvě možné varianty. Svíčka se zeleným tělem primárně vyznačuje růst. Konkrétně určíme, že daná svíčka zobrazuje například interval 1 minuta. Na začátku cena začne na hladině *open*. V průběhu patnácti minut potom cena dosáhne hodnoty *high* a *low*. Na konci intervalu se svíčka uzavře na hladině *close*. Obdobně je tomu u svíčky s červeným tělem, která vyznačuje pokles.



Obrázek 2.2: V pravé části je pro příklad uveden interval trvající jednu minutu. V intervalu je zaznamenán vývoj ceny v podobě ticků. V případě timeframeu jedné minuty by byla svíčka vytvořena cenovým vývojem uvedeným na levé straně



Obrázek 2.3: Levá svíčka ukazuje cenový růst oproti svíčce pravé, která má *close* pod *open* a značí pokles

Svíčkové formace jsou jednoduchým zpracováním ceny. Většina indikátorů je počítána nad hodnotami *open*, *high*, *low* a *close*. Pro příklad obchodní strategie jsem vybral spojení klouzavého průměru a indikátoru RSI.

Klouzavý průměr je základním indikátorem<sup>2</sup>, počítá průměr svíček přes interval. Většinou je počítán přes hodnotu close. Klouzavý průměr  $MA$  a exponenciální klouzavý průměr  $EMA$  jsou počítány následovně:

$$MA_i(n) = \frac{1}{n} \sum_{i=1}^n P_{n-i} \quad (2.1)$$

$$EMA_i(n) = (P_i - EMA_{i-1}) \frac{2}{n+1} + EMA_{i-1} \quad (2.2)$$

Index relativní síly se neustále pohybuje v rozmezí 0 - 100. RSI uvádím zejména kvůli jeho širokému využití. Jeho stav je reprezentací síly trhu, indikuje případné překoupení trhu a možnost uzavření pozice. Uvádím ho pro vlastnosti výpočtu<sup>3</sup>, který je uvedený níže.

$$RSI_t(n) = 100 - \frac{100}{1 + \frac{U(n)}{D(n)}} \quad (2.3)$$

$U(n)$  součet kladných cenových změn za období délky  $n$

$D(n)$  součet záporných cenových změn za období délky  $n$

Strategie spojující tyto indikátory je graficky znázorněna na obrázku 2.2 a mohla by vypadat následovně: klouzavé průměry počítané přes rozdílné období budou indikovat možné otevření pozice. Křivka modré barvy vyznačuje  $EMA$  počítané přes 18 uzavřených cen close. Červeně značená křivka je klasický  $MA$  počítaný přes 80 uzavřených cen close. V dolní části je zobrazený indikátor  $RSI$  s kritickými hladinami 70 % a 30 %. Překřížení  $EMA$  směrem dolů přes  $MA$  určí zadání pozice short. Situace překřížení je označena modrou vertikální čarou v čase 16:00. Při vstupu do pozice zajistíme obchod pomocí *stop loss* a určíme, kdy pozici ukončit. Ukončení může být stanoveno pevně. Já jsem pro ukončení pozice zakomponoval indikátor  $RSI$ . Konkrétně pozice short bude ukončena, pokud  $RSI$  překoná dolních 30 % a opět je vrátí. Modrá vertikální čára v čase ukončení 18:00 ukazuje návrat nad hodnotu 30 %.

<sup>2</sup>Více informací je možné zjistit zde: [https://en.wikipedia.org/wiki/Moving\\_average](https://en.wikipedia.org/wiki/Moving_average)

<sup>3</sup>Více informací je možné zjistit zde: [https://en.wikipedia.org/wiki/Relative\\_strength\\_index](https://en.wikipedia.org/wiki/Relative_strength_index)

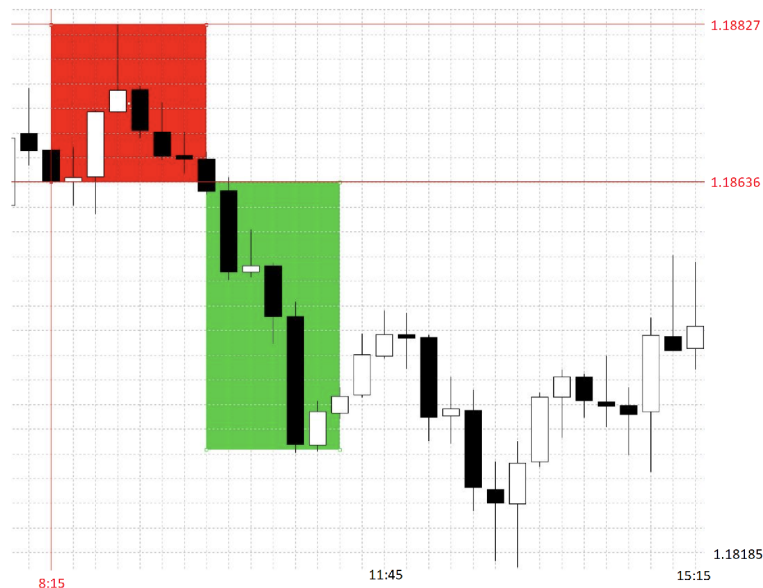


Obrázek 2.4: Na obrázku je ukázána strategie na určitém časové intervalu využívající 3 technické indikátory ( $MA$ ,  $EMA$ ,  $RSI$ ). Křivka modré barvy vyznačuje  $EMA$  počítané přes 18 uzavřených cen close. Červená křivka je klasický  $MA$  počítaný přes 80 uzavřených cen close. Klouzavé průměry indikují vstup do pozice. V dolní části je zobrazený indikátor  $RSI$  s kritickými hladinami 70 % a 30 %. Překřížení v čase 16:00 označené vertikální modrou barvou značí pozici short. Pozice short bude ukončena, pokud  $RSI$  překoná dolních 30% a opět je vrátí. Modrá vertikální čára v čase ukončení 18:00 ukazuje návrat nad hodnotu 30%. Jde o příklad toho, jak mohou být indikátory využity pro obchodní strategie

## 2.3 Závěr

Systém v pozici tradera bude podléhat výše uvedeným principům trhu, a tak je třeba do návrhu zahrnout řízení rizik spojené s finanční pákou a volnou jistinou. Využití vysoké finanční páky bude možné pouze za předpokladu otevření pozice s předpokladem rychlé realizace a důkladnou analýzou o budoucím vývoji trhu nejen v čase předpokládaného ukončení. Uvedu příklad pro demonstraci situace, která by byla kritická při vysoké finanční páce a velkém objemu pozice. Samozřejmě všechna negativa reálného nasazení nebudu schopný simulovat obchodním systémem uvedeným v práci. To by překračovalo časové možnosti práce. Ale v kapitole 3.5 jsem uvedl vhodnou metodu pro co nejpřísnější testování.

Na obrázku 2.5 jsou označeny oblasti, kde pomyslně mohlo dojít ke vstupu do pozice short, tedy spekulace na pokles ceny. V čase 8:15 označeném začátkem červené oblasti předpokládáme, že v následujících přibližně tří hodin dojde k poklesu. Pokud bychom otevřeli pozici o objemu, který spotřebuje veškerou jistinu, budeme v čase červené zóny nuceni jistinu doplnit. Pokud tento krok neuděláme, naše pozice bude uzavřena ve ztrátě i přes to, že v čase zóny označené zeleně by naše predikce uspěla.



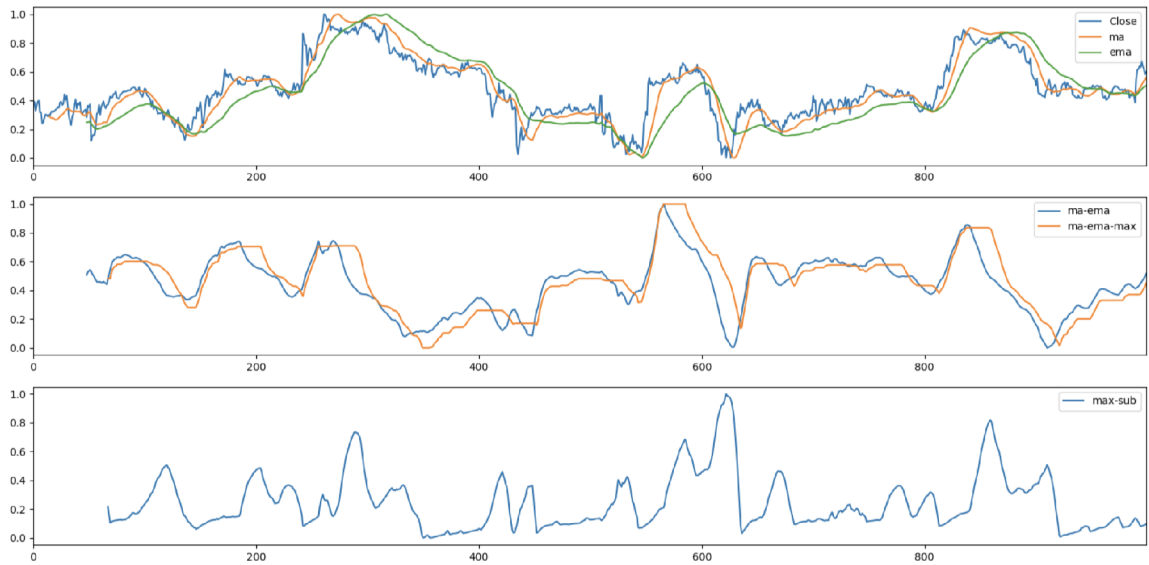
Obrázek 2.5: Na obrázku jsou označeny oblasti, kde pomyslně mohlo dojít ke vstupu do pozice short, tedy spekulace na pokles ceny. V čase 8:15 označeném začátkem červené oblasti předpokládáme, že v následujících přibližně tří hodin dojde k poklesu. Pokud bychom otevřeli pozici o objemu, který spotřebuje veškerou jistinu, budeme v čase červené zóny nuceni jistinu doplnit. Pokud tento krok neuděláme, naše pozice bude uzavřena ve ztrátě i přes to, že v čase zóny označené zeleně by naše predikce uspěla

Systém bude vhodné vytvořit primárně pro technickou analýzu. Je nespočet možností, jak indikátory využít. Navržených strategií je mnoho a všechny mají za jistých podmínek svůj potenciál. Od automatického obchodního systému požadují, aby se pokusil vytvořit **různé kombinace** a různé další přepočty nad množinou technických indikátorů. Podle charakteru dat a modelů, které se snaží finanční data popsat, je zřejmé, že různé kombinace a výpočty mohou fungovat pouze určitý časový úsek.

Pro představu, jak **bude přepočet fungovat**, uvedu krátký příklad, který jsem vytvořil v začátcích implementace systému jako základ překladače 4.4. Na vstupu bude číselná řada, konkrétně časová řada s hodnotami *close*. Nad touto řadou se pokusí systém provést náhodné výpočty za účelem nalezení výpočtu vhodného pro ziskovou strategii.

Na obrázku 2.6 jsou zobrazeny signály, které byly odvozeny z hodnot *close*. Systém nejdříve provedl výpočet klouzavých průměrů *ma* a *ema*, které jsou zobrazeny v horní části spolu s hodnotami *close*. V dalším kroku provedl odečtení  $ma - ema$  a následně nad hodnotami z předchozího kroku aplikoval okenní funkci pro nalezení maxima. Obě časové řady jsem opět zobrazil společně v druhém okně. Poslední operací bylo odečtení časových řad z druhého okna  $(ma - ema - max) - (ma - ema)$ . Takto nalezený výpočet může participovat na výsledné strategii. V podstatě takto provedený výpočet může být nalezen náhodnou aplikací operací nad vstupními časovými řadami *Open*, *High*, *Low*, *Close*. **Tento princip** aplikace výpočtů a získání vypočítané časové řady je klíčový pro implementovaný systém.





Obrázek 2.6: Obrázek ukazuje možný postup operací nad vstupní časovou řadou jako ukázka postupu výpočtu obchodní strategie. Systém provedl výpočet klouzavých průměrů *ma* a *ema*. Oba klouzavé průměry jsou zobrazeny v prvním okně spolu s časovou řadou *close*. Dalším výpočetním krokem je odečtení klouzavých průměrů a aplikace okenní funkce pro získání maxima. Nové časové řady odečtení klouzavých průměrů a jejich maximum jsou zobrazeny v druhém okně. V posledním okně je konečná operace odečtení časových řad z druhého okna  $(ma - ema - max) - (ma - ema)$ . Časová řada v posledním okně může být využita jako indikátor. Na základě překročení určité hranice může být indikován vstup do pozice

## Kapitola 3

# Prostředky, techniky a poznatky pro návrh systému

I přesto, že kapitola o Forexu je stěžejní, neboť popisuje prostředí, pro které celý systém vyvíjíme, nutné hledat prostředky, na kterých bude systém vytvořen. V úvodu bylo řečeno, že je možné využít mnoho různých přístupů. Tato kapitola by mohla jít napříč obory a shrnovat metody pro analýzu trhu, signálů, chaosu a mnoho dalších. Vybral jsem však zlomek, který zasahuje do oborů strojového učení a technické analýzy. Kapitola by měla čtenáři ukázat rozmanitost, s jakou je možné přistupovat k analýze forexového signálu, jak bylo řečeno v úvodu.

Zásadním podkladem pro tuto práci je kniha [5] publikující techniky pro generování popisu signálu na základě gramatik a gramatické evoluce. Uvádí myšlenku vytvořit systém na základě řetězců, které popisují signál forexu. Řetězec popisující signál například za účelem obchodních příležitostí může zahrnovat postup s využitím klasických matematických operací nebo i implementovaných heuristik, jako například strojového učení nebo technických indikátorů. Kapitola 3.1 tedy popisuje princip generování popisu. Kapitola 3.2 obsahuje popis optimalizace, kterou využijeme pro optimalizaci vytvořených řetězců. Vzhledem k několika publikacím, které hovoří o aplikaci neuronových sítí pro zpracování signálu, jsem se rozhodl prostudovat i problematiku neuronových sítí pro jejich zanesení do systému. Kapitola 3.4 shrnuje vlastnosti finančních dat a další prostředky pro jejich popis. Konkrétně jsem uvedl lineární modely pro jejich popis a model pro popis volatility za účelem nahlédnutí do principů těchto modelů pro případné využití jejich vlastností.

### 3.1 Generování popisu signálu na základě gramatiky

V knize [5] je popsána metoda, jak generovat popis časových řad pomocí gramatik. Následně je využívána i gramatická evoluce. Obecně lze tento přístup využít ke generování řetězců, které reprezentují program. Uvádí zápis jedince jako číselný kód: [166|225|180|132|187|219|179|249], který je následně mapován na derivované operace. Přesněji, podle čísel jsou vybrány pravidla. V knize je uveden jasný příklad.

Operátor modulo slouží k určení neterminálu. Tedy pro příklad 166%2 je vybrána první možnost startovacího neterminálu. Dále se pokračuje derivováním neterminálů zleva doprava. V knize se mluví o využití genetických algoritmů nad tímto principem a také je upozorněno na problémy, které tento přístup přináší. Jako například jedinci, kteří nemají

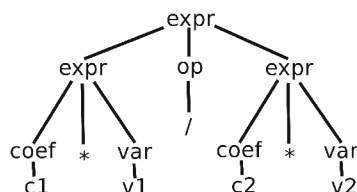
Neterminály = $\text{expr}, \text{op}, \text{coef}, \text{var}$	$\langle \text{expr} \rangle$ je počáteční
Terminály = $\div, \times, +, -, V1, V2, C1, V2, (, )$	
Pravidla:	
$\langle \text{expr} \rangle$	$: (\langle \text{expr} \rangle \langle \text{op} \rangle \langle \text{expr} \rangle)$ (1.a)
	$  (\langle \text{coef} \rangle \times \langle \text{var} \rangle)$ (1.b)
$\langle \text{op} \rangle$	$: \div   \times   +   -$ (2.a), (2.b), (2.c), (2.d)
$\langle \text{coef} \rangle$	$: C1   C2$ (3.a), (3.b)
$\langle \text{var} \rangle$	$: V1   V2$ (4.a), (4.b)

Tabulka 3.1: Tabulka s gramatikou, kde jsou neterminály, terminály a pravidla pro derivace

Modulo	Pravidlo	Stav derivace
166 MOD 2	0	$(\langle \text{expr} \rangle) \langle \text{op} \rangle (\langle \text{expr} \rangle)$
225 MOD 2	1	$(\langle \text{coef} \rangle \langle \text{var} \rangle) \langle \text{op} \rangle (\langle \text{expr} \rangle)$
180 MOD 2	0	$(c1 \times \langle \text{var} \rangle) \langle \text{op} \rangle (\langle \text{expr} \rangle)$
132 MOD 2	0	$(c1 \times v1) \langle \text{op} \rangle (\langle \text{expr} \rangle)$
187 MOD 2	3	$(c1 \times v1) \div (\langle \text{expr} \rangle)$
219 MOD 2	1	$(c1 \times v1) \div (\langle \text{coef} \rangle \times \langle \text{var} \rangle)$
179 MOD 2	1	$(c1 \times v1) \div (c2 \times \langle \text{var} \rangle)$
249 MOD 2	1	$(c1 \times v1) \div (c2 \times v2)$

Tabulka 3.2: Tabulka s překlady výsledků operace modulo na výběr pravidla

dokončené derivování, ale přímé řešení popsáno není. Uvedené kroky v tabulce lze zobrazit jako derivační strom, který je ukázaný na obrázku :



Obrázek 3.1: Derivační strom ukazuje derivace z neterminálů na terminály až do derivace věty jako listy stromu. Obrázek doprovází tabulku 3.2, tedy příklad generování matematického popisu časových řad za pomoci gramatik z knihy [5]

V knize [5] se dále píše o možnostech využití technických indikátorů a genetických algoritmech. Informace z knihy měly největší dopad na výslednou architekturu

## 3.2 Genetické algoritmy

Optimalizace je jedním z přístupů, které systém využívá. Za předpokladu, že systém zpracovává stochastické signály, které nelze popsat deterministicky, je vhodné při modelování využít principů optimalizace.

Optimalizace je proces nad systémem, který vede k co nejlepším výsledkům podle daného kritéria. Optimalizaci je možné uvažovat na několika úrovních. Pojetí této kapitoly je zaměřeno na optimalizaci z hlediska matematiky, kdy jde o proces nad parametry

funkce, tak aby funkce nabývala maximálních, případně minimálních hodnot. Tedy pro funkci  $A \rightarrow \mathbb{R}$  v případě maximalizace hledáme  $x_h \iff A$ , kde  $f(x_h) \geq f(x)$  a  $x \iff A$ .

Na základě studie několika vybraných algoritmů jsem se rozhodl použít genetické algoritmy. V úvahu jsem bral Simulované žíhání, který využívá heuristiku procesu žíhání, kdy se látka zahřeje a atomy se vyváží z krystalické mřížky. Při chlazení se atomy vážou v krystalickém uspořádání s co nejmenším výdejem energie. Algoritmus tak vnáší možnost vymanit se z lokálního minima. Symbolický parametr teploty je pravděpodobnost, že krok bude přidán mezi plánované kroky.

Genetické algoritmy jsem vybral zejména díky architektuře celého systému a vlastnostem celé optimalizace, které jsou popsány dále. Na základě heuristiky odvozené z evolučních principů algoritmus postupně vylepšuje parametry systému. Napodobuje procesy, na kterých je evoluce založena. Algoritmus postupuje v generacích, které obsahují různá řešení. Každé řešení v generaci je ohodnoceno fitness funkcí, která reprezentuje přirozený výběr. Fitness funkce udává kvalitu řešení v generaci neboli kvalitu jedince. V další generaci jsou vybráni nejlepší jedinci a nad nimi je aplikováno křížení a mutace, takže takto vznikne generace nová.

Algoritmus je možné zapsat v krocích:

1. Inicializuj první populaci z náhodných jedinců.
2. V cyklu pomocí hodnotící funkce vyber několik nejlépe ohodnocených jedinců.
3. Z vybraných jedinců vytvoř novou populaci za pomoci operátorů pro křížení, mutaci a reprodukci. Křížení náhodně vymění části jedinců mezi sebou. Mutace náhodně změní část jedince. Reprodukce jedince beze změny zkopíruje.
4. Vypočti ohodnocení nových jedinců.
5. Konec cyklu, který kontroluje ukončovací podmínku.
6. Řešení reprezentuje jedinec z poslední populace s nejvyšším ohodnocením.

Výše popsaný postup algoritmu jsem vyvodil z knihy [9]. V literatuře [20] jsou nejčastěji jedinci uváděni jako binární kombinace o délce  $N$  a operace nad nimi také. Proto dále rozvedu myšlenku, jak budu s tímto faktem pracovat a zda zachovám vlastnosti algoritmu [3.2.2](#).

1. Jednobodová mutace: je zvolen náhodně index  $i$  s podmínkou  $1 \leq i \leq N$ , kde  $N$  je délka jedince a nejnižší index je 1. Na daném indexu  $i$  je změněna hodnota. Například opět pro index  $i = 3$ :

jedinec	010111
jedinec po mutaci	011111

2. Jednobodové křížení: je zvolen náhodně index  $i$  s podmínkou  $1 \leq i \leq N$ , kde  $N$  je délka jedince a nejnižší index je 1. Nový potomek je pak složen z části obou rodičů. Podle indexu  $i$  je vybráno z rodiče A prvních  $X$  genů s indexem  $0 \leq x \leq i$  a z rodiče B je vybráno zbylých  $Y$  genů s indexem  $i + 1 \leq x \leq N$ . Možné křížení dvou jedinců by mohlo vypadat takto, pokud by  $i = 3$ :

rodič A	010111
rodič B	111000
potomek	010000

3. Přirozený výběr: selekce může být realizována mnoha způsoby za podmínek, že jedinci s lepším ohodnocením budou s větší pravděpodobností reprodukováni a páry křížení budou vybrány náhodně. Dále představím konkrétní algoritmus turnaje, který je přímo využíván v systému

### 3.2.1 Gramatická evoluce

Vzhledem k systému zmíním z oblasti genetických algoritmů gramatickou evoluci. Pro vyřešení problému je stanovena bezkontextová gramatika. Gramatika  $G = (N, \Sigma, P, S)$  je bezkontextovou gramatikou, pokud pravidla z množiny  $P$  mají tvar:

$$A \rightarrow \alpha; A \exists N; \alpha \exists (N \text{ sup } \Sigma)^*$$

Gramatická evoluce využívá překladač jazyka generovaného gramatikou. Musí pracovat s proměnlivou délkou jedinců. Pravidla gramatiky jsou zapsána ve tvaru:

$$\langle symbol \rangle := \langle derivace \rangle$$

$\langle symbol \rangle$  náleží do množiny neterminálů a  $\langle derivace \rangle$  je posloupnost jednotlivě oddělených terminálů a neterminálů. Gramatika tak slouží ke generování syntakticky správné posloupnosti. Genotypem je posloupnost, která určuje pořadí jednotlivých operací.

### 3.2.2 Aplikace genetických algoritmů

Pro účely systému je nutné principy genetických algoritmů vhodně uchopit. Vzhledem k celkové architektuře je nutné uvažovat, jak celkově budou pojaty pojmy jedinec, generace, křížení, mutace a jak bude navržena fitness funkce. Nad časovými řadami budou počítány uvedené funkce a technické indikátory. Půjde o posloupnost matematických operací s následným ohodnocením fitness funkcí, zda daná posloupnost vedla k očekávaným výsledkům. Jedinec bude reprezentován řetězcem a operace genetických algoritmů jako operace nad řetězcem.

## 3.3 Neuronové sítě

Tato podkapitola čerpala z knihy [1]. Model umělé neuronové sítě vznikl již v padesátých letech, kdy se výzkumníci snažili o simulaci biologického neuronu. Pokud bych měl tento model uvést z hlediska praktického nasazení, tak bych začal jeho schopností obecné regrese funkce. Schopností nastavit parametry tohoto modelu, tak aby simulovaly funkci, kterou žádáme. Nastavení parametrů je iterativním procesem, kdy neuronové síti předkládáme předem zpracovaná data, která model natrénují. Nasazení neuronové sítě obnáší zpracování dat do požadované podoby, návrh samotného modelu sítě, jeho natrénování a následné nasazení na reálných neviděných datech.

Model je tvořen z  $n$  vrstev, kde jednu vrstvu definujeme jako lineární kombinaci vstupů  $x_1, \dots, x_m$ :

$$a_j = \sum_{i=1}^D w_{ji}^1 x_i + w_{j0}^1 \quad (3.1)$$

Horní index říká, že jde o první vrstvu neuronové sítě. Index  $j$  udává počet neuronů v dané vrstvě. Následně je hodnota  $a_j$  transformována aktivační funkcí  $h()$ . Aktivačních funkcí je více, podmínkou je jejich možnost derivace.

$$z_j = h(a_j) \quad (3.2)$$

Výsledná hodnota je vstupem do skryté vrstvy neuronové sítě:

$$a_k = \sum_{j=1}^D w_{kj}^2 z_j + w_{k0}^2 \quad (3.3)$$

Jednotlivé vrstvy je možné propojit do vícevrstvého modelu se vstupní vrstvou, vrstvami skrytými a vrstvou výstupní. Neuronová síť je tedy nelineární funkcí se vstupním a výstupním vektorem a vektorem vah, který je nastavitelný.

Nejčastěji používané aktivační funkce jsou logistická sigmoida:

$$y = \frac{1}{1 + e^{-x}} \quad (3.4)$$

a funkce *tanh*:

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.5)$$

Využití aktivační funkce se odvíjí od charakteru dat a od použité normalizace. Obor hodnot funkce logistická sigmoida je v intervalu 0 až 1, pro funkci *tanh* je to -1 až 1. Je tedy třeba myslet na to, zda jsou naše data v tomto rozsahu a také zda splňují pravděpodobnostní rozložení modelované aktivační funkcí. Tyto dvě funkce jsou vhodné vzhledem ke své snadné derivaci. Výpočet zpětné propagace je relativně snadný.

### 3.3.1 Trénování neuronové sítě

Trénování je iterativní proces, ve kterém je žádoucí minimalizovat funkci  $E(w)$ . Vstupem funkce je vektor vah. Pro trénování sestavíme datovou sadu jako vektor vstupů  $x$  a vektor referenčního výstupu jako target  $t$  o velikosti  $N$ . Výstupní data jsou nezávislá a v podmíněném rozložení:

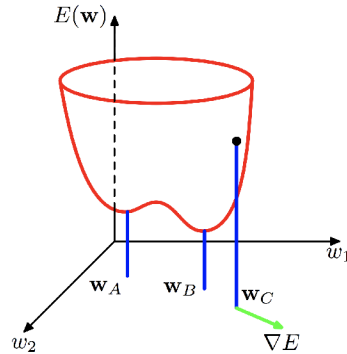
$$(t|x, w) = \mathcal{N}(t|y(x, w), \beta^{-1}) \quad (3.6)$$

kde  $\beta^{-1}$  je přesnost gausovského šumu. Můžeme říct, že hledáme nejvěrohodnější vektor vah  $w$ . Příkladem hodnotící funkce je například nejmenší součet čtverců:

$$\frac{1}{\beta_{ml}} = \frac{1}{NK} \sum_{i=1}^N |y(x_i, w_{ml}) - t_i|^2 \quad (3.7)$$

Pro optimalizaci vah je široce používaný algoritmus Gradient descent. Jde o numerickou metodu, která aktualizuje vektor vah ve směru největšího gradientu. Tato metoda má parametr learning rate, který určuje, o jak velký krok ve směru gradientu půjde. Velikost

parametru learning rate je důležité uvážit. Podívejme se na obrázek 3.2. Obrázek ukazuje chybový prostor s lokálním minimem a minimem globálním jako  $W_B$ . Pokud bychom learning rate zvolili příliš velký pak bychom i přes krátkou vzdálenost globální minimum přeskočili.



Obrázek 3.2: Chybový prostor [1]

### 3.3.2 Zpětná propagace chyby

Uvedu zpětnou propagaci chyby pomocí evaluace Jakobyho matice. Tento algoritmus bude vysvětlen pomocí jednoduchého vícevrstvého modelu založeného na učení z gradientu. Vrstvy jsou definovány funkcí  $F_n(W_n, X_{n-1})$ .  $X_n$  je vstupem vrstvy a  $X_{n-1}$  je výstupem.  $W$  je vektorem proměnlivých vah a  $T$  je vektorem dat výstupních. Je známa parciální derivace funkce  $E$  podle proměnné  $X_n$ . A pokud lze spočítat parciální derivaci funkce  $E$  podle  $W_n$  pomocí diferenčních rovnic, pak je možné spočítat  $X_{n-1}$ :

$$\frac{\partial E}{\partial W_n} = \frac{\partial F_n}{\partial W}(W_n, X_{n-1}) \frac{\partial E}{\partial X_n} \quad (3.8)$$

$$\frac{\partial E}{\partial W_{n-1}} = \frac{\partial F_n}{\partial X}(W_n, X_{n-1}) \frac{\partial E}{\partial X_n} \quad (3.9)$$

kde  $\frac{\partial F_n}{\partial W}$  je jakobian  $W$  podle proměnné  $W$  v bodě  $(W_n, X_{n-1})$  a  $\frac{\partial F_n}{\partial X}$  je jakobian podle proměnné  $X$ . Jakobian vektorové funkce obsahuje parciální derivace všech výstupů s ohledem ke všem vstupům. Reverzním výpočtem pro každou vrstvu spočítáme všechny parciální derivace vzhledem ke všem parametrům. Tento postup je uveden podle knihy [26].

### 3.3.3 Promíchání dat

Promíchání vstupních dat je důležitým krokem v knize [26]. Z charakteru zpětné propagace vyplývá, že k chtěnému učení dojde nejlépe, pokud je propagována velká chyba. Pokud bychom měli podobná data seskupená, pak by byla výstupní chyba vzhledem k seskupení malá v porovnání se sekvencí náhodně vytvořenou napříč datovou sadou.

### 3.3.4 Normalizace dat

Trénování je vhodné provádět nad daty, která mají nulový průměr. Pokud jsou data posunuta jedním směrem, pak je proces trénování pozdržen. Všechny váhy jsou ovlivněny jedním směrem a trénování je pozdrženo, neboť je upravován pouze bias směrem k nulovému průměru.

Dalším popsaným krokem je i transformace covariance dat, tak aby byla rovna jedné v případě výše popsaných aktivačních funkcí. Normalizaci je možné provést například se zachováním variance:

$$X_i = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (3.10)$$

nebo je možné použít normalizaci, která převede data tak, že mají nulový průměr a jednotkovou covarianci:

$$X_i = \frac{X_i - \mu}{\sigma} \quad (3.11)$$

### 3.3.5 Redukce dimenze

Využívanou transformací je metoda PCA (Principal Component Analysis). Metoda určí lineární transformaci ortogonálních bází. Její aplikací jsou data dekorelována a je snížena jejich dimenzionalita. V transformovaných datech je zachována největší variabilita. Vlastní vektory kovarianční matice určují bázové vektory PCA. Matice je počítána takto [18]:

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - m)(x_i - m)^T \quad (3.12)$$

kde  $C$  značí kovarianční matici. Matice je počítána na základě  $N$  vstupních trénovacích vektorů. Dále  $m$  je odhadovaný střední vektor a  $x_i$   $i$ -tý trénovací vektor:

$$m = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.13)$$

Vlastní vektory, které odpovídají největším vlastním hodnotám, jsou použity jako báze pro účel redukce. Vlastní hodnota s odpovídajícím vlastním vektorem je určena množstvím variability získané projekcí vstupního vektoru do vektoru vlastního. Pokud tuto transformaci využijeme, je nutné myslet na fakt, že není možné transformovat testovací data do nově vypočítaného prostoru, ale je nutné využít parametry pro transformaci dat trénovacích.

## 3.4 Finanční kalkul

Kapitola zhodnocuje některé finanční modely. Uvozuje pravděpodobnostní rozložení, které se týká finančních dat. Také celkově popisuje charakter dat a vysvětluje v jakém formátu je vhodné uvádět změnu ceny. V knize [25] tento přepočítání zmíněn spíše ve smyslu reprezentace zhodnocení investice.

Účelem je uvést základní pohled na finanční matematiku a vyvodit poznatky pro cíl této práce 1.2. Pro hlubší pochopení je nutné nastudovat kontext dále uvedených informací [25].

### 3.4.1 Reprezentace cenové změny

v knize [25] popisují jednoduchou reprezentaci změny ceny z času  $t - 1$  na  $t$  jako poměr dvou cen. Cenu v daném čase značí  $P_t$ , kde  $t$  je časový index.

$$1 + R_t = \frac{P_t}{P_{t-1}}$$



$$R_t = \frac{P_t}{P_t - P_{t-1}}$$

V rozšíření pro  $k$  změn z času  $t - k$  na  $t$ . Je změna reprezentována

$$1 + R_t[k] = \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \times \frac{P_{t-1}}{P_{t-2}} \times \dots \times \frac{P_{t-k+1}}{P_{t-k}} = (1 + R_t)(1 + R_{t-1}) \dots (1 + R_{t-k+1})$$

Celková změna pro periodu o délce  $k$  může být zapsána jako:

$$1 + R_t[k] = \frac{P_t}{P_{t-k}}$$

V knize [25] je pak dále vysvětlený vliv úroku na dané aktivum. Tento princip je založen na přirozeném růstu a pokud chceme vyjádřit změnu ceny s ohledem na tento fakt. Musíme poměr změny logaritmovat:

$$\ln(1 + R_t) = \ln\left(\frac{P_t}{P_t - P_{t-1}}\right)$$

Dělení je vzhledem k logaritmu možné převést na odčítání:

$$R_t[k] = P_t - P_{t-1}$$

### 3.4.2 Vlastnosti rozložení pravděpodobnosti finančních dat

Pro základní pochopení finančních časových řad a utvoření obecné představy o vlastnostech finančních dat uvedu jejich vlastnosti pravděpodobnostního rozložení.

Budu mluvit o spojitě, marginální a podmíněné pravděpodobnosti. Mějme  $n$ -dimensionální Euklidovský prostor  $R^n$ , kde bod  $x \in R^n$ . Uvažujme vektory  $X = (X_1, \dots, X_i)$  a  $Y = (Y_1, \dots, Y_i)$  a  $P(X \in A, Y \in B)$  je pravděpodobností, že  $X$  je v podprostoru  $A \in R^n$  a  $Y$  je v podprostoru  $B \in R^n$ .

**Spojitá pravděpodobnost** je popsána funkcí  $F(x, y, \Theta)$ . funkce charakterizuje vlastnosti  $X$  a  $Y$ . Kde  $x \in R^n$  a  $y \in R^k$ :

$$F(x, y, \Theta) = P(X \leq x, Y \leq y; \Theta)$$

Pokud jde o spojitě hodnoty a funkce hustoty spojitě pravděpodobnosti  $f_{x,y}(x, y, \Theta)$  existuje pro  $X$  a  $Y$ :

$$F(x, y, \Theta) = \int_{-\inf}^x \int_{-\inf}^y f_{x,y}(w, z, \Theta) dz dw$$

**Marginální pravděpodobnost** je dána funkcí pro  $X$ :

$$F_X(x, \Theta) = F_{X,Y}(x, \inf, \dots, \inf; \Theta)$$

Pak můžeme získat marginální pravděpodobnost pouze pro  $X$ . Stejně tak to může být získána pro  $Y$ .

**Podmíněné rozdělení pravděpodobnosti** pro veličinu  $X$  podmíněnou  $y$  je pravděpodobnostní rozdělení  $X$  za podmínky, že náhodná veličina  $Y$  nabyde určené hodnoty  $y$ . Podmíněné rozdělení je dáno podílem rozdělení spojitého a marginálního. Tedy pro  $X$  za podmínky  $Y \leq y$  je dáno:

$$F_{X|Y \leq y}(x, \Theta) = \frac{P(X \leq x, Y \leq y; \Theta)}{P(Y \leq y)}$$

Podmíněná hustota pravděpodobnosti je potom:

$$f_{X|Y}(x, \Theta) = \frac{f_{x,y}(x, y; \Theta)}{f_y(y; \Theta)}$$

Stěžejní je potom vztah těchto tří pravděpodobnostních rozdělení:

$$f_{x|y}(x, y, \Theta) = f_{x|y}(x; \Theta) \times f_y(y; \Theta)$$

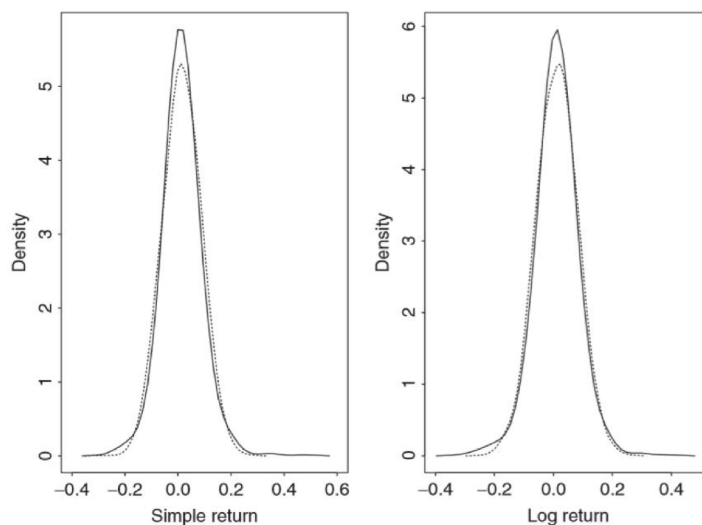
Náhodné vektory  $X$  a  $Y$  jsou nezávislé právě tehdy, když:

$$f_{x|y}(x, y, \Theta) = f_x(x, y, \Theta)$$

Dalším důležitým termínem je **moment náhodné promenné** [25]. Prvními dvěma momenty jsou variance a směrodatná odchylka, které popisují normální rozložení. Jsou i další momenty pro popis dalších pravděpodobnostních rozložení. Třetí centrální moment určuje symetrii vzhledem k průměru a čtvrtý centrální moment udává míru s jakou jsou data vychýlena.

Čtvrtý centrální moment by měl být pro přesné normální rozložení nulový. Pokud nabývá pozitivních hodnot pak značí takzvaně heavy tails rozložení pravděpodobnosti. Z praktického hlediska to znamená, že datová množina obsahuje silné výkyvy. V knize [25] jsou také zveřejněny testy na základě těchto momentů, kde vychází, že finanční data mají silné heavy tails rozložení. Také je z textu zřejmé, že podmíněné rozložení pravděpodobnosti může být podmíněno časem  $F(r_{it}|(r_{i,t-1}))$ . Jedna z teorií založených na modelu náhodné procházky říká, že podmíněná pravděpodobnost časem  $F(r_{it}|(r_{i,t-1}, \dots, (r_{i,1}))$  je rovna marginální pravděpodobnosti  $F(r_{it})$ . V tomto případě jsou data kontinuálně nezávislá a nepredikovatelná.

Závěrem je přiložen obrázek, který ukazuje výše popsané z praktického hlediska. V knize [25] zpracovali cenová data akcií firmy IBM a zobrazili rozložení pravděpodobnosti normálního rozdílu a logatirmovaného rozdílu cen. Zobrazeny jsou měsíční rozdíly od roku 1926 až 2008. Tečkami je zobrazeno odpovídající normální rozložení. Je tedy vidět, že reálné rozložení je mírně jiné. Také je vidět je logaritmovaný rozdíl cen nemá takové výkyvy hodnot.



Obrázek 3.3: Ukázka rozložení pravděpodobnosti měsíčních rozdílů cen akcií firmy IBM od roku 1926 až 2008

Dále jsem se rozhodl zmínit některé informace ze zdroje [23]. Pokud bychom uvažovali pravděpodobnostní rozložení ceny samotné. Bude spadat do rozložení exponenciálního. Centrální limitní věta<sup>1</sup> není splněna u tohoto typu rozložení. Normální rozložení splňuje podmínku centrální limitní věty. Nechť  $\bar{X}$  je průměr náhodného výběru z rozdělení se střední hodnotou  $\mu$  a rozptylem  $\sigma > 0$ . Pak

$$W = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} \quad (3.14)$$

má rozdělení  $N(0, 1)$  pro  $n \Rightarrow \infty$ .

**Aplikace:** Pokud je  $n$  dost velké, pak:

$$\begin{aligned} W &= \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} = \frac{\frac{1}{n} \sum \bar{X} - \mu}{\frac{\sigma}{\sqrt[2]{n}}} = \\ &= \frac{\frac{n}{n} \sum \bar{X} - \mu * n}{n * \frac{\sigma}{\sqrt[2]{n}}} = \frac{\sum \bar{X}_i - \mu * n}{\sqrt[2]{n} * \sigma} \end{aligned} \quad (3.15)$$

má přibližně standardní normální rozložení  $N(0, 1)$ .

Centrální limitní věta říká, že pokud náhodná veličina  $V$  z dané množiny může být vyjádřena jako součet náhodných proměnných složených z variancí, tak distribuce  $V$  je přibližně normální [23].

### 3.4.3 Lineární modely pro popis časových řad

Jako první bych chtěl uvést metodu, jakou je možné analyzovat závislost dat v časové řadě. Uvažujme slabě stacionární řadu. Pokud jsou v časové řadě závislé hodnoty  $r_t$  a  $r_{t-i}$ , pak

<sup>1</sup>Více informací je možné zjistit zde: <http://vychodil.inf.upol.cz/kmi/pras/pr09.pdf>

mluvíme o autokorelaci. Pokud mluvíme o závislosti v jistém rozmezí, pak ji nazýváme i-lag autokorelation. Výpočet autokorelačního koeficientu je veden takto:

$$ac = \frac{Cov(r_t, r_{t-i})}{\sqrt{Var(r_t)Var(r_{t-i})}}$$

Čím vyšší autokorelační koeficient je, tím vyšší je nalezení závislosti v datech.

**Autoregresivní model** slouží pro stochastické lineární modelování časové řady. Podmínkou pro aplikaci modelu je slabá stacionarita signálu. AR(1) je počítán:

$$x_t = \phi_1 + \phi_2 x_{t-1} + \epsilon_t$$

Nad tímto polynomem je počítána autokorelace, která vyjadřuje závislost dat v čase. Tomuto jevu se říká kauzalita. Podle výše popsaných informací je nasazení tohoto modelu nemožné.

Složitějším rozšířením je model ARMA, což je spojení autoregresivního modelu a klouzavého průměru. Vzorec nejjednoduššího modelu  $ARMA(1, 1)$ :

$$r_t - \phi_1 r_{t-1} = \theta_1 a_t - \theta_1 a_{t-1}$$

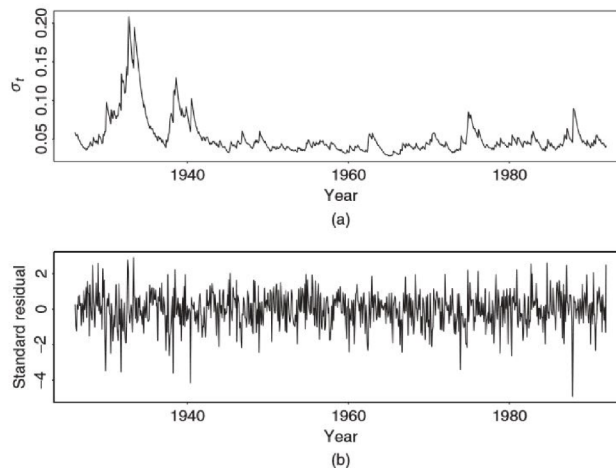
U systému musí platit podmínka stability lineárního systému, pokud chceme vyjadřovat náhodný proces jako kauzální, respektive pokud nad tímto modelem počítáme autokorelaci.

#### 3.4.4 Autoregressive conditional heteroskedasticity

Posledním popsaným modelem bude GARCH neboli generalized autoregressive conditional heteroskedasticity. Jde o model pro stochastické modelování volatility časové řady. GARCH má dva parametry a je definován takto GARCH(m,s):

$$a_t \epsilon_t, \omega_t^2 = \alpha_0 + \sum_{n=1}^{\infty} \alpha_n a_{t-n}^2 + \sum_{n=1}^{\infty} \beta_n \omega_{t-n}^2$$

kde  $\epsilon_t$  je sekvence náhodných čísel s jednotkovou variancí a nulovým průměrem.  $\alpha_0 > 0$ ;  $\alpha_i \geq 0$ ;  $\beta_j \geq 0$  a  $\sum_{n=1}^{\max(s,m)} (\alpha_n + \beta_n) < 1$ . Z podmínek lze vyvodit, že variance  $a_t$  je konečná. V knize je psáno, že GARCH model dobře popisuje heavy-tailed rozložení pravděpodobnosti. Pro ukázkou je v knize praktický příklad, kde na obrázku 3.4 je vidět využití GARCH modelu pro odhad směrodatné odchylky a standardizace cenové změny:



Obrázek 3.4: (a) ukazuje modelovanou směrodatnou odchylku finančního indexu S&P 500 pomocí GARCH modelu. (b) ukazuje standardizace měsíční změny indexu S&P 500. [25]

### 3.5 Studium Forexu z hlediska praxe

Pro porozumění této problematice je důležité nalézt vhodnou literaturu a také pochopit praxi. Otázkou je, za má někdo motivaci publikovat informace o fungujícím přístupu automatického obchodování. Z principů trhu, tak jak je popsán, tato motivace rozhodně nevyplývá.

Forex má silnou základnu malých obchodníků. Kromě knih existuje i mnoho dobrých webů, které nabízí edukaci a trénink obchodování pro jednotlivce přes brokera. Například web [fxstreet.cz](http://fxstreet.cz) uvádí informace, jak co nejpřesněji testovat historické obchody automatického systému pomocí systému Metatrader4 .

Z prostudovaných webů, jako je předchozí nebo například sociální síť [tradingview.com](http://tradingview.com), jsem vyvodil závěr o principech trhu, které jsou uvedeny v publikacích v kapitole 2. Trh mění svůj charakter, a tak je nutné analýzu trhu po jistém časovém úseku opakovat a aktualizovat přístup obchodování.

Systém vytvořený v této práci by měl simulovat práci obchodníka a adaptovat se na aktuální stav trhu.

### 3.6 Závěr

Pro návrh obchodního systému 6 jsou důležité podkapitoly 3.4.1 a 3.4.2. Informace o cenovém pohybu v čase je přínosem z důvodu nezávislosti na relativním měřítku vzhledem k danému aktivu. Také informace o charakteru dat je přínosem například pro případné strojové učení 3.3. Zbytek kapitol je hrubý popis pro ukázkou, co se od stochastického modelování ve finančnictví čeká. Systém by měl být navrhnout tak, aby podobné modelování případně uměl také.

**Nejpodstatnější souvislosti** jednotlivých podkapitol je možnost propojení jednotlivých přístupů. Podle přístupu v podkapitole 3.1 je možné do výpočtu zahrnout i složitější

<sup>1</sup><https://www.fxstreet.cz/>

<sup>1</sup><https://www.fxstreet.cz/serial-mql-a-backtestovani-strategii.html>

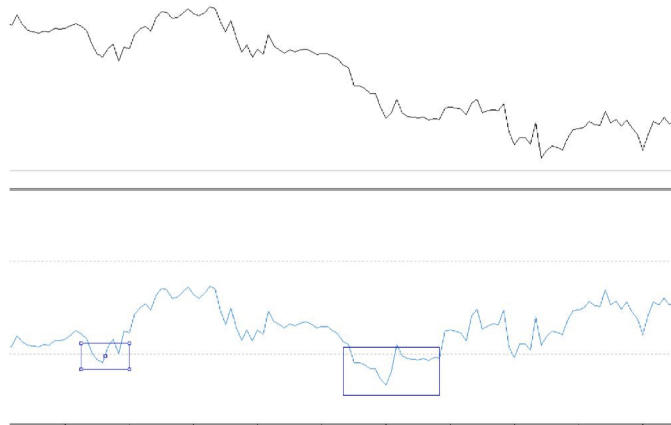
<sup>1</sup><https://www.tradingview.com/>

výpočty než pouze aplikaci operací sčítání, odečítání, násobení a dělení. Do výpočtu je možné zahrnout složitější heuristiky. V knize [5] je ukázáno, jak je možné dojít jednotlivými výpočty pomocí gramatické evoluce k výpočtu technických indikátorů. Také je možné využít heuristiky, jako jsou technické indikátory, neuronové sítě nebo autokorelace přímo jako operaci. Případně lze implementovat dostatek menších operací, jako jednoduché okenní i normální funkce, díky kterým se dopočítáme například k výše uvedeným lineárním modelům nebo autokorelaci, kterou můžeme také využít. Například pokud bychom měli v systému k dispozici funkci nad signálem pro posunutí, odečítání, dělení, výpočet průměru, odmocnění, druhou mocninu a násobení, tak je možné na základě gramatické evoluce získat výpočet s vlastnostmi pro výpočet *korelace* dvou signálů nebo *autokorelace*.

V kapitole 3.4.2 je uvedena zmínka o teorii náhodné procházky, která tvrdí, že cena není predikovatelná na základě historické ceny, tedy že náhodné vektory  $X$  a  $Y$  jsou nezávislé právě tehdy, když:

$$f_{x|y}(x, y, \Theta) = f_x(x, y, \Theta)$$

Od systému očekáváme, že bude hledat vektory na základě historické ceny, které tuto podmínku poruší. Pokud jsem návrh finančních modelů pochopil správně, tak z širšího pohledu přesně to dělají. Příkladem, jak takový vektor lze vytvořit, může být například opět výpočet *RSI*. Vektor náhodných hodnot  $X$  jako ceny close a vektor  $Y$  jako hodnoty *RSI* budou podmínku nezávislých proměnných porušovat. Je zanedbatelné, že vektor  $Y$  není náhodný. Důležité je, že podmíněná pravděpodobnost náhodného jevu  $x$  je rozdílná, pokud nastane jev  $y \leq 30\%$  v čase  $i$  od pravděpodobnosti marginální. Na obrázku 3.5 je vidět, že systém se může pokusit stanovit podmínky, které najdou korelaci dvou jevů a určit jejich podmíněnou pravděpodobnost dvou jevů. Konkrétní případ, kdy systém optimalizuje a hledá nejvhodnější hladinu pro indikátor *RSI*, je uveden v 4.



Obrázek 3.5: Obrázek ukazuje korelaci dvou jevů. Za podmínky, že modrá funkce vrací hodnoty menší než 30, pak je vyšší pravděpodobnost jevu, že černá funkce je ve svém lokálním minimu.

Závěrem je, že pokud budeme chtít aplikovat nějaké analýzy variance, případně posuzovat korelaci různých měnových párů, pokusíme se pracovat se slabě stacionárním signálem. Stacionaritu se pokusíme zajistit odečtením klouzavého průměru od uzavírací ceny.

## Kapitola 4

# Návrh systému

Možnosti návrhu musí podléhat základním podmínkám. Těmi jsou obecnost, rozšiřitelnost, možnost testování modulů. Vzhledem k tomu, že neexistuje obecný předpis, jak navrhovat obchodní systémy, je možné vzít v potaz více přístupů. Z prostudovaných materiálů je vidět pestrost různých přístupů, které jsou často diametrálně rozdílné a případně využívané v kombinaci. Je možné na problém pohlížet obecněji a snažit se využít co nejvíce informací. Vzhledem k charakteru dat je možné na problém pohlížet jako na teorii chaosu [24], je možné využít technik pro zpracování signálu [16]. Kapitola 3 je tedy kompendiem informací, které jsou podstatné pro návrh. Je otázkou, jak tyto informace vhodně využít. Možností je mnoho, vzhledem k prostudovaným materiálům [7] je možné vytvořit obchodní systém založený na technické analýze a na principech strategií z [13].

V úvahu jsem bral možnost rozhodovacího stromu, případně konstrukci konečného automatu [11] nebo systém založený na optimalizaci. Vzhledem k obecnosti chceme, abychom mohli systém rozšiřovat o parametry, které budeme chtít do obchodování zahrnout. Opět se zde otevírá otázka, jak parametry zobecnit. V úvahu připadalo sdílené pole hodnot, které by však bylo pevně dané a případná optimalizace by byla aplikována pouze na tyto parametry. Obchodní systém může brát v potaz širokou škálu možností. Pokud by měl využívat výše zmíněné, je nutné navrhnout, jak obecně pracovat se všemi prostředky. Jde o využití technických indikátorů, nastavení obchodů, zakomponování strojového učení. Výsledkem této úvahy je architektura popsaná v kapitole 6. V návrhu jsem se snažil vzít v potaz možnost přidávání nových výpočtů a heuristik. Pomocí gramatiky jsem schopen zajistit inicializaci systému s velkou počáteční rozmanitostí různých strategií. Systém by měl být dobře rozšiřitelný o nové prvky, například techniky strojového učení.

Tato kapitola podrobně popisuje první implementaci a její testování. Je důležitá pro pochopení finální verze systému 6. V práci je vytvořen systém, který využívá pouze knihovny pro genetické algoritmy, zpracování velkých dat a výpočty technických indikátorů. Logika vytváření obchodních strategií, simulace obchodů, návrh optimalizace, kontrola výpočtů a další logika přímo související s tímto systémem je vytvořena v této práci. První implementace je otestována spolu se základními principy, kterými jsou optimalizace jedné jednoduché strategie 5.1 a uvedení problému, který by měla práce vyřešit 4.1. Dále je vytvořen robustní systém, kde je navržen základní princip fungování. Na základě testů prvního návrhu je vylepšen do finální podoby 6.

## 4.1 Základní demonstrace

Před popisem systému uvedu základní problém, který by měl následující systém řešit. Podstatná myšlenka v nadcházejícím návrhu je nasazení systému pouze v aktuálním okně. Systém hledá strategie, které jsou v okně ziskové. Trh je proměnlivý a jsou přístupy, které fungují pouze v určitých intervalech. Jako příklad byla zvolena strategie překřížení klouzavého průměru počítaného přes 18 hodnot a ceny *Close*. Využil jsem naprogramovaný systém 6 pro její vyhodnocení. Ručně jsem napsal předpis výpočtu, jako:

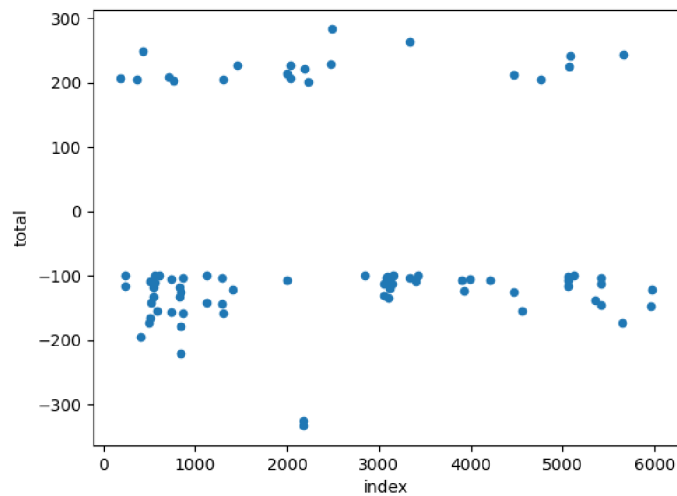
```
MA-18-MA|srcnorm|sub-MA,Close-subC|bool-subC-0.7-bool
```

Výše uvedený výpočet reprezentuje výpočetní postup, na kterém je systém založený. Myšlenka zaznamenání výpočtu byla uvedena v kapitole 3.1. A aplikace operací nad časovými řadami byla uvedena v kapitole o studiu Forexu 2.3, kde se zmiňuji konkrétně na obrázku 2.6 o postupné aplikaci operací nad vstupním signálem *Close*. Výše uvedený výpočet rozepíše podobným způsobem. Na vstupu do systému jsou signály přímo z Forexu, tedy 5 signálů *Open*, *High*, *Low*, *Close*, *Volume*. V prvním kroku je vzat signál *close* a dále jsou provedeny následující kroky:

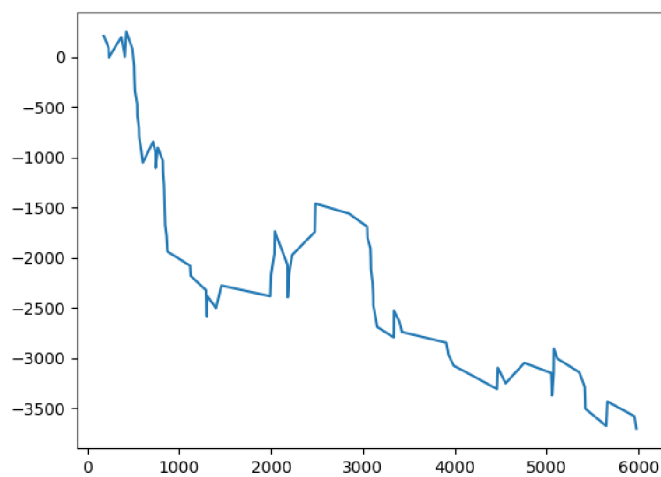
1. Nad signálem *Close* vypočítej klouzavý průměr o velikosti okna 18 a přidej ho do matice vypočítaných signálů s názvem *MA*. Akce *srcnorm* provádí speciální funkci, jejíž logika je využita v genetických algoritmech. V tomto případě také data normalizuje.
2. Následně je provedena funkce odečtení dvou signálů *sub*, která odečítá již vypočítaný signál *MA* a signál *Close*. Výsledný signál pět ulož do matice pod názvem *subC*. V matici jsou tedy signály *Open*, *High*, *Low*, *Close*, *Volume*, *MA*, *subC*.
3. V posledním kroce je aplikována operace *bool* nad signálem *subC* s dalším parametrem 0.7 jako hladinou, která indikuje vstup do pozice v případě překročení signálu *subC*. Hodnotu hladiny jsem zvládl určit snadno sám, neboť systém první verze využíval normalizaci, a tudíž je možné zvolit hodnotu mezi 0 a 1.

Vyhodnocení jsem provedl na časovém okně o velikosti 6000. Jednotlivé pozice jsem zobrazil na obrázku 4.1 a kumulativní součet pozic na obrázku 4.5.





Obrázek 4.1: Ukázka zisku jednotlivých pozic v čase

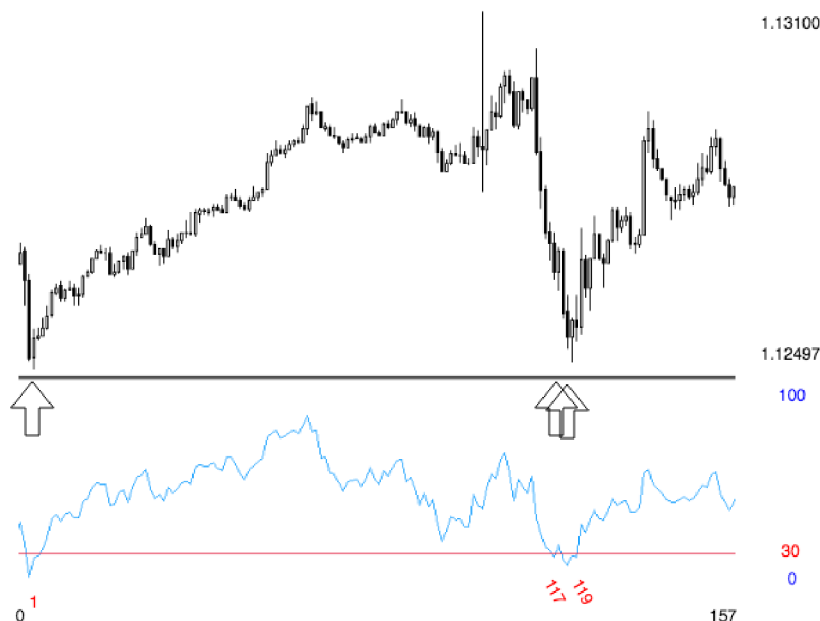


Obrázek 4.2: Kumulativní součet pozic.

Z obrázků by mělo být jasné, že pokud bychom strategii nasadili přibližně mezi indexy 1300 až 3000, tak bude zisková. Pokud bychom měli systém, který by například na indexu 1200 až 2000 odhalil, že tato strategie funguje, pak bychom ji mohli nasadit na určitou dobu, stanovenou například z předešlých testů.

Pro lepší pochopení, jak jsme došli k časům pro otevření pozice, je na obrázku vyznačen princip operace *bool*. Pro příklad je uveden krátký interval dlouhý 157 hodnot. V horní části obrázku jsou zobrazeny signály burzy, na kterých je pozice vyhodnocena z hlediska zisků a ztrát. V dolní části je signál, který je obsažen v matici výsledků a předán jako parametr pro operaci *bool* spolu s hladinou pro překřížení 30. Na místech, kde je signál překřížen shora dolů, je otevřena pozice. Tedy pro tento interval bychom získali tři místa (indexy).

Popsaná interpretace předepsaného výpočtu je spolu s genetickými algoritmy jádrem implementovaného systému v této práci.



Obrázek 4.3: Obrázek ukazuje princip získání indexů pro otevření pozice pomocí operace *bool*. Pro ukázkou je dolní modrý signál výsledkem předchozích výpočtů zapsaných podle výše uvedeného předpisu a následně je parametrem pro operaci *bool* spolu s hladinou pro překřížení nastavenou na 30. Kritické indexy jsou na místech, kde došlo k překřížení shora dolů. Tedy k danému příkladu bychom na intervalu 157 našli 3 místa pro vyhodnocení jako obchodní pozice

V druhé části základní demonstrace vygeneruji 1000 strategií na okně o velikosti 1000 hodnot. Strategie obsahuje 10 náhodně rozmístěných indexů pro zadání pozice. Jinými slovy jsou vybrány náhodné časy, ve kterých bude simulován vstup do pozice *long*. Výsledek v tabulce 4.1 není překvapivý. Ukazuje, že většina rozmístění je neziskových. Funkce pro ohodnocení vrací pro libovolnou ztrátu vždy hodnotu nula. Vše kromě zisku je nepřijatelné. Funkce pro simulaci obchodů na testovacích datech vrací reálnou ztrátu, ale pro optimalizaci je nejhorší možná hodnota 0.

Minimum	Maximum	Průměr	Směrodatná odchylka
0.00	316.00	0.56	10.79

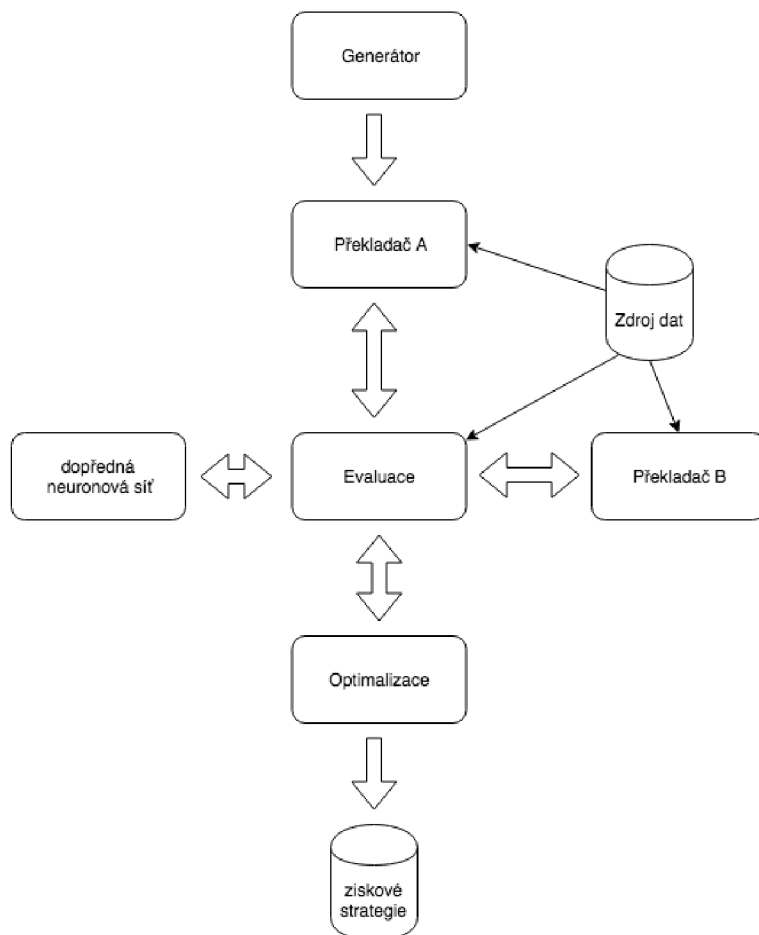
Tabulka 4.1: Tabulka ukazuje statistické shrnutí testu náhodně rozmístěných obchodů. Hodnoty ukazují, že většina náhodně rozmístěných obchodů bylo ztrátových. Hodnoty jsou vyhodnoceny z 1000 pokusů náhodného rozmístění nad intervalem 1000 hodnot. Každé náhodné rozmístění bylo vyhodnoceno jako obchodní pozice *long* a započítáno do statistiky

Na tomto příkladu bych rád uvedl, že pokud bychom našli postup operací aplikovaných na vstupní signály *Time, Open, High, Low, Close, Volume*, které vedou k výpočtu indexů, kde jsou otevřené pozice ziskové, pak bychom našli strategii ziskovou na daném období a můžeme spekulovat o její ziskovosti i v čase budoucím. Příklad demonstrující postup hledání jsem uvedl zde 2.3.

Jde o stejný princip, který je aplikován obchodníky při technické analýze. Obchodník pracuje s historickou cenou a hledá výnosnou strategii popisující aktuální vlastnosti trhu.

## 4.2 Architektura prvního návrhu systému

Systém je založen na gramatice generující jazyk, jehož věty reprezentují danou obchodní strategii. Systém generuje strategie reprezentované řetězcem pro popis posloupnosti výpočtů. Řetězce zpracovává překladač. Výsledkem je matice obsahující vypočtené hodnoty a popis vstupů do obchodní pozice. Hodnotící modul přijímá tuto matici a vyhodnotí průměrné zhodnocení obchodních pozic. Kooperace modulů a architektura systému je vyznačena na obrázku 4.4. Níže je popsáno, že překladač funguje ve dvou režimech. Překládá věty generované gramatikou, ale také výpočetní předpis generovaný prvním překladem a genetickou optimalizací. Důvodem byla parametrizace překladače, která měla v budoucnu sloužit k limitaci parametrů jednotlivých funkcí. A celková logika doplňování parametrů funkcím a jejich výpočet měl být přímo v překladači.



Obrázek 4.4: Architektura první implementace systému. Proces v čase nejdříve vygeneruje věty. Provede první překlad A vět na předpis výpočtu. Následně při evaluaci a optimalizaci je využíván překlad B pro předpis výpočtu.

Při inicializaci jsou vygenerovány věty. Věty jsou dále zpracovány překladačem A a jsou vygenerováni jedinci s předpisem výpočtu pro optimalizaci genetickými algoritmy. Modul pro poskytování dat drží formát popsany v kapitole 2. Jde o matici se sloupci *Time, Open, High, Low, Close, Volume*, kde velikost na ose  $x$  určuje počet historických dat. Následně je spuštěna optimalizace, která využívá překladač B pro překlad mutovaných předpisů výpočtu. Po ukončení optimalizace jsou k dispozici strategie pro testování nebo případné reálné nasazení.

### 4.3 Generátor vět

Podstatnou částí systému je generátor vět, který inicializuje první populaci. Vytváří množinu vět, které se přeloží a tvoří první populaci. Předpis gramatiky ovlivňuje celkový výsledek. V rámci testování jsem použil gramatiku, která využívá všechny výpočty. Gramatika je parametrem systému. Zápis gramatiky má jistá omezení. Pro správnou funkcionalitu je nutné po výpočtu technických indikátorů provést normalizaci celé tabulky, například takto:

```
S -> iS | iA
A -> nB
B -> sB | dB | C
C -> bC | k
```

Normalizaci jsem zavedl pro správnou kontrolu výpočtů prováděných nad maticí po vygenerování indikátorů a také pro ulehčení implementace. Indikátory počítám s hodnotami trhu, které jsou na vstupu. Pokud od systému chceme, aby výsledné strategie byly vyhodnoceny podle pozic, je nutné gramatiku navrhnout s generováním neterminálu pro spuštění algoritmu označení pozice v překladači.

terminál	význam
i	výpočet a přidání indikátoru
n	(srenorm) normalizace generované matice
s	(sub) odečtení dvou náhodně vybraných sloupců
d	(div) vydělení dvou náhodně vybraných sloupců
f	(diff) pro náhodně vybraný sloupec se aplikuje funkce pro výpočet rozdílu dvou po sobě jdoucích hodnot
l	(lag) pro náhodný sloupec se aplikuje posunutí směrem do budoucna
q	(log) pro náhodný sloupec se aplikuje logaritmus
p	(oneprops) generování vlastnosti sloupce
x	(windowmax) výpočet maxima v klouzavém okně pro náhodný sloupec
w	(windowmax) výpočet maxima v klouzavém okně pro náhodný sloupec
v	(onediv) pro náhodný sloupec se náhodně vybere jedno číslo, které všechny hodnoty vydělí
e	(oneexp) pro náhodný sloupec se náhodně vybere celé jedno číslo, které všechny hodnoty umocní
m	(onemul) pro náhodný sloupec se náhodně vybere jedno číslo, které všechny hodnoty vynásobí
b	(bool) pro náhodný sloupec se náhodně vybere hodnota jako hranice, kde dojde k možnému nastavení pozice

Tabulka 4.2: Tabulka s popisem jednotlivých terminálů

Například pro předpis níže může být vygenerována množina vět ve formátu předepsaném regulárním výrazem (*operace*|)+. Předpis níže je oproti předchozímu rozšířen o další operace z tabulky 4.5. Předchozí předpis gramatiky byl příkladem aplikace technických indikátorů, normalizace a odečítání.

```
S -> iS | iA
A -> nB
B -> sB | dB | C
C -> fC | lC | gC | D
D -> pD | E
E -> xE | wE | F
F -> vF | eF | mF | G
G -> bG | k
```

Příklad vět:

```
i|srcnorm|sub|div|div|sub|div|sub|sub|div|div|div|div|lag|lag|bool|bool|
  bool|bool|bool|final
```

```
i|srcnorm|div|sub|div|div|sub|div|sub|sub|log|diff|lag|lag|lag|log|log|diff
  |diff|lag|lag|lag|lag|log|diff|diff|log|diff|lag|log>windowmin|onemul|
  oneexp|bool|bool|bool|bool|bool|bool|final
```

Generátor vět je základem celého konceptu a tvoří jeden z parametrů systému. Podle stavu gramatiky můžeme ovlivnit výsledné chování. Vynecháním některých pravidel můžeme zcela pozměnit charakter strategií. Návrh gramatiky jsem využil především pro testování a vyhodnocení, jaké výpočty a heuristiky jsou zásadním přínosem.

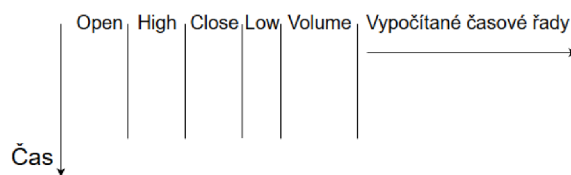
## 4.4 Překladač

Komponenta překladače vytváří na základě vět kompletní jedince jako objekt strategie, který drží několik parametrů pro popis strategie. Parametry reprezentující strategii jsou vyznačeny dále.

**Věta** je jediným neprázdným parametrem před tím, než objekt strategie zpracuje překladač.

**Předpis výpočtu** je celkový záznam výpočtu matice.

**Hlavní matice** je záznam všech vypočtených kroků. Uchovává také časový index pro označení obchodních pozic. Matice slouží pro celkové zpracování a vyhodnocení strategie evaluačním modulem, který poskytuje fitness funkci. Matice je vyznačena na obrázku. Zpočátku obsahuje pouze informace o ceně pro daný timeframe.



Obrázek 4.5: Ukázka uložení matice. Směr postupně přidávaných vypočtených dat, neboli nově vypočítaných signálů

**Matice vlastností** drží doplňující informace pro výpočet hlavní matice.

Pro pochopení, jak přesně **hlavní matice** vzniká, se podívejme na postup. Například pokud by nám předpis výpočtu určoval odečíst signál *open* a *close*, pak výsledek uložit jako *sub* a následně provést operaci *bool* s hladinou překřížení 0, tak průběh vytváření matice by vypadal jako postup v následujících tabulkách. Samozřejmě pro účely úspory místa postačí pouze malá délka matice.

Open	High	Low	Close	Volume
1.16250	1.16259	1.16222	1.16259	849.920002
1.16261	1.16289	1.16256	1.16276	860.410002
1.16277	1.16293	1.16268	1.16283	989.560001
1.16282	1.16292	1.16269	1.16274	519.260001

Tabulka 4.3: První krok postupu je inicializace matice

Open	High	Low	Close	Volume	sub
1.16250	1.16259	1.16222	1.16259	849.920002	-0,00009
1.16261	1.16289	1.16256	1.16276	860.410002	-0,00015
1.16277	1.16293	1.16268	1.16276	989.560001	0,00001
1.16282	1.16292	1.16269	1.16274	519.260001	0,00008

Tabulka 4.4: V druhém kroce je vypočítán rozdíl sloupců **Open** a **Close**

Open	High	Low	Close	Volume	sub	bool
1.16250	1.16259	1.16222	1.16259	849.920002	-0,00009	0
1.16261	1.16289	1.16256	1.16276	860.410002	-0,00015	0
1.16277	1.16293	1.16268	1.16276	989.560001	0,00001	1
1.16282	1.16292	1.16269	1.16274	519.260001	0,00008	0

Tabulka 4.5: V posledním kroku je vypočítán signál **bool** jako binární hodnoty, kde 1 určuje pozici překřížení zadané hladiny. Konkrétně zde šlo o sloupec **sub** a hladinu 0

Při zpracování jedince překladač pracuje ve dvou módech, pokud je prázdný parametr. *Předpis výpočtu* pak překládá větu a počítá matice. V opačném případě počítá hlavní matici pouze na základě *Předpisu výpočtu*. Při překladač věty jsou postupně zleva doprava zpracována všechna pravidla.

Zpracováním je myšleno provedení požadované funkce podle příkazu, jejíž výsledek je případně přidán do hlavní matice nebo do matice vlastností. Funkce jsou popsány v kapitole

4.3. V této kapitole bude podrobněji popsán jejich výpočet. Při zpracování věty jsou funkcím přidány náhodně parametry. Pro provedení výpočtu je výsledek uložen do jedné z matic a název funkce i s parametry je přidán do Předpisu výpočtu. V případě neterminálu  $i$  je náhodně vybrán jeden z indikátorů pro výpočet 4.4.

MA - klouzavý průměr.

$$MA_i(n) = \frac{1}{n} \sum_{i=1}^n P_{n-i}$$

EMA

$$EMA_i(n) = (P_i - EMA_{i-1}) \frac{2}{n+1} + EMA_{i-1}$$

Momentum

$$P_i - P_{i-n}$$

ROC / Closing price n periods ago x 100

$$\frac{P_i - P_{i-n}}{P_{i-n} \times 100}$$

ATR

$$EWMA(\max(Phigh_i, Pclose_{i-1}) - \min(Plow_i, Pclose_{i-1}))$$

kde EWMA je exponentially weighted moving average počítaný takto:

$$S_t = aP_i + (1 - a)S_{t-1}$$

$a$  - Koeficient reprezentuje stupeň snížení váhy. Vyhlažovací faktor mezi 0 a 1.

$S_t$  - exponentially weighted moving average v daném čase.

STO

$$S_i = \frac{Pclose_i - Plow_i}{Phigh_i - Plow_i} \times 100$$

$$STO_i = \frac{1}{n} \sum_{i=1}^n S_{n-i}$$

Trix

$$Trix_i = 3 \times EWMA_i - 3 \times EWMA(EWMA_i) - EWMA(EWMA(EWMA_i))$$

Vortex

$$TR_i = (\max(Phigh_i, Pclose_{i-1}) - \min(Plow_i, Pclose_{i-1}))$$

$$VM_i = (\text{abs}(Phigh_i - Plow_{i-1}) - \text{abs}(Plow_i - Pclose_{i-1}))$$

$$Trix_i = \frac{\sum_{k=i}^n TR_k}{\sum_{k=i}^n VM_k}$$

RSI

$$RSI_t(n) = 100 - \frac{100}{1 + \frac{U(n)}{D(n)}} \quad (4.1)$$

U(n) součet kladných cenových změn za období délky n

D(n) součet záporných cenových změn za období délky n

OBV

$$L_i = \begin{cases} volume_i & \text{if } Pclose_i - Pclose_{i-1} > 0 \\ 0 & \text{if } Pclose_i - Pclose_{i-1} = 0 \\ volume_i \times -1 & \text{if } Pclose_i - Pclose_{i-1} < 0 \end{cases} \quad (4.2)$$

$$OBV_i(n) = \frac{1}{n} \sum_{k=(i-n)}^n L_{i-k}$$

EoM - Ease of Movement:

$$L_i = (Phigh_i - Phigh_{i-1}) + (Plow_i - Plow_{i-1}) \times \frac{(Phigh_i - Plow_{i-1})}{2 \times volume_i}$$

$$EoM_i(n) = \frac{1}{n} \sum_{k=(i-n)}^n L_{i-k}$$

CCI

$$EoM_i(n) = \frac{P_i - MA_i}{0.015 \times \sigma}$$

kde  $\sigma$  je směrodatná odchylka  $MA$

Copp Copp = EWMA(ROC(11) + ROC(14), n)

Výpočet indikátorů je heuristikou, od které očekávám rychlejší nalezení ziskových strategií. Seznam indikátorů 4.4 jsem uvedl zejména pro ukázkou, jaké matematické operace obsahují. V podstatě je možné, že by systém k těmto výpočtům došel pouze s ostatními operacemi 4.5. Provedení operace *srcnorm* představuje normalizaci, která nenaruší varianci, tedy pouze časovou řadu transformuje do oboru hodnot mezi 0 a 1. Tato operace také zajistí, že všechny nové vypočítané řady budou normalizovány předtím, než jsou přidány do hlavní matice.

$$X_i = \frac{X_i - \min(X)}{\max(X) - \min(X)} \quad (4.3)$$

Operace *sub*, *div*, *mul*, *diff*, *log*, *lag*, *oneprops*, *windowmax*, *windowmin*, *onediv*, *oneexp*, *onemul* náhodně vyberou jeden nebo dva operandy mezi již existujícími sloupci hlavní matice. Náhodně vygenerují parametr. Zvolí název sloupce a provedou danou operaci. Výsledkem je nová časová řada, která projde případnou normalizací a je přidána do hlavní matice. Tvar matice se tedy po každém výpočtu změní. Provede se nejen přidání sloupce, ale také může dojít k odebrání některých řádků. Například pokud je aplikovaná operace v klouzavém okně, pak jsou v matici hodnoty, pro které není vypočítána hodnota provedenou operací. Je tedy nutné odstranit celý řádek. Pokud počítáme například klouzavý průměr přes deset hodnot, pak popsání situace nastane pro prvních devět hodnot.

Operace *bool*, která je gramatikou umístěna na konec, určuje časový index v hlavní matici pro otevření pozice. Náhodně vybere jeden ze sloupců a provede algoritmus *označení pozice*. Pro vybraný sloupec, tedy časovou řadu, je vybrána hladina z oboru hodnot, kde protnutí bude určovat vstup do pozice. Graficky byl tento algoritmus popsán na obrázku 4.3 v podkapitole 4.1. Konkrétně v krocích je tento algoritmus implementován takto:



1. Ze zkoumaného signálu vytvoř signál, kde kladná čísla jsou 1 a záporná 0.
2. Signál zkopíruj, zůstanou dva stejné signály obsahující pouze hodnoty 1 a 0.
3. Jeden se signálů posuň o jedna.
4. Signály odečti. Výsledkem je nový signál, kde -1 a 1 značí index překřížení.

Algoritmus nastaví v časovém indexu 1 nebo  $-1$ , pokud dojde k překřížení hladiny shora nebo zespoda. Operace *bool* opět po provedení přidá operaci i s parametrem do *předpisu výpočtu*.

Po zpracování věty je tedy *Předpis výpočtu* ve tvaru  $(operace - operandA(, operandB)? - (parametry -) * |) *$ . Podle *Předpisu výpočtu* může být strategie dále zkoumána nebo nasazena pokud ji systém vyhodnotí jako ziskovou. Příklad překladu *věty* na *Předpis výpočtu* je následující:

```
i|i|i|i|srcnorm|div|div|div|onediv|bool|bool|bool|final
```

```
MA-33-MA_XTMFZE|ATR-46-ATR_JQEVIM|ATR-52-ATR_EREYFL|CCI-7-CCI_JQWPFV|
srcnorm|div-Low,CCI_JQWPFV-div_QUTZGY|div-Volume,div_QUTZGY-div_DTQWRI|
div-ATR_EREYFL,div_DTQWRI-div_WHWWQA|onediv-Close-0.33973084225923667-
onediv_OBAIRF|onediv-div_DTQWRI-0.017998461092171847-onediv_IMFFJV|bool
-Volume-0.6469928741458684-bool_WFOFCY|bool-div_WHWWQA
-0.6457190703788932-bool_GXVDRZ|bool-div_QUTZGY-0.6158056574393338-
bool_DVGMUT
```

Funkcionalita překladače je dále rozšířena o možnost výpočtu hlavní matice pomocí *Předpisu výpočtu*. Překladač je pro přepočítání využíván při optimalizaci a také je možné separátně strategii přepočítat například na datech z jiného časového období.

## 4.5 Optimalizační modul

Modul optimalizace využívá generátor vět s překladačem pro inicializaci první generace. Překladač je dále využíván pro přepočítání matice po operaci mutace a křížení. Využívá evaluační modul, který poskytuje fitness funkci.

Optimalizace je inspirována genetickými algoritmy 3.2. Reprezentaci jedince jsem navrhl jako objekt s parametry uvedenými v kapitole překladače 4.4. Pro optimalizaci je důležitým parametrem *Předpis výpočtu*. Nad tímto řetězcem jsou počítány operace mutace a křížení.

Mutace by měla být postavena následovně: prostor jedinců, který se optimalizace snaží prohledat a najít optimální strategie, je třeba projít celý i přesto, že jsme našli lokální maximum. Pokud by byla délka řetězce omezena, pak by měla být mutace sama schopna projít celý uvažovaný prostor.

Tedy implementace mutace projde celý řetězec a náhodně vymění buď funkci, nebo pouze její parametr. Je třeba kontrolovat, zda je výměna možná vzhledem k tomu, že jednotlivé operace mohou být unární nebo binární.

```
MA-33-MA_XTMFZE|ATR-46-ATR_JQEVIM|ATR-52-ATR_EREYFL|CCI-7-CCI_JQWPFV|
srcnorm|div-Low,CCI_JQWPFV-div_QUTZGY|div-Volume,div_QUTZGY-div_DTQWRI|
div-ATR_EREYFL,div_DTQWRI-div_WHWWQA|onediv-Close-0.33973084225923667-
onediv_OBAIRF|onediv-div_DTQWRI-0.017998461092171847-onediv_IMFFJV|bool
-Volume-0.6469928741458684-bool_WFOFCY|bool-div_WHWWQA
```

-0.6457190703788932-bool\_GXVDRZ|bool-div\_QUTZGY-0.6158056574393338-bool\_DVGMUT

STO-33-MA\_XTMFZE|ATR-46-ATR\_JQEVIM|ATR-52-ATR\_EREYFL|CCI-7-CCI\_JQWPFV|srcnorm|sub-Low,CCI\_JQWPFV-div\_QUTZGY|div-Volume,div\_QUTZGY-div\_DTQWRI|div-ATR\_EREYFL,div\_DTQWRI-div\_WHWWQA|onediv-Close-0.33973084225923667-onediv\_OBAIRF|onediv-div\_DTQWRI-0.017998461092171847-onediv\_IMFFJV|bool-Volume-0.6469928741458684-bool\_WFOFCY|bool-div\_WHWWQA-0.6457190703788932-bool\_GXVDRZ|bool-div\_QUTZGY-0.752-bool\_DVGMUT

Návrh křížení byl složitý. Bylo vyzkoušeno několik přístupů, jak bych mohl provést náhodné křížení v bodě. Po konzultaci s autorem přednášky [22] prof. Sekaninou jsem tento problém uzavřel s cílem křížit jedince v pevném bodě. Za výběrem bodu stojí opět heuristika stanovená v této práci, že optimální bude křížit jedince v bodě normalizace. Již efektivní kombinace indikátorů s již efektivní kombinací dalších výpočtů u daného jedince může být zkombinována s jedincem s podobnými vlastnostmi. Příklad křížení je následující:

MA-33-MA\_XTMFZE|ATR-46-ATR\_JQEVIM|ATR-52-ATR\_EREYFL|CCI-7-CCI\_JQWPFV|srcnorm|div-Low,CCI\_JQWPFV-div\_QUTZGY|div-Volume,div\_QUTZGY-div\_DTQWRI|div-ATR\_EREYFL,div\_DTQWRI-div\_WHWWQA|onediv-Close-0.33973084225923667-onediv\_OBAIRF|onediv-div\_DTQWRI-0.017998461092171847-onediv\_IMFFJV|bool-Volume-0.6469928741458684-bool\_WFOFCY|bool-div\_WHWWQA-0.6457190703788932-bool\_GXVDRZ|bool-div\_QUTZGY-0.6158056574393338-bool\_DVGMUT

Copp-95-Copp\_KCLEFO|STO-54-STO\_SHEZSF|ROC-33-ROC\_HMLBFS|srcnorm|div-Copp\_KCLEFO,Low-div\_KQQCZL|div-STO\_SHEZSF,ROC\_HMLBFS-div\_ITSUME|div-High,STO\_SHEZSF-div\_AXNOLJ|div-div\_KQQCZL,Volume-div\_IYWEOL|div-Close,High-div\_KATWYM|sub-div\_KQQCZL,ROC\_HMLBFS-sub\_JOVHLZ|sub-Open,div\_ITSUME-sub\_ALVBVZ|sub-div\_KQQCZL,sub\_ALVBVZ-sub\_IJFUKS|div-Open,Volume-div\_WYCNDX|sub-sub\_IJFUKS,div\_KATWYM-sub\_EBPPKQ|sub-ROC\_HMLBFS,div\_IYWEOL-sub\_HKGEK|div-div\_ITSUME,div\_KQQCZL-div\_GPTAYG|div-sub\_HKGEK,STO\_SHEZSF-div\_HTLZDJ|sub-div\_ITSUME,sub\_HKGEK-sub\_QOBCVC|div-Low,div\_KQQCZL-div\_XSIQEJ|div-div\_ITSUME,div\_IYWEOL-div\_KJHZJZ|bool-sub\_IJFUKS-0.6096366953175696-bool\_QPIUHA|bool-div\_KQQCZL-0.8278115654647632-bool\_IBAFGO

MA-95-Copp\_KCLEFO|ATR-54-STO\_SHEZSF|ATR-33-ROC\_HMLBFS|srcnorm|div-Copp\_KCLEFO,Low-div\_KQQCZL|div-STO\_SHEZSF,ROC\_HMLBFS-div\_ITSUME|div-High,STO\_SHEZSF-div\_AXNOLJ|div-div\_KQQCZL,Volume-div\_IYWEOL|div-Close,High-div\_KATWYM|sub-div\_KQQCZL,ROC\_HMLBFS-sub\_JOVHLZ|sub-Open,div\_ITSUME-sub\_ALVBVZ|sub-div\_KQQCZL,sub\_ALVBVZ-sub\_IJFUKS|div-Open,Volume-div\_WYCNDX|sub-sub\_IJFUKS,div\_KATWYM-sub\_EBPPKQ|sub-ROC\_HMLBFS,div\_IYWEOL-sub\_HKGEK|div-div\_ITSUME,div\_KQQCZL-div\_GPTAYG|div-sub\_HKGEK,STO\_SHEZSF-div\_HTLZDJ|sub-div\_ITSUME,sub\_HKGEK-sub\_QOBCVC|div-Low,div\_KQQCZL-div\_XSIQEJ|div-div\_ITSUME,div\_IYWEOL-div\_KJHZJZ|bool-sub\_IJFUKS-0.6096366953175696-bool\_QPIUHA|bool-div\_KQQCZL-0.8278115654647632-bool\_IBAFGO

Vyměněno bylo tolik indikátorů, kolik obsahoval jedinec s menším počtem indikátorů. Zbytek indikátorů byl ponechán nezměněn. Výsledkem křížení může být i potomek s opačnou výměnou. Zamýšlenou heuristikou je především myšlenka, že indikátory tvoří základ technické analýzy, tedy obchodních strategií. Pokud byl jistý přepočít nad indikátory úspěšný, je možné, že s jinými indikátory bude také úspěšný. Optimalizační modul je možné nastavit, tak aby provedení operací mutace a křížení proběhlo pouze s jistou pravděpodobností. Pro selekci potomků v další generaci je využit algoritmus turnajový výběr [12]. Jde o populární přístup, ve kterém jsou jedinci náhodně vybráni a ohodnoceni fitness funkcí. V daném kole vyhrává jedinec s nejlepším ohodnocením a postupuje dále. Výherce postupuje do dalšího kola turnaje.

## 4.6 Evaluační modul

Obsahuje několik užitečných funkcí pro zpracování dat. Tento modul je využíván modulem pro strojové učení a modulem optimalizačním. Zastupuje jakékoliv funkce pro vytvoření statistik. Například počet ziskových a ztrátových obchodů. Poskytuje pole indexů, na kterých jsou vypočítané pozice v *hlavní matici*.

Nejdůležitějším článkem je poskytování funkcí pro ohodnocení jedince. Tedy poskytuje fitness funkce pro genetickou optimalizaci. Důležitost návrhu ohodnocení jedince je patrná z teorie 3.2. Je žádoucí, aby ohodnocení odpovídalo podmínkám při reálném nasazení.

Konkrétně v této práci beru v potaz, že ohodnocení bude závislé na čase a opět není vhodné využívat informace v čase  $t + n$ , pokud poslední přijatou cenou je cena v čase  $t$ . Dále je popsána funkce pro ohodnocení pozice long. Ohodnocení pozice short je postavené na stejných principech.

Otevření pozice přináší dva problémy. Prvním je správný výběr času pro výstup z pozice neboli prodání měny za co největší cenu v případě pozice *long*. Problém správného výstupu by mohl být opět řešen systémem samotným. Ovšem rozšiřování a obecné nasazení systému je nad rámec této práce. Pro začátek jsem se řídil základním postupem při návrhu strategie 2.2. Tedy určil jsem pevnou hladinu pro výstup z pozice, která bude dosahovat vyšších zisků než potenciální ztráta určená hladinou pro ukončení pozice v případě rostoucích ztrát.

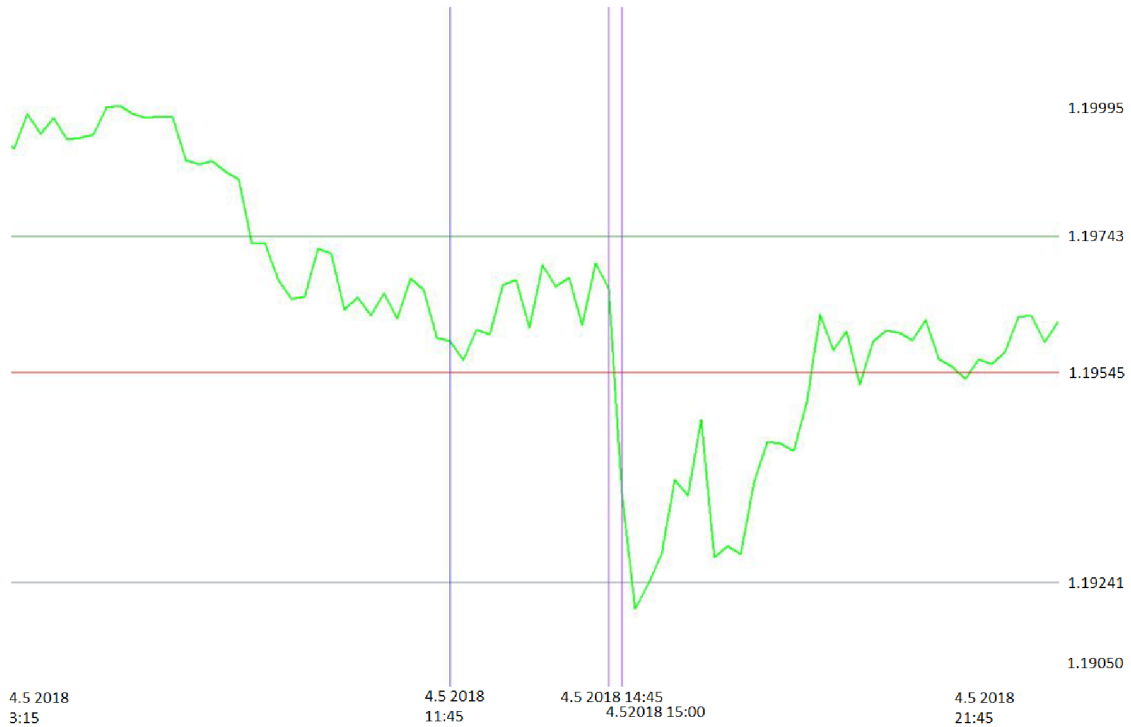
Také je možné aplikovat metodu posouvání *stoplossu* zvanou **trailing stop** 2.2. Metoda *trailing stop* byla implementována jako první a bylo provedeno několik testů 5.2.

Dále systém poskytuje vyhodnocení pozice podle dosaženého maxima v případě zisku a ztrát opět podle určené hladiny *stoploss*. Použití této funkce vysvětluje, proč jsem použil výraz, že není vhodné využívat informace z budoucnosti. Přímo jsem tento přístup nevyloučil. Určení lokálního maxima v aktuálním čase bez znalosti budoucích hodnot není možný. Funkci jsem implementoval pro porovnání, zda takto přesné ohodnocení nepřinese lepší výsledky v průběhu optimalizace. Nasazení optimalizované strategie touto funkcí by podléhalo omezení nemožnosti hledané maximum určit. Ziskový výstup z pozice by mohl být stanoven jako *take profit* na cenové hladině průměru mezi maximem a minimem z trénovací sady.

Poslední implementovanou a výsledně používanou ohodnocovací funkcí je stanovení pevné hladiny *take profit* a *stop loss*. Řešení *take profitu* a *stop lossu* bere v potaz, že zkoumaný signál je diskretní. Zaznamenání hladiny tak nemusí být přesné a je nutné tuto skutečnost modelovat. Uvedená nepřesnost spolu s vyhodnocením pozice je uvedena na obrázku 4.6

Černá vertikální čára v čase 11:45 značí vstup do pozice. Funkce sleduje budoucích 100 uzavíracích cen a pokud v tomto období není zaznamenán průraz *take profitu* nebo *stop lossu*, pak je připsána cena konečná. Tedy v čase  $t = 11 : 45$  začne simulace vyhodnocení

na historických datech. Horizontální zelená čára na cenové hladině 1,19743 vyznačuje *take profit* a fialová na hladině 1,19545 *stop loss*. Tedy máme nějaký *money management*, který určuje možnou ztrátu přibližně 50 bodů. Mezi přijetím uzavírací ceny v čase 14:45 a 15:00 dojde k propadu přibližně 200 bodů a *stop loss* je tak zaznamenán s přibližnou ztrátou 200 bodů. K této situaci může dojít na libovolném časovém rozlišení včetně tickových dat.



Obrázek 4.6: Ukázka nepřesného překročení hladiny

## 4.7 Modul dopředné neuronové sítě

Díky nabytým znalostem o dopředné neuronové síti (ANN) jsem se rozhodl ji do systému zakomponovat. Pokud jde o model, který obecně aproximuje funkci, pak by mohla být využita pro aproximaci strategie, která je výsledkem optimalizace. Byly navrženy dva přístupy. Prvním, navrženým bylo trénování přímo na burzovních datech jako cílené funkci k aproximaci. Z kapitoly 3.4 víme, že data nemají normální rozložení. Také víme, že vzhledem k vlastnostem ANN je vhodné trénovat neuronovou síť s daty s normálním rozložením. Také je problematické určit způsob normalizace. Signál  $s$  má definiční obor  $s \in [0, \infty)$ . Je tedy problém určit minimum a maximum pro správnou normalizaci. Z kapitoly 3 lze vyvodit, jak signál burzy zpracovat tak, aby data byla v normálním rozložení, a navrhnout strategii pro trénování celého signálu jako regresi.

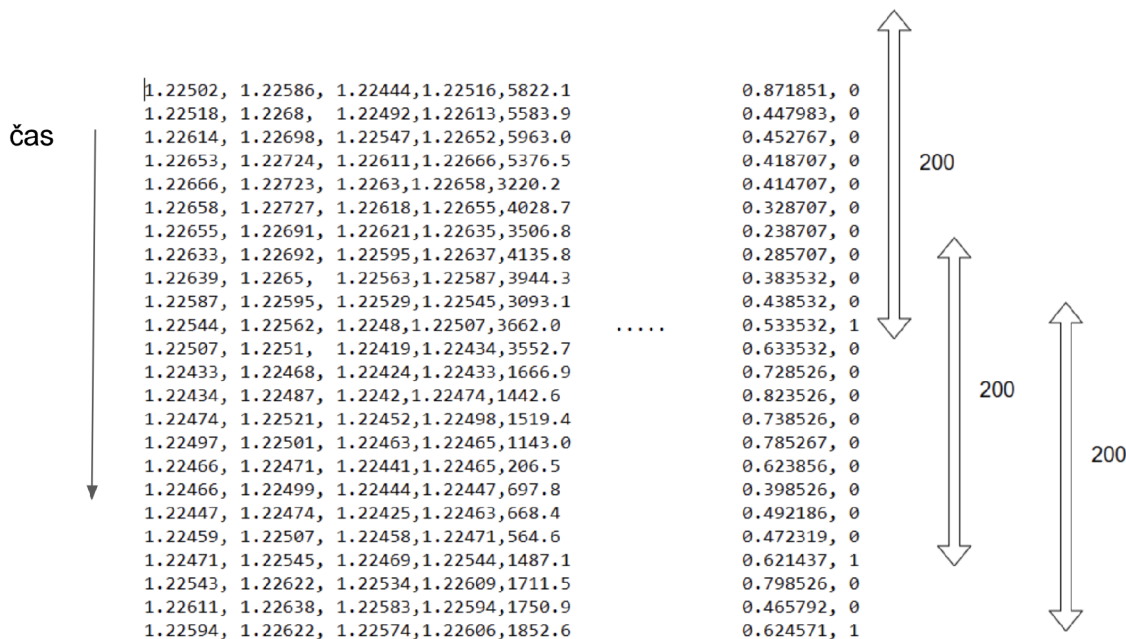
Rozhodl jsem se však pro implementaci druhého způsobu, která je napojena na logiku GA. Strategie je využita jako funkce projekce. Signály jsou transformovány na matici, která je výsledkem funkce klouzavého okna nad signálem. Vektor s indexem  $i$  o délce  $k$  je vybrán jako okno:

$$w_i = (n_{i-k}, n_k) \quad (4.4)$$

Funkce *klouzavého okna* je aplikována na všechny sloupce *hlavní matice*. Vznikne tedy vektor  $v$  matic vypočítaných klouzavými okny.

Následně jsou vybrány pouze vektory s indexem  $i$ , u kterých je splněna podmínka, že vektor ze signálu *bool* je ukončen hodnotou 1 pro index  $i$ . To indikuje, že jde o vektory, které přímo předcházejí otevření pozice. Matice  $z$  vektoru  $v$  po selekci drží separovaně pouze vektory  $w$ , které budou využity pro trénování.

Pro lepší představu je možné se podívat na obrázek:



Obrázek 4.7: Ukázka selekce vektorů. Intervaly ukazují 200 hodnot, které jsou použity pro vytvoření trénovacího vektoru od indexu, označeném logicky 1.

Vektory  $w$  jsou dále zpracovány pro následné trénování neuronové sítě. Jejich referenčním vektorem je záznam, zda byla pozice zisková, tedy vektor o velikosti 1. Vektory se společným indexem  $i$  jsou konkatenovány a normalizovány se zachováním variance. Dále jsem aplikoval redukci dimenze pomocí *PCA*.

Sestavení samotného modelu *ANN* jsem definoval zejména s ohledem na malé množství trénovacích dat, a tak jde o model s jednou vstupní vrstvou o velikosti závisující na nastavení *PCA*. Systém jsem testoval s *PCA*, které vybírá 100 komponent z vstupního vektoru. Velikost je tedy 100. Výstupní vrstva je o velikosti 1. Využívám funkci *tanh* pro vstupní a *softmax* pro výstupní vrstvu. Jako optimalizační techniku jsem zvolil *gradient descent* a *learning rate* 0.001. V testech jsem zkoušel různá nastavení, ale výše zmíněné беру jako konečné.

Logika napojení *ANN* do systému je přímočará. Jde o oddělený systém, který vedle vstupu pouze na základě strategie predikuje, zda bude pozice zisková, a pokud je predikce kladná, pak pozici otevře.

Závěrem dodávám, proč jsem neprováděl další přepočty vyplývající z kapitoly 3. U jednotlivých strategií očekávám totiž různé chování. Každá strategie bude držet vypočítané hodnoty, které budou mít různé vlastnosti. Empiricky bych chtěl při testování zjistit, zda tomu tak bude.

## Kapitola 5

# Testování a vyhodnocení problémů

V následujících podkapitolách jsou popsány testy trénování a vyhodnocení na testovacích datech v různé konfiguraci. Kapitola je přínosem i pro čtenáře, který si chce udělat přehled o experimentech, jejichž výsledky stanovily další směr a vedly ke konečnému návrhu 6, neboť jsou také obsahem kapitoly.

Pro testování a trénování jsem stáhl historická data měnového páru EUR/USD v časovém rozlišení patnácti minut. Celkově je k dispozici 12 tisíc záznamů o ceně za rok 2017 a začátek roku 2018. Z pohledu dní jde přibližně o data ze 125 historických dní. Trénování je prováděno v rámci jednoho měsíce. Test je proveden na navazujícím období sedmi dnů. Historická data pro tuto práci poskytla banka Dukascopy <sup>1</sup>. Obecně napříč testy byly testována různá období. Žádný časový úsek nebyl vyhodnocen jako extrém. Vyvodil jsem tak závěr, že systém pracuje rovnoměrně napříč časovými úseky. Ovšem v průběhu testování jsem zjistil několik problémů, které vedly k úpravě systému do konečné verze 6.

### 5.1 Test základního principu optimalizace

Vyhodnocení chování genetických algoritmů je nutné provést na dostatečném počtu pokusů [22]. Cílem této podkapitoly je uvést několik možností a konstatovat chování navrženého systému vzhledem k optimalizaci pomocí genetických algoritmů. Genetické algoritmy jsou založeny na operacích 3.2, které mohou být pojety různě. V kapitole 4.2 je popsáno jejich nastavení. Při návrhu systému jsem uvažoval i jiné možnosti. Než přejdu k testům samotné architektury, zmíním test, který testuje základní princip a posuzuje nasazení genetických algoritmů na jednoduchém případu. Pokusím se vyhledat hladiny RSI, které nejlépe odpovídají indikaci, že dojde k obratu kurzu. Test jsem provedl na datech páru EUR/USD s intervalem patnácti minut. V rámci testu jsem uvažoval pouze uzavírací cenu a indikátor RSI. V rámci testu byl prověřen algoritmus označení pozice, jak je uveden v kapitole 4.2. Cílem bylo nalézt optimální hodnotu RSI, která bude indikovat vhodnou situaci pro otevření pozice long a short. První populace byla množina jedinců s přepočtem RSI v intervalu 5 až 100 a náhodnou hladinou pro vstup do pozice v intervalu oboru hodnot RSI. Jedinec je tedy reprezentován dvěma parametry. Mutace byla implementována jako náhodné přičtení k aktuální hodnotě parametrů v intervalu 1 až 100 s validací, že výsledná hodnota parametru je kladná. Nastavení a funkcionality ostatních faktorů genetického algoritmu byla stejná jako výsledná architektura. Provedl jsem třicet spuštění s množinou 100 jedinců pro první populaci. Průměrné zhodnocení výsledků je uvedeno v tabulce.

---

<sup>1</sup><https://www.dukascopy.com/>

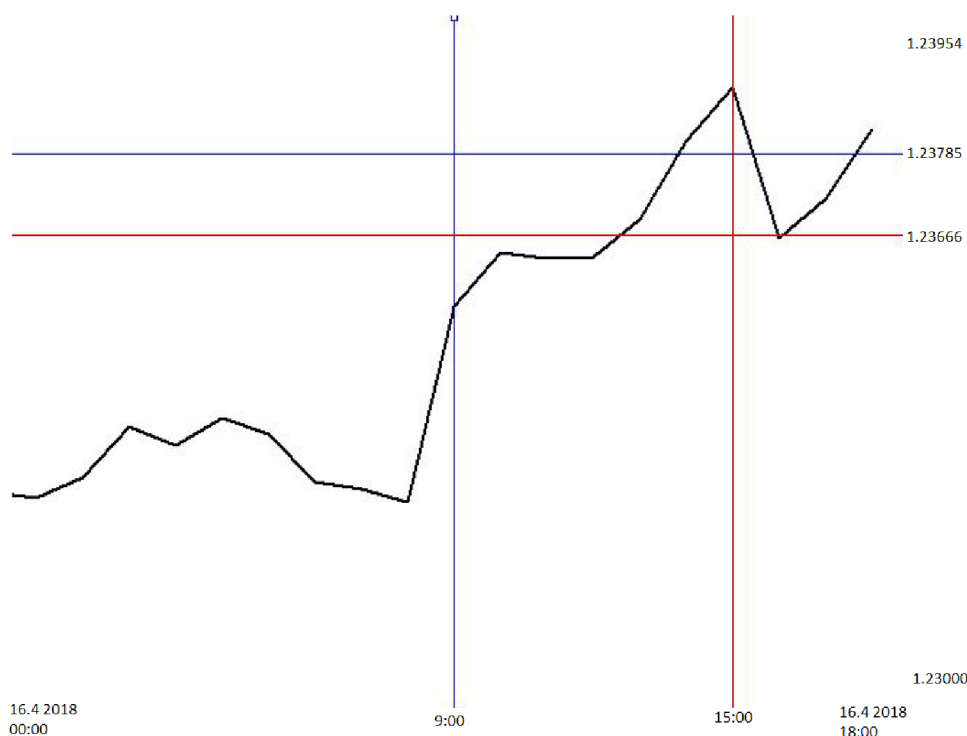
Z výsledků jsem vyvodil, že na tomto základním testu návrh optimalizace uspěl, neboť jednotlivé výsledky došly k podobnému závěru. Výsledky se shodují s teoretickým popisem charakteru indikátoru RSI.

Tabulka 5.1: Test základního principu systému na indikátoru RSI

	průměr	variance
Hladina RSI pro pozici long	21	8
Hladina RSI pro pozici short	83	5
Období pro výpočet RSI	28	4

## 5.2 Testy fitness funkce trailing stop

Z velkého počtu možností, jak implementovat fitness funkci, byla vybrána jako první simulace trailing stop. Překážkou nastavení byl především charakter dat. Vzhledem k výpočetní náročnosti optimalizace jsem zvolil vstupní data v rámci daného timeframu. Pokud by systém zpracovával ticková data, výpočet by byl paměťově náročný a zpomalení by bylo neúměrné vzhledem k přínosu, který by tento charakter dat měl. Například pro testovací data pro timeframe 15 minut je rozdíl mezi cenou v čase značný, a tak stop loss může být aktivován v propadu větším než je nastavený limit. Následující příklad situaci vysvětluje.



Obrázek 5.1: Obrázek ukazuje průběh pozice s funkcí trailing stop. Vstup do long pozice je označen modrou vertikální čarou v čase 9:00, pak následuje vzestup do maxima v čase 15:00. Trailing stop je vyznačen pomocí horizontální modré čáry a je nastaven přibližně na 50 bodů. Pokud cena poklesne, očekává se ukončení pozice na hladině horizontální modré čáry, ale pokles je zaznamenán až na hladině horizontální červené čáry, kde je přijata následující cena

Vstup do pozice long v čase 9:00 je označen modrou vertikální čarou. Následuje vzestup až do lokálního maxima v čase 15:00. Horizontální modrá čára vyznačuje nastavený trailing stop loss přibližně na 50 bodů. Pokud cena poklesne, očekává se ukončení pozice na hladině horizontální modré čáry. Pokles je však zaznamenán až na hladině horizontální červené čáry, kde je přijata následující cena.

## Test 1

- data - 15 minutové svíčky páru EURUSD
- fitness funkce - simulace trailing stop s omezením 15 min.
- počet spuštění pro každé nastavení - 30
- použitá gramatika:

```
S -> iS | iA
A -> nB
B -> sB | dB | C
C -> fC | lC | gC | D
D -> pD | E
E -> xE | wE | F
F -> vF | eF | mF | G
G -> bG | k
```

Význam neterminálů je vysvětlen zde [2.1](#)

- Sloupec A ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 100. Jsou použity operace **mutace** a **křížení**. Mutace je založena pouze na náhodné **změně číselných parametrů**, jak je uvedeno v popisu architektury [4.2](#). Stejně tak křížení.
- Sloupec B ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Také jsou použity operace **mutace** a **křížení**. A to ve stejném nastavení jako test pro sloupec A.
- Sloupec C ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá pouze operaci mutace založenou na náhodné **změně číselných parametrů**.
- Sloupec D ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá mutaci, která zaměňuje číselné parametry, operace a operandy.
- Sloupec E ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 1000. Jinak je nastavení stejné jako pro sloupec D.

Z výsledků jsem vyvodil, že výsledek je závislý na velikosti první populace, která je generována uvedenou gramatikou. Gramatika samotná ovlivňuje výsledek. Křížení v normalizaci je přínosem a bude dále využíváno. Mutaci je opravdu nutné navrhnout, tak aby byla schopna vytvořit jakéhokoliv jedince z uvažovaného prostoru jedinců.



	A	B	C	D	E
Průměr zlepšení (v násobku)	3	3	1,8	3	4
Průměrná směrodatná odchylka	0,5	0,6	0,2	2	3
Průměr generací, kde docházelo ke zlepšení	5	8	9	9	11
Průměr nejlepšího výsledku	0,1	0,21	0,26	0,3	0,58

Tabulka 5.2: Tabulka ukazuje zprůměrování 30 testů. Řádek **průměr zlepšení** ukazuje, kolikanásobně se v průběhu optimalizace strategie zlepšily. **Průměr generací, kde docházelo ke zlepšení** průměrně ukazuje, kolik generací proběhlo, než vylepšení stagnovalo

### 5.3 Test finální optimalizace

V další části jsem provedl sérii testů 6.6.2 zaměřených na finální nastavení genetických algoritmů. Nastavení a popis jednotlivých operací je uveden v architektuře systému 4.2. Kapitola také obhajuje, proč některé varianty uvedené v 4.2 byly pozměněny do konečné podoby *testu 3*. *Test 2* bere v potaz obchodní pozice všech sloupců *bool*. V závěru popisu *testu 2* je vysvětleno, proč byla dále nasazena fitness funkce s výběrem pouze nejlepšího sloupce určujícího pozici.

Testování bylo obtížné vzhledem k výpočetní náročnosti, která je nevýhodou genetických algoritmů obecně. Ovšem systém by měl být schopný reagovat na změny trhu v rámci obchodovaného timeframu. Vzhledem k faktu, že systém testuji na měnovém páru EUR/USD v rámci patnáctiminutového intervalu, pak jsem určil, že adekvátní doba výpočtu bude v rámci hodin maximálně jednoho dne.

S nastavením finální fitness funkce a operací jsem v průběhu testování dosahoval lepších výsledků než v případě fitness funkce z prvního testu. Ale vzhledem k nedostatku výpočetního výkonu jsem byl nucený omezit počet generací na 50. Toto omezení však z 90% nemělo vliv na ukončení. Pokud se výsledek nezlepšil po 4 generace, pak byla optimalizace ukončena.

#### Test 2

- data - 15 minutové svíčky páru EURUSD
- fitness funkce - lokální maximum s omezením 15 min.
- počet spuštění pro každé nastavení - 30  
význam neterminálů je vysvětlen zde 2.1
- Sloupec A ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 100. Jsou použity operace **mutace** a **křížení**.
- Sloupec B ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Také jsou použity operace **mutace** a **křížení**.
- Sloupec C ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá pouze operaci mutace.
- Sloupec D ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 1000. Jsou použity operace **mutace** a **křížení**.

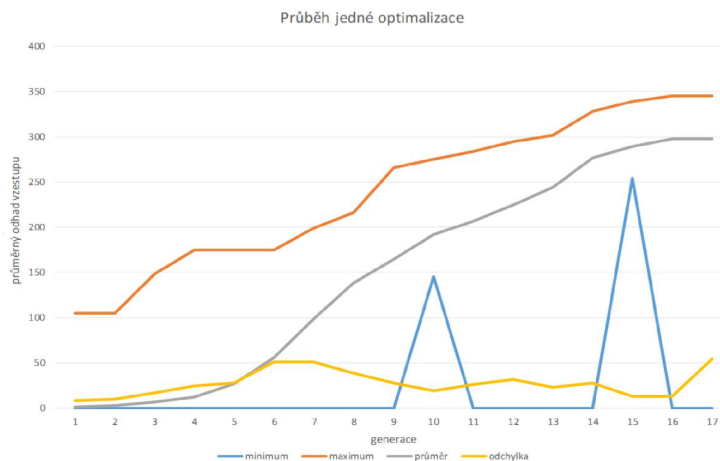
Z testů jsem zjistil, že velikost počáteční populace zvyšuje pravděpodobnost rychlejšího nalezení lepšího řešení. Operace křížení také celý proces urychluje, ale není nutná.

Tabulka 5.3: My caption

	A	B	C	D
Průměr zlepšení (v násobku)	8	9	5	3
Průměrná směrodatná odchylka	54	161	42	197
Průměr generací, kde docházelo ke zlepšení	25	28	16	31
Průměr nejlepšího výsledku	212	374	286	523

Tabulka 5.4: Tabulka ukazuje zprůměrování 30 testů. Řádek **průměr zlepšení** ukazuje, kolikanásobně se v průběhu optimalizace strategie zlepšily. **Průměr generací, kde docházelo ke zlepšení** průměrně ukazuje, kolik generací proběhlo, než vylepšení stagnovalo.

Na grafu jsem zobrazil průběh jednoho spuštění 5.2. Průběh ukazuje, že postupem evoluce se neustále přibližuje průměrný zisk a zisk maximální. Pokud se stále zlepšovala průměrná kvalita populace a pokud i odchylka byla relativně vysoká, pak průběh evoluce indikoval nalezení ještě lepšího jedince. Optimalizaci jsem ukončil ve chvíli, kdy rozdíl 4 předchozích maxim byl zanedbatelný. Tento test sice ukazoval velmi dobrý průměrný násobný výsledek, ale po důkladném prozkoumání, jak strategie vypadají, jsem zjistil, že závislost jednotlivých sloupců bool je minimální, a tak byl počet vygenerovaných sloupců úměrný zisku, což plně nereprezentovalo kvalitu daného výpočtu. Proto jsem v dalším testu upravil fitness funkci pro hodnocení strategie pouze podle nejlepšího sloupce bool. Tím je myšleno, že je vybrán pouze sloupec, který má nejlepší zisk. Z předešlého textu, konkrétně z kapitoly o vytváření vět 4.3 je vidět, že v rámci jedné strategie je vypočítáno více sloupců bool. Jde tedy o nezávislé vyhodnocení napříč sloupci bool.



Obrázek 5.2: Průběh optimalizace

## 5.4 Ohodnocení podle nejlepšího sloupce bool

Tento test prezentuje konečné nastavení. Komentuje jak průběh trénování, tak nasazení na testovacích datech.

### Test 3

- data - 15 minutové svíčky páru EURUSD
- fitness funkce - pevný *take profit* a *stop loss*.
- počet spuštění pro každé nastavení - 30  
význam neterminálů je vysvětlen zde [2.1](#)
- Sloupec A ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 100. Jsou použity operace **mutace** a **křížení**.
- Sloupec B ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Také jsou použity operace **mutace** a **křížení**.
- Sloupec C ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 500. Využívá pouze operaci mutace.
- Sloupec D ukazuje výsledky pro nastavení, kde je počáteční populace o velikosti 1000. Jsou použity operace **mutace** a **křížení**.

	A	B	C	D
Průměr zlepšení (v násobku)	2	4	3	3
Průměrná směrodatná odchylka	5	11	4	13
Průměr generací, kde docházelo ke zlepšení	19	24	18	25
Průměr nejlepšího výsledku	20	26	17	37

Tabulka 5.5: Tabulka ukazuje zprůměrování 30 testů. Řádek **Průměr zlepšení** ukazuje, koliknásobně se v průběhu optimalizace strategie zlepšily. **Průměr generací, kde docházelo ke zlepšení** průměrně ukazuje, kolik generací proběhlo, než vylepšení stagnovalo

Na vyhodnocení je vidět, že výsledky nejsou zkreslené počtem sloupců bool v jedné strategii. Z předešlého textu konkrétně z kapitoly o vytváření vět [4.3](#) je vidět, že v rámci jedné strategie je vypočítáno více sloupců bool. Jde tedy o nezávislé vyhodnocení napříč sloupci bool. V tomto testu je však započítán pouze sloupec operace bool s nejlepším výsledkem. Tedy násobek zlepšení není tak vysoký jako v testu 2. Test přinesl pozitivní výsledky, že při trénování je možné najít ziskové strategie. Vzhledem k tomu, že toto je finální test, tak dále přikládám výsledek poslední optimalizace. Nejlepší průměrný výsledek dosahoval průměrného zisku 20 bodů. Průměrný zisk populace byl 2 body. V další části je výsledek této optimalizace na testovacích datech, který pokračuje testem trénování neuronové sítě, a opětovné vyhodnocení na testovacích datech. Důležité je zmínit nasazení na reálných datech. Test byl vždy proveden na následujícím období, které bylo poloviční oproti trénovací sadě. Optimalizace byla z 70% procent přetrénovaná a v testu převažovaly ztráty. Proto sem provedl další test, ve kterém jsem přidělil jistá testovací data na validaci a testoval, zda strategie bude výdělečná, pokud bude zvalidována. Pokud strategie prošla validací, pak byla z 85% procent zisková i na testovací sadě. Zisk se lišil průměrně o 20% od zisku při trénování.

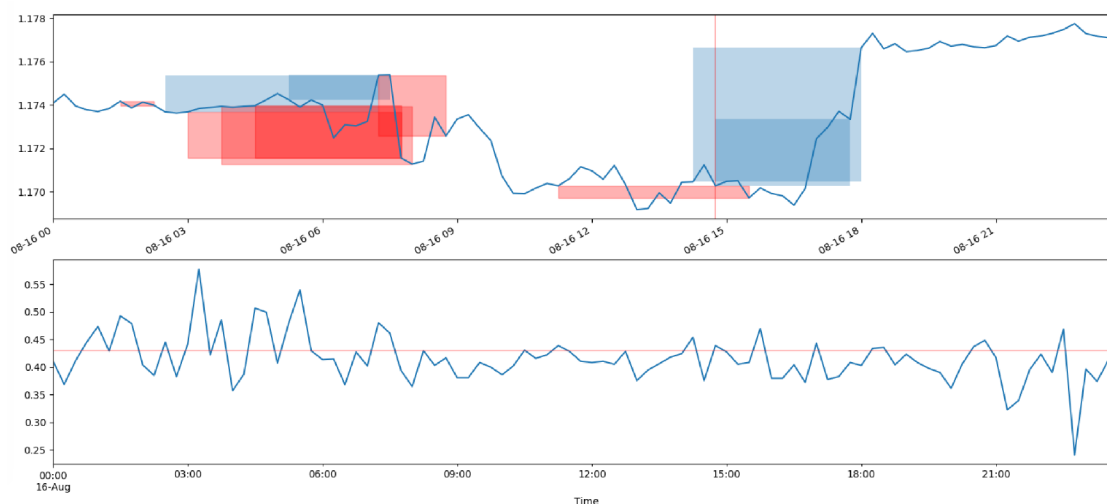
## 5.5 Nasazení strategie na testovacích datech

Provedl jsem optimalizaci s nejlepší strategií o průměrném výdělku 20 bodů. Do výsledku je nutné započítat náklady na otevření pozice pro zcela přesnou představu o tom, zda je možné uvažovat o nasazení strategie. Další sledované hodnoty na trénovacích datech také vykazovaly dobrý výsledek. Počet ztrátových pozic byl 186 a počet ziskových 328 s celkovým ziskem 5163 bodů. Velikost trénovacích dat byla 1418, což jsou všechna možná data získaná pro tento test. Následně byla strategie nasazena na testovacích datech, kde bylo simulováno reálné nasazení. Testovací sada měla velikost 680. Průměrný zisk činil 11 bodů a celkový zisk byl 1131 bodů. Ztrátových pozic bylo pouze 67 oproti ziskovým, kterých bylo 130. Strategie obsahovala následující výpočet:

$$\begin{aligned} & \text{Momentum} - 3 - \text{STO\_DDGLPN} | \text{srcnorm} | \text{sub} - \text{High}, \text{Volume} - \text{sub\_VCQSGG} | \\ & \text{sub} - \text{Close}, \text{Open} - \text{sub\_SVCZSP} | \text{sub} - \text{High}, \text{Close} - \text{sub\_EPINNX} | \\ & \text{lag} - \text{Low} - 84 - \text{lag\_WKLQJK} | \text{lag} - \text{High} - 161 - \text{lag\_HKKZFS} | \\ & \text{lag} - \text{Close} - 53 - \text{lag\_RDEJTM} | \text{windowmin} - \text{STO\_DDGLPN} - 25 - \text{windowmax\_WSGAXA} | \\ & \text{bool} - \text{sub\_SVCZSP} - 0.436 - \text{bool\_KIWZJJ} \end{aligned}$$

Obrázek 5.3: Předpis výpočtu

Odečtením hodnot ze sloupce close a open dostáváme ziskovou strategii při vstupu na hladině 0.43. Pro představu, jak vypočítaná data vypadají, jsem přiložil obrázek 5.4. Obchod je spuštěn, pokud je hodnota překročena. Zobrazeny jsou jen obchody, které jsou v zobrazeném rozmezí ukončeny. V případě neukončené pozice systém započítá stav na posledním zkoumaném indexu. Obchody po čase vyznačeném červenou vertikální čarou nejsou ukončeny, proto na grafu zobrazeny nejsou.



Obrázek 5.4: Průběh optimalizace

## 5.6 Nasazení neuronové sítě

Při testování ANN jsem sledoval různorodost chování a profit s nasazením tohoto modulu. Jako první krok jsem zaznamenal průběh trénování pro různé strategie. Každou strategii

jsem trénoval pětkrát, abych zjistil závislost na náhodné inicializaci dat. V rámci testování trénoval jsem se také snažil určit ideální nastavení délky vektorů do historie, jak je popsáno v návrhu 4.7. Také jsem testoval parametr *learning rate* a *batch size*. První tabulka na obrázku 5.6 ukazuje počet dosažených epoch u každé strategie s nastavením historického okna na velikost 200, s *learning rate* na 0.001 a *batch size* na 50. Trénování jsem provedl na stejně velké datové sadě, jako byla stanovena pro vstup GA pro nalezení strategií, tedy 1500.

Dále jsem zkoušel nastavit *batch size* na 1, kde se mi potvrdilo, že trénovalí trvá nepřiměřeně dlouho s nulovým posunem k lepší generalizaci problému. Ani změna *learning rate* nepřinesla významnou změnu. Zkoušel jsem hodnoty 0.1 a 0.01.

id strategie	Generace 1	počet epoch 1	počet epoch 2	počet epoch 3	počet epoch 4	počet epoch 5
1	3	2	2	2	2	
2	4	4	4	3	4	
3	4	5	3	4	4	
4	32	4	3	20	2	
5	2	2	2	2	2	
6	45	89	112	98	12	
7	4	3	4	4	4	
8	2	2	2	3	2	
9	2	2	2	2	2	
10	3	2	2	3	3	
11	7	2	21	5	6	
12	2	6	2	2	2	
13	8	4	8	2	2	
14	61	78	101	164	17	
15	2	2	2	2	2	
16	3	3	3	3	5	
17	2	2	2	2	6	
18	3	2	3	2	2	
19	6	2	6	5	2	
20	2	3	2	2	3	

Tabulka 5.6: Záznam o počtu dosažených epoch v průběhu trénovalí na vybrané sadě dvaceti jedinců

Výsledek jsem dále rozebral a podíval jsem se na charakter výpočtu u strategie (jedince) s id 6 a 14. Strategie 6. vypočítala 4 různě nastavené klouzavé průměry a strategie 14. využívala výpočet 2 klouzavých průměrů a funkce diference. Také mají malý interval mezi pozicemi, neboť strategie 6. měla velikost trénovalí sady 261 a strategie 14. trénovala s 129 vektory.

V druhém kroku trénovalí jsem vyhodnotil nasazení na testovacích datech v porovnání se strategií. Jak je vidět z tabulky na obrázku 5.7, neuronová síť problém příliš negeneralizovala. Ve většině případů vyhodnotila vše jako ztrátové a pozici neotevřela.

id strategie	počet ziskových pozic	počet ziskových pozic s ANN	počet ztrátových pozic	počet ztrátových pozic s ANN
1	2	0	1	0
2	18	18	15	15
3	1	0	1	0
4	6	0	7	0
5	5	0	7	0
6	18	17	8	8
7	5	0	3	0
8	1	0	0	0
9	2	0	3	0
10	7	0	5	
11	1	0	2	0
12	3	3	5	5
13	1	0	1	0
14	9	0	7	0
15	5	0	6	0
16	7	2	7	0
17	2	0	2	0
18	7	7	3	3
19	11	0	10	0
20	3	0	1	0

Tabulka 5.7: Porovnání pozic na testovacích datech o délce 150.

V celkovém zhodnocení testů 5.6.1 se snažím přístup vylepšit. Zejména vyřešit problém s nedostatkem dat pro trénování a přidat způsoby zpracování dat.

### 5.6.1 Vyhodnocení testů pro návrh finálního systému

Tato kapitola zhodnocuje celkové testování a vlastnosti systému. I přesto, že testování první verze nedopadlo špatně, tak jsou možnosti, jak systém zlepšit a odstranit chování, které systém zpomaluje a silně zhoršuje zejména průběh nalézání strategií. Tato kapitola kriticky zhodnocuje chování systému vzhledem k předešlým testům. Shrnutí chyb a nedostatků nasazení první verze jsem se snažil seskupit v této kapitole.

První chybou, kterou jsem podchytil v další verzi, je generování věty a následný překlad na předpis výpočtu. Myšlenka parametrizace funkcí přímo v překladači byla zbytečná a celkově přispěla k menší přehlednosti architektury a větší složitosti dalšího vývoje. Proto přesunu celou logiku generování předpisů výpočtu přímo do generátoru a budu generovat předpisy výpočtu hned v inicializaci systému. Tedy i parametry systému ovlivňující generování funkcí pro výpočet matice budou přímo v generátoru vět. Tento krok zpřehlední a zrychlí proces nalézání ziskových strategií, avšak vnese do systému limitace. Systém bude předpokládat několik vlastností. Generátor vět bude generovat parametr, tedy přijdeme o možnost dynamického určování parametrů. Princip generování je vysvětlen v kapitole 4.

Nedostatkem, který byl příčinou dalších kroků pro vylepšení je především častý výskyt odchylek. Podívejme se na obrázek 5.2. Velký výkyv minima indikuje, že systém nepracuje zcela správně. V systému první verze je implementována kontrola, zda ve *finální matici* není hodnota "nedefinováno". Bohužel většina implementovaných funkcí systému nepomohla spíše jej zpomalila. Systém pracuje nad reálnými čísly a tak se často v průběhu výpočtu

musí vypořádat s potenciálními problémy jako je nemožnost dělení nulou, dělení a násobení extrémními čísly, celkové extrémy v časových řadách způsobené aplikací funkcí jako je například *log* nebo *exp* a mnoho dalších problémů, které přinášejí výpočty nad reálnými čísly. Pokud řádek matice obsahuje "nedefinováno", pak je smazán. Systém dále ošetřuje i velikost matice. Provedl jsem tedy test, ve kterém jsem sledoval, kolik matic zůstane validních. Výsledkem bylo, že **průměrně 97 % vypočítaných neprošlo**. Odpovědí na tento test je také nová architektura.

Také zásadní zpomalení, a tedy i znemožnění pracovat na malých timeframech, je generování více sloupců *bool* a jejich následné procházení s tím, že je vybrán nejlepší. Přesně toto chování by mělo zůstat na *genetických algoritmech*.

V uvedených testech pouze ověřuji, že daná architektura jde správným směrem. Vylučuji některé cesty, ale testování samotného reálného nasazení je v kapitole 5.5 je pouze na jednom časovém úseku. V průběhu testování bylo vyhodnoceno více úseků. V další části uvedu více testů v různých časových úsecích. Účelem systému je jeho kontinuální vyhodnocování aktuálního stavu s určitým ohledem do historie. Testy druhého systému budou jednak ve více časových oknech, ale také se pokusím najít způsob, jak lépe simulovat prostředí reálného nasazení, na ověření výsledků se pokusím využít Metatrader4 Backtesting s tickovými daty.

Poslední neprobranou částí je modul neuronové sítě. Snahou je co nejobecnější napojení na logiku zbytku systému. Ovšem napojení ukázalo, že s aktuálním přístupem nejsem schopný získat dostatek dat pro trénink neuronové sítě, a z prvních testů se nezdá, že by napojení bylo velkým přínosem. Vzhledem k tomu, že s omezeným časem na práci nemám prostředky pro širší implementaci ANN, jsem tento přístup vyloučil a ve finální implementaci se budu o to více věnovat systému bez ANN.

Také systém upravím pro vypnutí normalizace. I přesto, že zavedení normalizace ulehčilo náročnou implementaci, na druhou stranu ztížilo vyhodnocování strategie pokud bylo překročeno minimum nebo maximum aplikované normalizace, což je u nestacionárního signálu na krátkém úseku normalizace časté.

## Kapitola 6

# Finální implementace

Kapitola vychází z předchozích kroků a finální návrh opravuje nalezené nedostatky a chyby. Jako shrnutí cíle a předchozích kroků uvedu několik vět.

Jde o dva pohledy, které určily směr a charakter práce, a to z hlediska strojového učení, kde práci nejvíce inspirovalo využití genetických algoritmů 3.2 a obecně principů strojového učení [1] spolu s možností využití gramatik [5], a dále z pohledu principu Forexu a fungování obchodníků při technické analýze. Z literatury 2 a z praxe vyplývá, že trh se neustále mění a obchodní strategie je třeba adaptovat za účelem co nejvyšších zisků. V případě dlouhodobých, ne tak ziskových strategií, je třeba zase obstát psychicky.

Vytvořený systém tedy využívá strojové učení za účelem automatizace technické analýzy a adaptace na změny charakteru signálu v určitém časovém intervalu.

Při testování je simulována adaptace na různé časové období a dodržení principů testování, kde časový úsek rozdělíme na trénovací časový úsek a následnou validaci a simulaci automatických obchodů na testovacím úseku.

### 6.1 Úvod

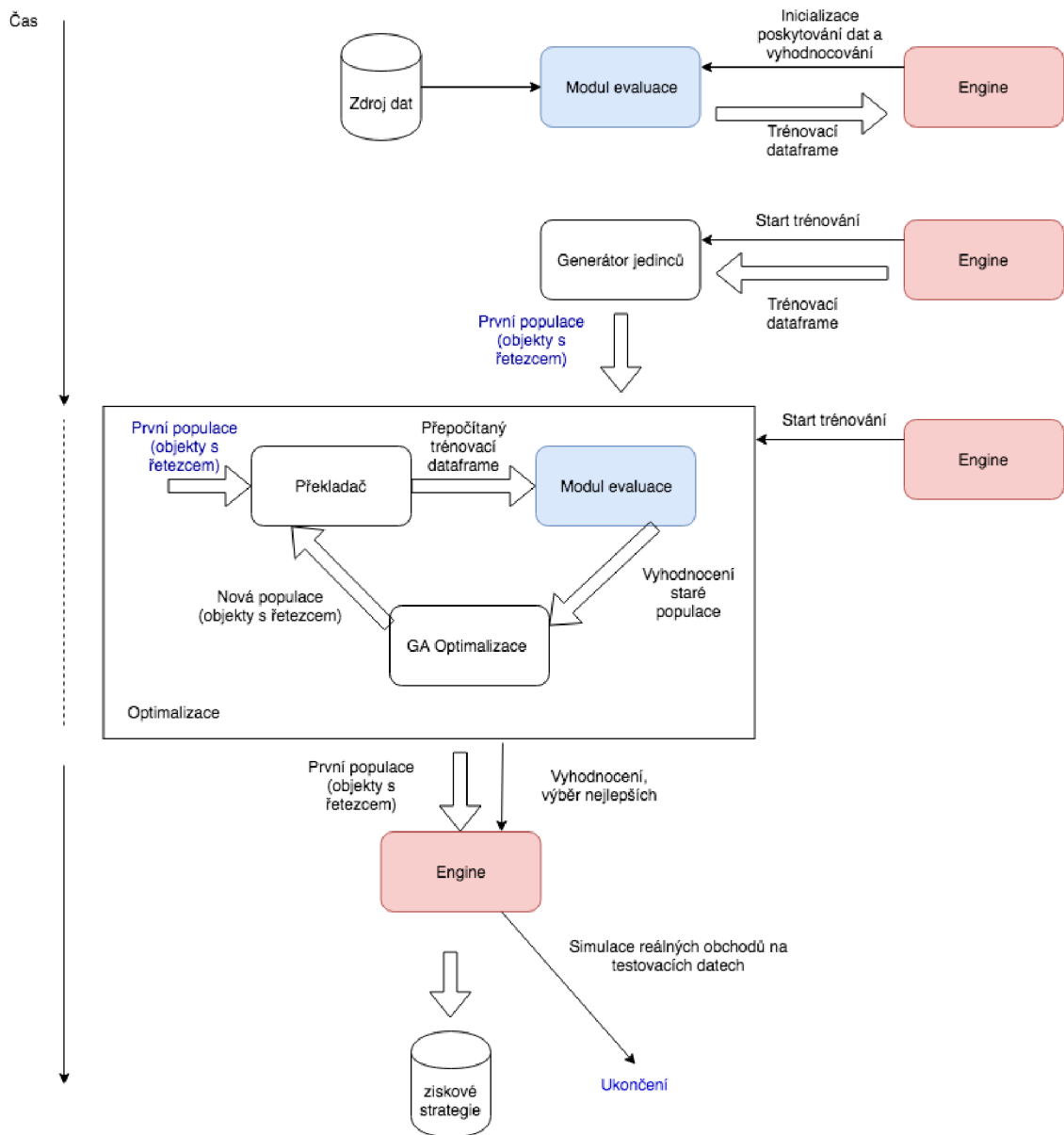
Fungování systému bylo pozměněno od předchozí verze. Důležitými změnami jsou především:

- Generování prvního jedince respektive jeho *computeString* (předpisu výpočtu). Z principu genetických algoritmů jsem pochopil, že strategie interpretace jedinců a logika generování první populace může mít velký vliv na výsledek. Logika generování je popsána v sekci 6.2. Díky efektivnějšímu výpočtu jedince se zmenšila celková paměťová náročnost.
- Odstranění normalizace, které mělo za následek složitější a novou implementaci logiky, ale ulehčilo simulaci obchodů a umožnilo udržení celkové variance dat mimo zpracovávaný interval, a tedy i možnost lehčího nasazení výpočetního předpisu na novém časovém úseku bez nutnosti řešení jiného minima a maxima v propočítávaném časovém úseku. Klady a zápory normalizace jsou popsány v sekci 6.5.
- Odstranění prvního překladače, který překládal vygenerovanou větu na *computeString*.
- Odstranění výpočtů, které měly za následek časté nedefinované hodnoty. Zkorigování výpočtů vylepšilo systém přibližně o 90 %
- Vylepšení optimalizace o možnost změny *stoploss* a *takeprofit*.



System je založen na jádru (engine) jako jednoduché nadstavbě, která určuje způsob inicializace, spuštění optimalizace a testování. Jádro inicializuje a předá objekt evaluace, který sestaví, drží a poskytuje data pro trénování a testování. Důležitou funkcionalitou je inicializace objektu jedince, která nastaví Předpis výpočtu. Dále princip překladu, který drží překladač a je využíván při inicializaci objektu jedince a vyhodnocení jedince po operaci křížení a mutace (optimalizaci).

Vzhledem k charakteru systému, který funguje v čase sériově, jsem uvedl časový diagram 6.1, kde jsou jednotlivé kroky a moduly zobrazeny. Prvně je inicializován objekt pro evaluaci, který načte a nastaví data pro trénování, testování a vyhodnocování v průběhu optimalizace. Následně je vygenerována populace jako inicializace N objektů s logikou jedince. Populace je předána genetickým algoritmům, které aplikují své principy nad **Předpisem výpočtu** a pro evaluaci využívají překladač spolu s evaluačním objektem. Po dokončení optimalizace je aplikována validace N nejlepších jedinců a simulace jejich případného nasazení na trh.



Obrázek 6.1: Diagram systému v režimu trénování a simulace obchodů

## 6.2 Vytvoření jedince v populaci

Při inicializaci dojde k sestavení **předpisu výpočtu**. Oproti předchozí verzi, kde byl zvlášť generátor vět a následný překlad na **předpis výpočtu**, je veškerá logika přímo v inicializaci. Třída jedince drží předpis gramatiky spolu s kompletní logikou pro převod věty s postupem aplikovaných funkcí na **předpis výpočtu**. Vytvoření probíhá v těchto krocích:

1. Vygenerování věty gramatikou
2. Iterativní průchod větou, ve které jsou příkazy pro použití konkrétních funkcí a vytváření předpisu výpočtu. Popis algoritmu je uveden dále.

3. Přidání speciálního nastavení jako sufix předpisu výpočtu.
4. Uložení výsledného předpisu výpočtu a zahazení vypočítané matice.

Bod dva funguje složitěji, zejména z důvodu odstranění normalizace. Systém již nemůže vědět, že obor hodnot signálů (sloupců v matici) je mezi hodnotami 0 až 1. Je tedy nutné simulovat výpočet matice jako držení stavu a určování parametrů pro nově přidané funkce.

Efektivnější využití vypočítaných sloupců je implementováno využitím držení informace o tom, které sloupce nebyly použity pro další výpočet, a na konci řetězce je určen sloupec s nejdelší historií výpočtu pro vstup do pozice. Algoritmus sestavení předpisu výpočtu pracuje zjednodušeně takto:

1. Inicializuj seznam nevyužitých sloupců jako prázdný. Inicializuj matici s počátečními signály (získej od evaluačního modulu) a procházej v cyklu větu s daným pořadím funkcí, které mají být použity.
2. Urči název funkce spolu s informací, kolik přijímá parametrů a zda jde o konstantu nebo sloupec z matice.
3. Pokud jde o sloupce, tak nejdříve vyber z nepoužitých, jinak jakékoliv v matici. Pokud byl vybrán z použitých, tak jej odstraň.
4. Vypočítej funkci s přiřazenými parametry. Parametry urči z matice.
5. Výsledek ulož jako sloupec do matice a název nového sloupce přidej do nevyužitých sloupců.
6. Proveď kontrolu nedefinovaných hodnot **6.3**
7. Pokud jde o konečný příkaz, tak vyber sloupec z nevyužitých a cyklus ukonči.

Výpočty funkcí jsou stejné jako v případě překladu. Dále uvedu, jak je inicializace nastavena spolu s příkladem předpisu výpočtu. Limit pro využití historie je nastaven na 2 až 200. Jde o hodnotu, jak maximálně velké okno může být pro výpočet technických indikátorů nebo klouzavých oken. Konečná gramatika byla obohacena o neterminál **j** pro `windowmean` jako výpočet průměru v klouzavém okně a s **u** pro `submean` jako odečtení průměru od signálu a vypadá takto:

```
S -> iS | iA
A -> nB
B -> sB | C
C -> fC | D
D -> xD | wD | jD | E
E -> uE | F
F -> bC | k
```

Takováto gramatika spolu s překladem může vygenerovat například následující předpis výpočtu: `/ROC=178=1/srcnorm/sub=1,Close=3/sub=3,Close=4/sub=4,3=5 /bool=5=0.000544=bool/takeprofit,220=stoploss,240`

Vysvětlení je následující: vygeneruj indikátor ROC v okně 178. Následně je klíčové slovo `srcnorm`, které slouží jen jako bod křížení pro GA. Funkce `sub` přijímá nevyužitý sloupec 1 a vstupní sloupec `Close`. Nový sloupec nazývá číslem 3. Opět `sub` s nevyužitým sloupcem 3. Podobně až k funkci `bool`, která určí vstup do pozice jako překročení hladiny 0.000544. Hladina je vyvozena z limitů signálu 5 v matici. Sufix `takeprofit,220=stoploss,240` je speciálním nastavením pro parametrizaci pozice.

## 6.3 Překlad a evaluace

Tyto funkce jsem se rozhodl uvést společně, neboť je spojují problémy, které jsem musel řešit. Tedy kromě popisu jejich funkce a návaznosti popíší i propojené problémy, na jejichž řešení spočívá kvalita celého systému.

**Překladač** funguje velmi přímočaře. Vstupem do překladače je jedinec s předpisem výpočtu obohacen o vypočítanou matici. Jde o stejný princip výpočtu funkcí jako při inicializaci jedince s rozdílem, že hodnoty parametrů jsou brány přímo z předpisu výpočtu.

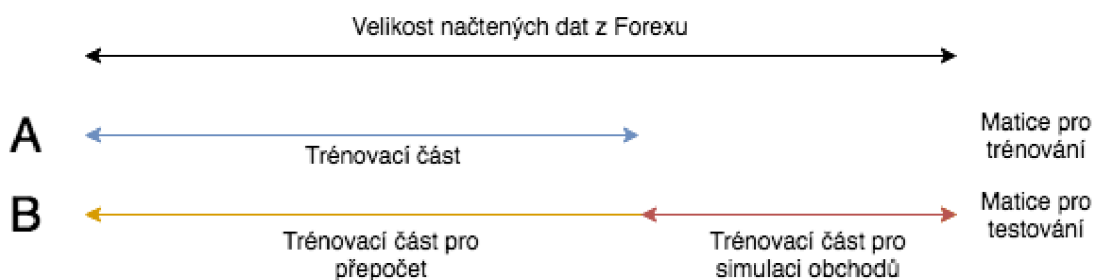
**Překladač** je navázán na modul evaluace. Pokud máme nový (například zmutovaný) předpis výpočtu, tak musíme vše přepočítat a vyhodnotit s nově vypočítanou maticí. Funkce evaluace jedince tedy využívá překladač.

Při překladu dochází k situaci, kdy výsledkem některých funkcí je na prvních  $N$  indexech nedefinovaná hodnota. Jde však o jinou situaci než problém s častými nedefinovanými hodnotami na různých indexech z důvodu matematických operací s extrémními hodnotami (zásadní problém z předchozí verze), které mají za následek nedefinované hodnoty jako důsledek nepřesnosti knihoven a principů práce s reálnými čísly. Zde jde o situaci, kdy proces překladu aplikuje jednu z funkcí, která využívá hodnoty z minulosti. Pokud funkce využije v čase  $t$  hodnoty  $t - n$  až  $t$ , tak není možné spočítat hodnoty menší než 0 až  $n$ . Prvních  $n$  hodnot je nedefinovaných.

Proces překladu funguje tak, že rozdělí předpis výpočtu na jednotlivé operace rozdělené znakem `|` a prochází jednotlivé operace v cyklu.

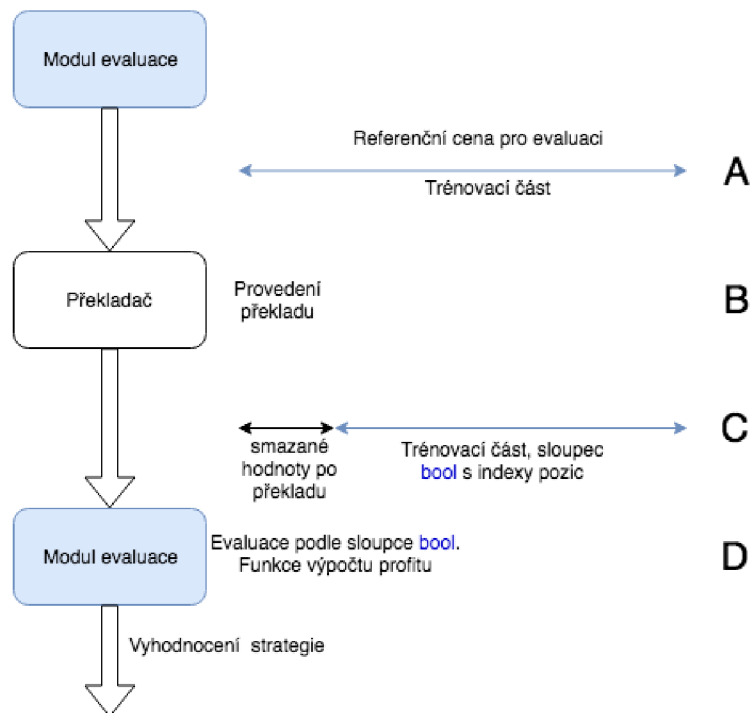
1. Inicializuj matici s počátečními signály (získej od evaluačního modulu). V cyklu procházej vytvořený seznam operací až po poslední, kterou je operace `bool`.
2. Vezmi operaci v pořadí.
3. Operaci ve formátu `funkce=parametr,parametr=název` nebo `funkce=parametr=název` a v případě číselných parametrů pak `funkce=parametr=číslo=název` zpracuj a aplikuj výpočet podle funkce a daných parametrů.
4. Podle názvu výsledek ulož jako sloupec do matice.
5. Proveď kontrolu nedefinovaných hodnot. Odstraň nedefinované hodnoty na začátku matice
6. Přejdi na krok 2, pokud je další operace v pořadí, jinak skonči.

Z předešlého textu vyplývá, nejen že matice bude procesem překladu zkrácena, ale také, a to i s ohledem i na předchozí kapitoly, že evaluační objekt využívají objekty jedinců a překladač. Evaluační modul je inicializován při spuštění systému pouze jednou a drží potřebná data pro trénování a testování. Při inicializaci vytvoří dvě matice. Na obrázku 6.6 jsou dvě symbolické délky matice A a B v porovnání k černě vyznačené celkové délce forexových dat na vstupu. Matice A určena pro trénování, tak aby nebyla k dispozici testovací data v procesu trénování a matice B pro testování, která drží historická data pro kontrolu, že byl vykonán stejný výpočet a pro simulaci kontinuálního přepočtu s nově přidanými daty při simulaci nad testovacími daty.



Obrázek 6.2: Porovnání vytvořených matic při inicializaci evaluačního objektu. Matice A je určena pro trénování, tak aby nebyla k dispozici testovací data a matice B pro testování, která drží historická data pro kontrolu, že byl vykonán stejný výpočet a pro simulaci kontinuálního přepočtu s nově přidávanými daty při simulaci nad testovacími daty

Dále je diagram 6.3 ukazující proces předání dat z evaluačního objektu překladači a následné předání sloupce (časové řady) bool. Jak je pravděpodobně patrné již z předešlého textu, tak v bodě A je předána matice se sloupci *Open*, *High*, *Low*, *Close*, *Volume*, což jsou signály s burzovními daty. Na diagramu jde zejména o vyznačení jejich relativní délky. V části B je proveden překlad, tak jak je vysvětlen výše a matice je obohacena o nové sloupce. Konečným sloupem je sloupec bool. V části C je vyznačeno příkladové zkrácení matice v důsledku výpočtu okenních funkcí. Část D je vysvětlena níže, neboť pro její vysvětlení a popis formátu jsou použity opět grafické prostředky.



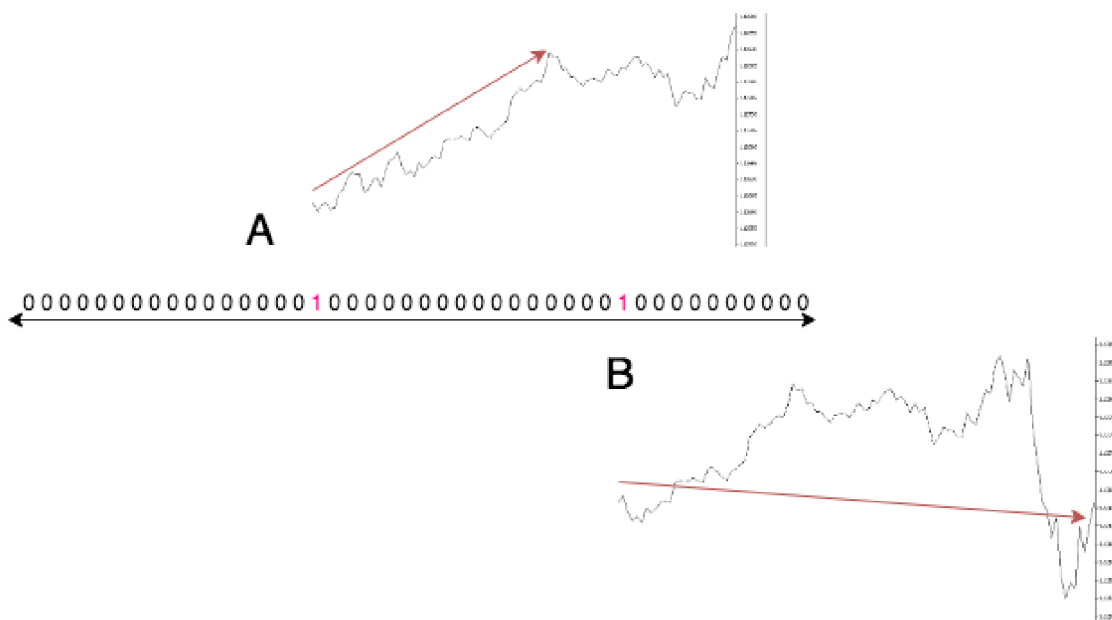
Obrázek 6.3: Znázornění překladu. Ukazuje, v jaké části dojde ke zmenšení matice. Jednotlivé části A, B, C a D jsou podrobně popsány v textu. A zobrazuje relativní délku předaných dat. B zobrazuje překlad. C popisuje, že po překladu dojde ke smazání počátečních hodnot a předání sloupce bool. D značí čas, ve kterém je volána funkce objektu evaluace, která vrací přímo zhodnocení strategie.

Pro přesnější pochopení systému rozeberu podrobněji sekci D. I přesto, že formát a způsob výpočtu sloupce bool by měl být jasný z předešlého textu. Jeho výpočet je následující:

1. Vyber sloupec z matice podle názvu. Název je parametr funkce (viz příklad: `bool=5=0.000544=bool`, kde je vybrán sloupec 5)
2. Vytvoř signál o velikosti matice, který drží všechny hodnoty 0 (jako false)
3. Na indexy, kde zkoumaný signál kříží ze zdola nahoru hodnotu drženou v parametru pro konstanty (v případě příkladu jde o hodnotu 0.000544), ulož hodnotu 1 (jako true) do vytvořeného signálu.
4. Nový signál s logickým označením indexů, kde dojde ke vstupu do pozice, ulož do matice jako nový sloupec pod názvem **bool**.

Pro funkci ohodnocující strategii je podstatný právě tento konečný signál bool. Jde o signál s nulami a jedničkami. Jde o stejný přístup jako v předchozí verzi systému. *Hodnotící funkce* přijímá kromě bool také signál *Close* z uložené matice pro testování. Prvním krokem je zarovnání indexů a useknutí prvních hodnot o počet stejný jako délka chybějících dat z matice vypočítané překladem. Důležitost tohoto kroku je zřejmá z důsledku jeho vynechání. Následně jsou vyhodnoceny všechny indexy označené hodnotou 1. Podle indexu je vybrána počáteční hodnota z časové řady *Close* okna jako zóny pro vyhodnocení pozice k danému

indexu. V systému je proměnná, která určuje délku po jakou je pozice zkoumána. Na obrázku je vyznačen princip vyhodnocení. Příklad obrázku vychází z délky pro vyhodnocení 100. V případě A jde o úspěšné dosažení takeprofitu nastaveného na 200 bodů. V případě B jde o případ, kdy nebylo dosaženo takeprofitu ani stoploss a pozice je ukončena s koncem pozorovacího období.



Obrázek 6.4: Znázornění překlada. Ukazuje, v jaké části dojde ke zmenšení matice. Jednotlivé části A, B, C a D jsou podrobně popsány v textu. A zobrazuje relativní délku předaných dat. B zobrazuje překlad. C popisuje, že po překlada dojde ke smazání počátečních hodnot a předání sloupce bool. D značí čas, ve kterém je volána funkce objektu evaluace, která vrací přímo zhodnocení strategie.

Jako shrnutí je tedy důležité pochopit, že při inicializaci objektu evaluace, jsou vytvořeny zvláště matice pro trénink a testování. Následně jsou podle potřeby předávány a využívány funkcí pro evaluaci strategie s ohledem na to, zda jde o test nebo trénink. V případě testování byla **provedena kontrola**, zda jsou stejné výpočty v časové části určené pro trénink. Šlo tedy o test, že budoucí charakter signálu nemění výsledek výpočtu minulého. S výsledky okenních funkcí je nakládáno až na prvním indexu po zóně, kde je okenní funkce aplikována. Pro pochopení je také důležité, že pro vyhodnocení je nutné strategii přeložit a využít funkce evaluace pro vyhodnocení sloupce bool a tím získat data vypovídající o zisku strategie za období, na kterém je matice vypočítána.

## 6.4 Optimalizace strategií

Zde je podstatná změna od předešlého systému, zejména snížení celkové náročnosti a zlepšení efektivity díky lépe postavenému *předpisu výpočtu*, neboť operace mutace a křížení jsou prováděny nad menším řetězcem a jejich aplikace má větší vliv na výsledek.

Nastavení genetických algoritmů zůstalo podobné. Pro funkci selekce využívám Tournament, který je přímo popsán v literaturách a implementován v knihovně DEAP pro genetické algoritmy (GA).

Pro funkci **křížení** zůstal princip podobný, kde klíčové slovo *srcnorm* již neznamená provedení normalizace, ale stále jde o bod křížení. Vzhledem k tomu, že poloha bodu křížení může být i zde jiná u různých jedinců, tak metoda křížení nemusí předat všechny vlastnosti. Zásadním omezením je logická vazba mezi operacemi. Není možné jednoduše předat parametry bez ověření nad vypočítanou maticí.

Jednotlivé části *Předpisu výpočtu* rozdělené znakem | jsou *operace*. Jak je zřejmé z popisu systému, je *Předpis výpočtu* rozdělen na tři části. Část před klíčovým slovem *srcnorm* představuje jednotlivé technické indikátory. Část druhá jsou další matematické operace a část třetí je nastavení pozice. Při křížení je možné vyměnit operace z první části a parametry nastavení z části třetí. Mohou nastat tyto varianty:

1. Pokud je počet operací (technických indikátorů) menší u prvního jedince než u druhého, pak vyměň pouze počet operací menšího jedince od začátku. Dále vyměň parametry nastavení.
2. Pokud je první část u obou jedinců stejná, tak dochází k výměně všech operací a výměně nastavení pozic.

Zde je jednoduchý příklad křížení, na kterém je vidět změna velikosti se zavedením lepšího algoritmu pro generování *předpisu výpočtu*, ale také nemožnost jejich jednoduchého předání druhému jedinci. Jedinec A má následující předpis výpočtu:

```
|ROC=62=1|srcnorm|sub=1,Close=3|sub=3,Close=4|sub=4,3=5|bool=5=0.000544=
  bool|takeprofit,190=stoploss,240}
```

Jedinec B potom:

```
|MA=178=1|MA=20=2|srcnorm|sub=1,2=3|bool=3=0.000544=bool|
takeprofit,200=stoploss,200
```

Výsledek křížení pro jedince A je následující:

```
|MA=178=1|srcnorm|sub=1,Close=3|sub=3,Close=4|sub=4,3=5
|bool=5=0.000544=bool|takeprofit,200=stoploss,200
```

Pro jedince B je výsledek:

```
|ROC=62=1|MA=20=2|srcnorm|sub=1,2=3|bool=3=0.000544=bool|
takeprofit,190=stoploss,240
```

Bohužel díky celkové návaznosti řetězce na jednotlivé operace jsem nedokázal navrhnout lepší způsob křížení. Možnosti, jak implementovat systém tak, aby bylo možné vytvořit lepší princip křížení, je uveden v závěru. Tato implementace křížení podle literatury nenarušuje princip GA. Při konzultaci s prof. Sekaninou mi také bylo řečeno, že křížení je možné vynechat a je důležité dobře implementovat mutaci, tak aby byla schopná projít celý prostor možností.

**Mutace** je také implementována podobně. Opět není možné měnit navázanou posloupnost využití sloupců, ale je možné mutovat číselné parametry funkcí, které přijímají stejné parametry a nastavení pozice.

1. Pro technické indikátory je možné mutovat z dané množiny dostupných indikátorů
2. Pro okenní funkce je možné vybrat z *windowmin*, *windowmax*, *windowmean*
3. Funkce *sub* již nemá společné funkce, viz kapitola 6. Funce *diff* s *submean* také ne.



Důležitou vlastností GA je jejich schopnost neuváznout v lokálním maximu. Proto jsem zavedl i možnost mutace celých funkcí. Ovšem pokud si to představíme jako průchod prostorem, jde o velký skok, který pravděpodobně celý výsledek zásadně změní. Proto při mutaci pravděpodobně dochází více ke změnám pouze těch parametrů, které zapříčiňují menší změny. Příkladem mutace může být tato změna:

```
|MA=178=1|MA=20=2|srcnorm|sub=1,2=3|bool=3=0.000544=bool|  
takeprofit,200=stoploss,200
```

```
|ROC=23=1|MA=20=2|srcnorm|sub=1,2=3|bool=3=0.000555=bool|  
takeprofit,210=stoploss,180
```

Změny parametrů jsou prováděny procentuálně, aby docházelo jak k malým, tak i velkým změnám. Touto kapitolou jsem se snažil co nejlépe popsat, jak optimalizace probíhá. Z pochopení by mělo být jasné, jaké bude mít vlastnosti a proč přidání nastavení je zajímavým vylepšením.

## 6.5 Závěr finální implementace

Ze systému byly odstraněny problémové operace. Díky tomu již tak často nedocházelo k výpočtu nedefinovaných hodnot. V předchozí implementaci byla znehodnocena většina výsledků. V aktuální verzi jsem systém pustil 10krát s velikostí populace 1000 a průměrně došlo k znehodnocení pouze 2 jedinců.

Stále může dojít k situaci, kdy jedinec bude ziskový, ale díky mutaci například funkce se stane neziskovým. Je důležité si uvědomit, že v populaci bude stále velké množství neziskových jedinců. Přidání možnosti optimalizace *stoploss* a *takeprofit* bylo provedeno zejména se záměrem vylepšení optimalizace.

I přesto, že se efektivita GA se může zdát malá, je tento přístup nejlepší i přes další možná řešení. Silnou stránkou systému je logika generování první generace a schopnost ziskové strategie nalézt a zaznamenat v přenositelném formátu. Díky tomu je možné strategii aplikovat přes různé platformy a systémy, což je důležité, neboť možnosti přístupu na trh se mohou měnit.

Posledním bodem pro zhodnocení je odstranění normalizace. I přes problémy, které odstranění přineslo z hlediska implementace, i přes znemožnění obecného využití normalizace jsem na úkor obecnosti získal výhodu, že není nutné kontrolovat změnu minima a maxima. Na obrázku je názorně ukázáno, že pokud bychom měli trénovací část od začátku až po čas 01:00, který je vyznačený červenou svislou čarou, pak v druhé části pro trénování bychom měli nové maximum a strategii by bylo nutné normalizovat s novými parametry pro normalizaci.



Obrázek 6.5: Ukázka problematiky často se měnícího lokálního maxima případně i minima

## 6.6 Testování

Jak bylo řečeno, systém je navržen pro kontinuální přetrénování v časovém okně v případě reálného nasazení. Proto i při testování budu simulovat 4 iterace, kdy se posunu v čase ovšem s větší časovou prodlevou, aby byl systém otestován na obdobích s různým charakterem. Tedy při různém trendu, při různé hladině klouzavého průměru a podobných vlastnostech.

Pro účely práce byla sehnána data ve formátu CSV od 20.12.2017 do 1.6.2018 pro pár EUR/USD, USD/CHF, USD/JPY. Na obrázku jsou vyznačeny jednotlivé části na kterých byl systém trénován a testován. Z hlediska strojového učení obecně se může zdát, že jde o málo dat, ale z hlediska této práce jde o jeden z požadavků, aby systém dokázal pracovat i s malým množstvím získaných dat. Také bylo žádoucí vytvořit systém, který by mohl fungovat bez nutnosti vlastnit výpočetní výkon nad možnosti průměrného uživatele. Systém byl testován na MacBook Pro (15-inch, 2016), 2,9 GHz Intel Core i7 a Intel HD Graphics 530 1536 MB.



Obrázek 6.6: Časové rozvržení jednotlivých testů

Každý ze 4 uvedených testů je spouštěn s velikostí populace 1000 a optimalizován po dobu 10 generací. Při testování byla navržena i metoda validace. Cílem bylo otestovat prvních N hodnot jako validačních a pokud bude výsledek pozitivní, pak pustit strategii na reálný trh, ale výsledky nebyly lepší než v případě okamžitého nasazení po natrénování. Je také důležité upozornit, že není brán ohled na prodlevu mezi tréninkem a nasazením. Trénování může trvat na různých počítačích různě dlouho a v práci je brán implicitně případ, kdy se systém stihne natrénovat do času prvního obchodu z množiny obchodů testování.

V rámci testování bylo zjištěno, že optimální je strategii aplikovat na prvních 5 obchodů a následně provést opětovné trénování. V každém testu je spuštěno maximálně 5 pozic. V kontrolních testech pomocí Metatrader jsem v příloze uvedl verze, kde je testováno pouze prvních 5, ale také všechny možné pozice za dané testovací období.

Kapitola 6.6.5 shrnuje testování na 8 různých obdobích pro 3 různé páry v podobě zisků. Kapitola testuje 3 strategie natrénované pouze na prvním testovacím období pro daný pár a aplikované i v jiných časech. Kapitola ověřuje nutnost adaptace.

### 6.6.1 Test 1

	Generace 1	Generace 2	Generace 3	Generace 4	Generace 5
Min	0.0	0.0	0.0	0.0	0.0
Max	75.7536	75.6462	75.5728	75.7536	71.78260
Avg	10.2249	17.2249	0.2249	15.2249	17.4589
Std	2.4459	2.4459	2.4459	2.4459	15.17616
	Generace 6	Generace 7	Generace 8	Generace 9	Generace 10
Min	0.0	0.0	0.0	0.0	0.0
Max	77.7536	101.7536	101.7536	103.93203	109.78260
Avg	39.2249	45.2249	50.2249	57.20096	57.74813
Std	14.4459	15.4459	14.4459	15.22802	12.89759

Tabulka 6.1:

Předpis výpočtu:

```
Donchian=100=1|Donchian=37=2|srcnorm|sub=2,1=4||diff=4=6|submean
=4=-3.9212573424356156e-07=9|submean=6=-4.406346343170172e-21=10|
submean=4=3.993251373497968e-21=11|bool=11=0.000518=bool|takeprofit
,170=stoploss,270
```

index	čas	zisk v bodech
6514	2018-01-23 08:20:00	90
6753	2018-01-24 04:15:00	211
6781	2018-01-24 06:35:00	-188
6866	2018-01-24 13:40:00	-191
6879	2018-01-24 14:45:00	221
Celkový zisk		143 bodů
Celkový zisk Metatrader		850 bodů

Tabulka 6.2: Tabulka shrnuje obchodní pozice vyhodnocené při testování pro test 1

## 6.6.2 Test 2

	Generace 1	Generace 2	Generace 3	Generace 4	Generace 5
Min	0.0	0.0	0.0	0.0	0.0
Max	54.9444	74.48214	54.9444	64.70731	83.78260
Avg	1.1532	1.8792	3.3912	6.3297	9.4549
Std	5.52560	7.0863	8.1739	10.4459	11.17216
	Generace 6	Generace 7	Generace 8	Generace 9	Generace 10
Min	0.0	0.0	0.0	0.0	0.0
Max	75.7346	75.7536	78.7536	78.9322	85.03636
Avg	18.2467	18.1738	21.7453	22.9752	25.16945
Std	12.7291	13.4459	13.4459	15.22802	15.89759

Tabulka 6.3:

Předpis výpočtu:

```
|Force=5=1|RSI=48=2|OBV=88=3|ROC=33=4|Momentum=15=5|OBV=40=6|STDDEV=41=7|
srcnorm|sub=3,5=9|sub=1,2=10|sub=4,9=11|sub=6,10=12|sub=12,7=13|sub
=13,11=14|sub=14,9=15|sub=15,11=16||windowmax=16=35=19||windowmean
=19=51=20||windowmin=20=19=21|submean=10=-2.4693154246637556=23|bool
=23=0.862715=bool|takeprofit,180=stoploss,240
```

index	čas	zisk v bodech
6704	2018-03-13 05:40:00	287
6706	2018-03-13 05:50:00	-309
6720	2018-03-13 07:00:00	213
6725	2018-03-13 07:25:00	278
6736	2018-03-13 08:20:00	-74
Celkový zisk		395 bodů
Celkový zisk Metatrader		540 bodů

Tabulka 6.4: Tabulka shrnuje obchodní pozice vyhodnocené při testování pro test 2

### 6.6.3 Test 3

	Generace 1	Generace 2	Generace 3	Generace 4	Generace 5
Min	0.0	0.0	0.0	0.0	0.0
Max	76.8945	76.8482	64.3214	75.9274	83.2898
Avg	1.4628	1.8999	5.8256	20.5734	26.3691
Std	3.7429	7.0972	8.7918	10.7731	11.1616
	Generace 6	Generace 7	Generace 8	Generace 9	Generace 10
Min	0.0	0.0	0.0	0.0	0.0
Max	75.7492	75.7492	84.5789	76.1947	76.9822
Avg	30.8466	24.1478	33.4285	36.8952	33.6912
Std	15.9189	15.8734	16.8062	16.3380	16.8794

Tabulka 6.5:

Předpis výpočtu:

```
|CCI=44=1|srcnorm|windowmean=1=67=5|bool=5=0.000320=bool|takeprofit,220=
stoploss,160
```

index	čas	zisk v bodech
6369	2018-04-18 20:20:00	-6
6486	2018-04-19 06:05:00	69
6537	2018-04-19 10:20:00	-181
6557	2018-04-19 12:00:00	77
6750	2018-04-20 04:05:00	-199
Celkový zisk		-240 bodů
Celkový zisk Metatrader		-136 bodů

Tabulka 6.6: Tabulka shrnuje obchodní pozice vyhodnocené při testování pro test 2

### 6.6.4 Test 4

	Generace 1	Generace 2	Generace 3	Generace 4	Generace 5
Min	0.0	0.0	0.0	0.0	0.0
Max	23.9444	31.48214	35.9444	62.70731	42.78260
Avg	0.1428	0.9837	0.3944	1.2349	7.9829
Std	1.0845	1.8425	4.2945	5.8173	5.7123
	Generace 6	Generace 7	Generace 8	Generace 9	Generace 10
Min	0.0	0.0	0.0	0.0	0.0
Max	39.7346	39.1438	43.8834	43.8834	46.03636
Avg	12.3479	11.1724	15.2955	21.4274	21.8951
Std	8.7291	12.4459	12.8301	11.0384	9.1834

Tabulka 6.7:

Předpis výpočtu:

```
EoM=36=1|Donchian=87=2|Momentum=85=3|Force=2=4|STDDEV=6=5|STO=3=6|srcnorm|
submean=2=0.006007763174767973=11|submean=3=-0.00012376777251184835=12|
bool=12=0.002472=bool|takeprofit,190=stoploss,190
```

index	čas	zisk v bodech
6072	2018-05-25 14:10:00	226
6080	2018-05-25 14:50:00	215
6082	2018-05-25 15:00:00	219
6087	2018-05-25 15:25:00	228
6094	2018-05-25 16:00:00	193
Celkový zisk		1 081 bodů
Celkový zisk Metatrader		552 bodů

Tabulka 6.8: Tabulka shrnuje obchodní pozice vyhodnocené při testování pro test 4

### 6.6.5 Přehled zisků na více párech

pár	01-23 - 01-30	03-13 - 03-20	04-18 - 04-26	05-25 - 05-31
EUR/USD	143	395	-240	1 081
USD/CHF	587	210	132	-300
USD/JPY	294	-620	431	62

Tabulka 6.9: Tabulka shrnuje zisky v uvedených datech za rok 2018 na třech různých párech

pár	01-23 - 01-30	03-06 - 03-13	04-11 - 04-18	05-18 - 05-25
EUR/USD	243	154	61	473
USD/CHF	108	368	-299	-185
USD/JPY	302	105	-212	336

Tabulka 6.10: Tabulka shrnuje zisky v uvedených datech za rok 2018 na třech různých párech

### 6.6.6 Test nutnosti adaptace

System je navržen tak, aby obstál při rychle se měnících vlastnostech signálu Forexu. Tento test je důkazem, že natrénovaná strategie v čase selhává. Tabulka 6.6.6 ukazuje selhání strategie mimo první sloupec, který je přímo za trénovacím intervalem.

pár	01-23 - 01-30	03-13 - 03-20	04-18 - 04-26	05-25 - 05-31
EUR/USD	143	-420	-240	-121
USD/CHF	587	-318	267	-482
USD/JPY	294	-394	-539	26

Tabulka 6.11: Tabulka shrnuje zisky v uvedených datech za rok 2018 na třech různých párech

# Kapitola 7

## Závěr

Hlavním cílem práce bylo vytvořit systém pro vytváření strategií a automatické obchodování. Vedlejším úmyslem bylo také vytvořit architekturu, která bude dostatečně obecná a robustní pro případné rozšíření o nové přístupy a heuristiky.

Hlavního cíle bylo dosaženo. Kromě vlastních testů, které sloužily i jako prostředí pro fitness funkci při optimalizaci a zhodnocení strategie při trénování, jsem také využil obecně uznávané testování strategií pomocí metatraderu. Při porovnání je nepochybně vidět, že testování implementované v práci má oproti variantě metatraderu nedostatky, ale je dostačující pro vytvoření ziskových strategií.

Reálné nasazení by bylo podobné simulaci zpracované v kapitole testování 6.6. Systém vytvořený v této práci by průběžně stahoval aktuální forexová data o předem dané velikosti a počítal nové strategie, které budou nasazeny na reálný trh na dobu určenou jako pravděpodobně ziskovou v poměru k velikosti historického intervalu určeného k trénování.

Testy ukazují, že implementovaný systém je v průměru výdělečný. Je značný rozdíl v zisku nalezených strategií v případě nasazení bezprostředně po natrénování a v případě časového intervalu, který časově nesouvisí s intervalem pro trénování. Je tedy podstatné systém v čase adaptovat. Tabulka 6.6.6 ukazuje ztrátovost strategie v době, která není přímo za intervalem pro trénování.

V rámci práce bylo zmíněno několik heuristik, které mohou urychlit nalezení ziskových strategií. Implementace všech by byla vzhledem k času implementace nad rámec této práce. V práci jsem implementoval propojení s technickými indikátory a otestoval navržené propojení s neuronovými sítěmi. Vzhledem k článkům zmíněných v úvodu a informacím uvedených v knize [1] jsem chtěl otestovat, zda předzpracování forexových dat způsobem rozdělení do dvou tříd jako ztrátových a ziskových pozic bude mít za následek dobré natrénování ANN. Nasazení neuronové sítě selhalo kvůli předem neviděné vlastnosti systému. Nalezené strategie neměly vyrovnané velikosti tříd nebo nedokázaly držet dostatek pozic, tedy dat pro trénování. Díky tomu, že je signál nestacionární a dalším vlastnostem byly často nalezeny pozice pouze na trénovaném intervalu a času, který je blízko tomuto intervalu. Možnosti dalšího vylepšení jsou uvedeny v následující kapitole 7.1

### 7.1 Návrh dalšího vývoje

Je nespočet cest, které mohou vést ke zlepšení. Prvním krokem je vylepšení simulace obchodování. Implementovat kontrolu přímo tickových dat. Práce s tickovými daty by byla výpočetně náročná. Bylo by tedy nutné systém optimalizovat. Držet vypočítané operace,

které předcházejí změně po mutaci, tedy nepře počítávat hlavní matici celou. Ukládat mezi výsledky na disk kvůli vyšším nárokům na paměť. Při kontrole každého ticku bude fitness funkce přesnější.

Do systému je možné implementovat další matematické operace, které například zvládnou nalezení kolerace, autoregrese nebo ARMA modelu. Případně ošetřit stávající matematické operace, tak aby nedocházelo k extrémním výkyvům. Případně ošetřit výpočet nedefinovaných hodnot u operací, které byly odstraněny.

Pro test ANN je nutné definovat vstupní data a matematické operace tak, aby byl vygenerován dostatečný počet obchodních pozic pro natrénování ANN i mimo interval, na kterém byla strategie nalezena. Selhání spočívalo v nedostatku dat kvůli málo vygenerovaným pozicím. Tedy pokud byla strategie nalezena na intervalu o délce 8000 a bylo provedeno například 500 pozic, pak očekáváme, že pokud bychom strategii aplikovali na větší historický interval, dostaneme v poměru více pozic. v aplikaci systému tomu však tak aktuálně není.



# Literatura

- [1] Bishop, C. M.: *Pattern recognition and machine learning*. Springer, 2006, ISBN 978-0387310732.
- [2] Brabazon, A.; O'Neill, M.: Evolving technical trading rules for spot foreign-exchange markets using grammatical evolution. *Computational Management Science*, ročník 1, č. 3, Oct 2004: s. 311–327, ISSN 1619-6988, doi:10.1007/s10287-004-0018-5. URL <https://doi.org/10.1007/s10287-004-0018-5>
- [3] Brogaard, J. A.: *HIGH FREQUENCY TRADING AND ITS IMPACT ON MARKET QUALITY*. [Online; navštíveno 12.01.2019]. URL <http://www.clasesdebolsa.com/archivos/HTF.pdf>
- [4] Carter, J. F.: *Mastering the Trade, Second Edition: Proven Techniques for Profiting from Intraday and Swing Trading Setups*. McGraw-Hill Education, 2012, ISBN 978-0071775144.
- [5] De Silva, L. P. H. W., Anthony Mihirana: *Grammar-based feature generation for time-series prediction*. Springer, 2015, ISBN 978-9812874108.
- [6] Ding, Y.; Cai, Y.; Sun, P.; aj.: The Use of Combined Neural Networks and Genetic Algorithms for Prediction of River Water Quality. *Journal of Applied Research and Technology*, ročník 12, č. 3, 2014: s. 493 – 499, ISSN 1665-6423, doi:[https://doi.org/10.1016/S1665-6423\(14\)71629-3](https://doi.org/10.1016/S1665-6423(14)71629-3). URL <http://www.sciencedirect.com/science/article/pii/S1665642314716293>
- [7] Durenard, E. A.: *Professional Automated Trading: Theory and Practice*. Wiley, 2013, ISBN 978-1-118-12985-2.
- [8] Eyden, R. J. V.: *The Application of Neural Networks in the Forecasting of Share Prices*. Finance Technology Pub, 1996, ISBN 978-0965133203.
- [9] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., první vydání, 1989, ISBN 0201157675.
- [10] Grimes, A.: *The Art and Science of Technical Analysis: Market Structure, Price Action, and Trading Strategies*. Wiley Trading, 2012, ISBN 978-1-118-11512-1.
- [11] Hopcroft; Motwani; Ullman: *Introduction to Automata Theory, Languages, and Computation*. AddisonWesley, 2000, ISBN 0-201-44124-1.
- [12] Hynek, J.: *Genetické algoritmy a genetické programování*. Grada Publishing,, 2008, ISBN 978-80-247-2695-3.

- [13] Kathy, L.: *FOREX – Ziskové intradenní a swingové obchodní strategie*. FXstreet.cz, 2013, ISBN 978-80-904418-2.
- [14] KAČER, P.: Forexový automatický obchodní systém založený na neuronových sítích [online]. *Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií*. URL <http://hdl.handle.net/11012/38431>
- [15] Khadjeh Nassirtoussi, A.; Aghabozorgi, S.; Wah, T.; aj.: Text mining for market prediction: A systematic review. *Expert Systems with Applications*, ročník 41, 11 2014: s. 7653–7670, doi:10.1016/j.eswa.2014.06.009.
- [16] Laxpati, S. R.; Goncharoff, V.: *Practical Signal Processing and Its Applications: With Solved Homework Problems (Advanced Series in Electrical and Computer Engineering)*. World Scientific Publishing Co Pte Ltd, 2018, ISBN 978-9813224025.
- [17] Mendes, L.; Godinho, P.; Dias, J.: A Forex trading system based on a genetic algorithm. *Journal of Heuristics*, ročník 18, č. 4, Aug 2012: s. 627–656, ISSN 1572-9397, doi:10.1007/s10732-012-9201-y. URL <https://doi.org/10.1007/s10732-012-9201-y>
- [18] MIKULENČÁK, R.: PREDIKCE KURSŮ PRO OBCHODOVÁNÍ NA AKCIOVÝCH TRZÍCH. *Vysoké učení technické v Brně.*, 2011.
- [19] Nassirtoussi, A. K.; Aghabozorgi, S.; Wah, T. Y.; aj.: Text mining for market prediction: A systematic review. *Expert Systems with Applications*, ročník 41, č. 16, 2014: s. 7653 – 7670, ISSN 0957-4174, doi:<https://doi.org/10.1016/j.eswa.2014.06.009>. URL <http://www.sciencedirect.com/science/article/pii/S0957417414003455>
- [20] Reeves, C.; Rowe, J. E.: *Genetic Algorithms: Principles and Perspectives*. Springer US, 2002, ISBN 978-1-4020-7240-6.
- [21] Sebastiani, F.: Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, ročník 34, č. 1, Březen 2002: s. 1–47, ISSN 0360-0300, doi:10.1145/505282.505283. URL <http://doi.acm.org/10.1145/505282.505283>
- [22] Sekanina Lukáš, p. I., P.: Evoluční design, Biologií inspirované počítače. *Vysoké učení technické*. URL <http://www.fit.vutbr.cz/study/course-1.php.cs?id=12653>
- [23] SHARPE, M. J.: LOGNORMAL MODEL FOR STOCK PRICES. 1998. URL <http://math.ucsd.edu/~msharpe/stockgrowth.pdf>
- [24] Sprott, J. C.: *Chaos and Time-Series Analysis*. Oxford University Press, 2001, ISBN 978-0198508403.
- [25] Tsay, R. S.: *Analysis of Financial Time Series*. Wiley,, 2010, ISBN 978-0470414354.
- [26] Yann, B. G. B. O., L.; Leon: Efficient BackProp. *Springer*, 1998. URL <http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf>

- [27] Zhang, G.; Patuwo, B. E.; Hu, M. Y.: Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, ročník 14, č. 1, 1998: s. 35 – 62, ISSN 0169-2070, doi:[https://doi.org/10.1016/S0169-2070\(97\)00044-7](https://doi.org/10.1016/S0169-2070(97)00044-7).  
URL <http://www.sciencedirect.com/science/article/pii/S0169207097000447>

## Příloha A

# Obsah příloženého paměťového média

Obsahem CD jsou zdrojové kódy v jazyce python ve složce dp-code. Složka dp-code také obsahuje příklad testovacích dat a složku debug, která ukládá objekt nalezené strategie. Obsahem je také demonstrační video a plakát.

# Příloha B

## Manuál

Program je postaven na jazyce Python 3. Využívá několik knihoven pro práci s daty, které je nutné instalovat:

- ScyPy - programový balíček pro práci s daty
- Pandas - knihovna poskytující operace a výpočty s maticí
- Numpy - knihovna poskytující operace a výpočty s maticí
- Matplotlib - knihovna pro zobrazení dat
- Tensorflow - backend pro simulaci neuronové sítě
- Keras - nadstavba nad knihovnou Tensorflow
- Deap - knihovna pro práci s genetickými algoritmy
- sklearn - knihovna pro PCA

Knihovna pickle by měla být nainstalována snumpy, pokud není, je třeba jí také doinstalovat. Stejně tak knihovna operator by měla být v základu. Další knihovny by měly být v základu Python 3. Pokud nebudou, je také potřeba je doinstalovat. Script GAengine.py slouží pro spuštění systému v různých módech. Pro výpočet pozic optimalizované strategie je možné spustit GAengine.py -o train. Optimalizace bude spuštěna s počáteční generací o velikosti 1000.

## Příloha C

# Konfigurační soubor

- data - eurusd4.csv - soubor s testovacími daty
- population-size - 1000 - velikost populace
- data-len - 8000 - úsek vyjmutý z dat pro testování

**Příloha D**

**Testy Metatrader**

# Strategy Tester Report

## tester

(Build 1170)

Symbol	EURUSD (Euro vs US Dollar)		
Období	5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.01.23 - 2018.01.25)		
Model	Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)		
Testování sloupcových grafů	74670	Značky namodelovány	25117219 Modelování kvality 99.90%
Chyba neshodných grafů	0		
Počáteční depozit	10000.00	Spread	Aktuální (18)
Celkový čistý zisk	850.00	Hrubý zisk	850.00 Hrubá ztráta -0.00
Ziskový faktor	Předpokládaný zisk		170.00
Absolutní pokles	170.00	Maximální pokles	367.00 (3.60%) Relativní pokles 3.60% (367.00)
Transakce celkem	5	Krátké pozice (výhra %)	0 (0.00%) Dlouhé pozice (výhra %) 5 (100.00%)
		Ziskové obchody (% z celkové výše)	5 (100.00%) Ztrátové obchody (% z celkové výše) 0 (0.00%)
		Největší ziskový obchod	170.00 ztrátový obchod -0.00
		Průměrný ziskový obchod	170.00 ztrátový obchod -0.00
		Maximálně návazné výhry (finanční zisk)	5 (850.00) návazné prohry (finanční ztráta) 0 (-0.00)
		Maximální návazný zisk (počet zisků)	850.00 (5) návazná ztráta (počet ztrát) -0.00 (0)
		Průměrný návazné výhry	5 návazné prohry 0



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.01.23 08:20	buy	1	1.00	1.22315	1.22027	1.22485		
2	2018.01.23 10:02	t/p	1	1.00	1.22485	1.22027	1.22485	170.00	10170.00
3	2018.01.24 04:15	buy	2	1.00	1.23263	1.22975	1.23433		
4	2018.01.24 06:35	buy	3	1.00	1.23133	1.22845	1.23303		
5	2018.01.24 08:14	t/p	3	1.00	1.23303	1.22845	1.23303	170.00	10340.00
6	2018.01.24 08:52	t/p	2	1.00	1.23433	1.22975	1.23433	170.00	10510.00
7	2018.01.24 13:40	buy	4	1.00	1.23534	1.23246	1.23704		
8	2018.01.24 14:03	t/p	4	1.00	1.23704	1.23246	1.23704	170.00	10680.00
9	2018.01.24 14:45	buy	5	1.00	1.23829	1.23541	1.23999		
10	2018.01.24 15:04	t/p	5	1.00	1.23999	1.23541	1.23999	170.00	10850.00



# Strategy Tester Report

## tester

(Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.01.23 - 2018.02.01)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=170; Lots=1; TrailingStop=30; StopLoss=270;

Testování sloupcových grafů 74670 Značky namodelovány 25117219 Modelování kvality 99.90%  
 Chyba neshodných grafů 0

Počáteční depozit	10000.00	Spread	Aktuální (18)
Celkový čistý zisk	517.40	Hrubý zisk	3398.70
Ziskový faktor	1.18	Hrubá ztráta	-2881.30
Absolutní pokles	262.00	Předpokládaný zisk	17.25
		Maximální pokles	1565.00 (13.85%)
Transakce celkem	30	Krátké pozice (výhra %)	0 (0.00%)
		Ziskové obchody (% z celkové výše)	20 (66.67%)
		Největší ziskový obchod	170.00
		Průměrný ziskový obchod	169.94
		Maximálně návazné výhry (finanční zisk)	7 (1190.00)
		Maximální návazný zisk (počet zisků)	1190.00 (7)
		Průměrný návazné výhry	3



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.01.23 08:20	buy	1	1.00	1.22315	1.22027	1.22485		
2	2018.01.23 10:02	t/p	1	1.00	1.22485	1.22027	1.22485	170.00	10170.00
3	2018.01.24 04:15	buy	2	1.00	1.23263	1.22975	1.23433		
4	2018.01.24 06:35	buy	3	1.00	1.23133	1.22845	1.23303		
5	2018.01.24 08:14	t/p	3	1.00	1.23303	1.22845	1.23303	170.00	10340.00
6	2018.01.24 08:52	t/p	2	1.00	1.23433	1.22975	1.23433	170.00	10510.00
7	2018.01.24 13:40	buy	4	1.00	1.23534	1.23246	1.23704		
8	2018.01.24 14:03	t/p	4	1.00	1.23704	1.23246	1.23704	170.00	10680.00
9	2018.01.24 14:45	buy	5	1.00	1.23829	1.23541	1.23999		
10	2018.01.24 15:04	t/p	5	1.00	1.23999	1.23541	1.23999	170.00	10850.00
11	2018.01.25 06:55	buy	6	1.00	1.24305	1.24017	1.24475		
12	2018.01.25 07:05	buy	7	1.00	1.24235	1.23947	1.24405		
13	2018.01.25 07:35	buy	8	1.00	1.24145	1.23857	1.24315		
14	2018.01.25 07:49	t/p	8	1.00	1.24315	1.23857	1.24315	170.00	11020.00
15	2018.01.25 07:49	t/p	7	1.00	1.24405	1.23947	1.24405	170.00	11190.00
16	2018.01.25 08:00	buy	9	1.00	1.24205	1.23917	1.24375		
17	2018.01.25 08:30	s/l	6	1.00	1.24017	1.24017	1.24475	-288.00	10902.00
18	2018.01.25 08:38	s/l	9	1.00	1.23917	1.23917	1.24375	-288.00	10614.00
19	2018.01.25 14:20	buy	10	1.00	1.24864	1.24576	1.25034		
20	2018.01.25 14:45	buy	11	1.00	1.24711	1.24423	1.24881		
21	2018.01.25 14:57	t/p	11	1.00	1.24881	1.24423	1.24881	170.00	10784.00
22	2018.01.25 15:07	t/p	10	1.00	1.25034	1.24576	1.25034	170.00	10954.00
23	2018.01.25 16:05	buy	12	1.00	1.24930	1.24642	1.25100		
24	2018.01.25 16:35	buy	13	1.00	1.24964	1.24676	1.25134		
25	2018.01.25 18:45	buy	14	1.00	1.24783	1.24495	1.24953		
26	2018.01.25 19:03	s/l	13	1.00	1.24676	1.24676	1.25134	-288.00	10666.00
27	2018.01.25 19:03	s/l	12	1.00	1.24642	1.24642	1.25100	-288.00	10378.00
28	2018.01.25 19:03	s/l	14	1.00	1.24495	1.24495	1.24953	-288.00	10090.00
29	2018.01.25 19:05	buy	15	1.00	1.24343	1.24055	1.24513		
30	2018.01.25 19:23	s/l	15	1.00	1.24055	1.24055	1.24513	-288.00	9802.00
31	2018.01.25 19:25	buy	16	1.00	1.23916	1.23628	1.24086		

# Strategy Tester Report

## tester

(Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.03.12 - 2018.03.20)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=180; Lots=1; TrailingStop=30; StopLoss=240;

Testování sloupcových grafů	74670	Značky namodelovány	25117219	Modelování kvality	99.90%
Chyba neshodných grafů	0				
Počáteční depozit	10000.00			Spread	Aktuální (18)
Celkový čistý zisk	540.00	Hrubý zisk	540.00	Hrubá ztráta	-0.00
Ziskový faktor		Předpokládaný zisk	180.00		
Absolutní pokles	306.00	Maximální pokles	432.00 (4.19%)	Relativní pokles	4.19% (432.00)
Transakce celkem	3	Krátké pozice (výhra %)	0 (0.00%)	Dlouhé pozice (výhra %)	3 (100.00%)
		Ziskové obchody (% z celkové výše)	3 (100.00%)	Ztrátové obchody (% z celkové výše)	0 (0.00%)
		Největší ziskový obchod	180.00	ztrátový obchod	-0.00
		Průměrný ziskový obchod	180.00	ztrátový obchod	-0.00
		Maximálně návazné výhry (finanční zisk)	3 (540.00)	návazné prohry (finanční ztráta)	0 (-0.00)
		Maximální návazný zisk (počet zisků)	540.00 (3)	návazná ztráta (počet ztrát)	-0.00 (0)
		Průměrný návazné výhry	3	návazné prohry	0



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.03.13 05:50	buy	1	1.00	1.23295	1.23037	1.23475		
2	2018.03.13 07:00	buy	2	1.00	1.23301	1.23043	1.23481		
3	2018.03.13 08:20	buy	3	1.00	1.23228	1.22970	1.23408		
4	2018.03.13 10:45	t/p	3	1.00	1.23408	1.22970	1.23408	180.00	10180.00
5	2018.03.13 10:45	t/p	1	1.00	1.23475	1.23037	1.23475	180.00	10360.00
6	2018.03.13 10:45	t/p	2	1.00	1.23481	1.23043	1.23481	180.00	10540.00

# Strategy Tester Report

## tester

(Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2018.12.01 00:00 - 2018.11.30 21:59 (2018.03.12 - 2018.03.20)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=180; Lots=1; TrailingStop=30; StopLoss=240;

Testování sloupcových grafů 74670 Značky namodelovány 25117219 Modelování kvality 99.90%  
 Chyba neshodných grafů 0

Počáteční depozit	10000.00		Spread	Aktuální (18)	
Celkový čistý zisk	282.00	Hrubý zisk	540.00	Hrubá ztráta	-258.00
Ziskový faktor	2.09	Předpokládaný zisk	70.50		
Absolutní pokles	306.00	Maximální pokles	432.00 (4.19%)	Relativní pokles	4.19% (432.00)
Transakce celkem	4	Krátké pozice (výhra %)	0 (0.00%)	Dlouhé pozice (výhra %)	4 (75.00%)
		Ziskové obchody (% z celkové výše)	3 (75.00%)	Ztrátové obchody (% z celkové výše)	1 (25.00%)
		Největší ziskový obchod	180.00	ztrátový obchod	-258.00
		Průměrný ziskový obchod	180.00	ztrátový obchod	-258.00
		Maximálně návazné výhry (finanční zisk)	3 (540.00)	návazné prohry (finanční ztráta)	1 (-258.00)
		Maximální návazný zisk (počet zisků)	540.00 (3)	návazná ztráta (počet ztrát)	-258.00 (1)
		Průměrný návazné výhry	3	návazné prohry	1



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.03.13 05:50	buy	1	1.00	1.23295	1.23037	1.23475		
2	2018.03.13 07:00	buy	2	1.00	1.23301	1.23043	1.23481		
3	2018.03.13 08:20	buy	3	1.00	1.23228	1.22970	1.23408		
4	2018.03.13 10:45	t/p	3	1.00	1.23408	1.22970	1.23408	180.00	10180.00
5	2018.03.13 10:45	t/p	1	1.00	1.23475	1.23037	1.23475	180.00	10360.00
6	2018.03.13 10:45	t/p	2	1.00	1.23481	1.23043	1.23481	180.00	10540.00
7	2018.03.14 01:00	buy	4	1.00	1.23960	1.23702	1.24140		
8	2018.03.14 08:15	s/l	4	1.00	1.23702	1.23702	1.24140	-258.00	10282.00

# Strategy Tester Report

## tester

(Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.04.17 - 2018.04.27)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=220; Lots=1; TrailingStop=30; StopLoss=160;

Testování sloupcových grafů 74670 Značky namodelovány 25117219 Modelování kvality 99.90%  
 Chyba neshodných grafů 0

Počáteční depozit	10000.00		Spread	Aktuální (18)
Celkový čistý zisk	-136.00	Hrubý zisk	220.00	Hrubá ztráta -356.00
Ziskový faktor	0.62	Předpokládaný zisk	-45.33	
Absolutní pokles	136.00	Maximální pokles	549.00 (5.27%)	Relativní pokles 5.27% (549.00)
Transakce celkem	3	Krátké pozice (výhra %)	0 (0.00%)	Dlouhé pozice (výhra %) 3 (33.33%)
		Ziskové obchody (% z celkové výše)	1 (33.33%)	Ztrátové obchody (% z celkové výše) 2 (66.67%)
		Největší ziskový obchod	220.00	ztrátový obchod -178.00
		Průměrný ziskový obchod	220.00	ztrátový obchod -178.00
		Maximálně návazné výhry (finanční zisk)	1 (220.00)	návazné prohry (finanční ztráta) 2 (-356.00)
		Maximální návazný zisk (počet zisků)	220.00 (1)	návazná ztráta (počet ztrát) -356.00 (2)
		Průměrný návazné výhry	1	návazné prohry 2



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.04.19 06:05	buy	1	1.00	1.23734	1.23556	1.23954		
2	2018.04.19 07:25	t/p	1	1.00	1.23954	1.23556	1.23954	220.00	10220.00
3	2018.04.19 10:20	buy	2	1.00	1.23687	1.23509	1.23907		
4	2018.04.19 12:00	buy	3	1.00	1.23828	1.23650	1.24048		
5	2018.04.19 14:58	s/l	3	1.00	1.23650	1.23650	1.24048	-178.00	10042.00
6	2018.04.19 15:33	s/l	2	1.00	1.23509	1.23509	1.23907	-178.00	9864.00

# Strategy Tester Report

## tester

(Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.04.17 - 2018.04.27)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=220; Lots=1; TrailingStop=30; StopLoss=160;

Testování sloupcových grafů	74670	Značky namodelovány	25117219	Modelování kvality	99.90%
Chyba neshodných grafů	0				
Počáteční depozit	10000.00			Spread	Aktuální (18)
Celkový čistý zisk	-850.60	Hrubý zisk	220.00	Hrubá ztráta	-1070.60
Ziskový faktor	0.21	Předpokládaný zisk	-121.51		
Absolutní pokles	850.60	Maximální pokles	1263.60 (12.13%)	Relativní pokles	12.13% (1263.60)
Transakce celkem	7	Krátké pozice (výhra %)	0 (0.00%)	Dlouhé pozice (výhra %)	7 (14.29%)
		Ziskové obchody (% z celkové výše)	1 (14.29%)	Ztrátové obchody (% z celkové výše)	6 (85.71%)
		Největší ziskový obchod	220.00	ztrátový obchod	-179.30
		Průměrný ziskový obchod	220.00	ztrátový obchod	-178.43
		Maximálně návazné výhry (finanční zisk)	1 (220.00)	návazné prohry (finanční ztráta)	6 (-1070.60)
		Maximální návazný zisk (počet zisků)	220.00 (1)	návazná ztráta (počet ztrát)	-1070.60 (6)
		Průměrný návazné výhry	1	návazné prohry	6



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.04.19 06:05	buy	1	1.00	1.23734	1.23556	1.23954		
2	2018.04.19 07:25	t/p	1	1.00	1.23954	1.23556	1.23954	220.00	10220.00
3	2018.04.19 10:20	buy	2	1.00	1.23687	1.23509	1.23907		
4	2018.04.19 12:00	buy	3	1.00	1.23828	1.23650	1.24048		
5	2018.04.19 14:58	s/l	3	1.00	1.23650	1.23650	1.24048	-178.00	10042.00
6	2018.04.19 15:33	s/l	2	1.00	1.23509	1.23509	1.23907	-178.00	9864.00
7	2018.04.23 20:30	buy	4	1.00	1.22092	1.21914	1.22312		
8	2018.04.24 01:11	s/l	4	1.00	1.21914	1.21914	1.22312	-179.30	9684.70
9	2018.04.24 05:15	buy	5	1.00	1.22179	1.22001	1.22399		
10	2018.04.24 07:07	s/l	5	1.00	1.22001	1.22001	1.22399	-178.00	9506.70
11	2018.04.24 18:50	buy	6	1.00	1.22394	1.22216	1.22614		
12	2018.04.25 01:20	s/l	6	1.00	1.22216	1.22216	1.22614	-179.30	9327.40
13	2018.04.26 06:40	buy	7	1.00	1.21845	1.21667	1.22065		
14	2018.04.26 07:23	s/l	7	1.00	1.21667	1.21667	1.22065	-178.00	9149.40

# Strategy Tester Report

## tester

(Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.05.25 - 2018.05.29)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=190; Lots=1; TrailingStop=30; StopLoss=190;

Testování sloupcových grafů 74670 Značky namodelovány 25117219 Modelování kvality 99.90%  
 Chyba neshodných grafů 0

Počáteční depozit	10000.00		Spread	Aktuální (18)	
Celkový čistý zisk	552.00	Hrubý zisk	760.00	Hrubá ztráta	-208.00
Ziskový faktor	3.65	Předpokládaný zisk	110.40		
Absolutní pokles	165.00	Maximální pokles	901.00 (8.35%)	Relativní pokles	8.35% (901.00)
Transakce celkem	5	Krátké pozice (výhra %)	0 (0.00%)	Dlouhé pozice (výhra %)	5 (80.00%)
		Ziskové obchody (% z celkové výše)	4 (80.00%)	Ztrátové obchody (% z celkové výše)	1 (20.00%)
		Největší ziskový obchod	190.00	ztrátový obchod	-208.00
		Průměrný ziskový obchod	190.00	ztrátový obchod	-208.00
		Maximálně návazné výhry (finanční zisk)	2 (380.00)	návazné prohry (finanční ztráta)	1 (-208.00)
		Maximální návazný zisk (počet zisků)	380.00 (2)	návazná ztráta (počet ztrát)	-208.00 (1)
		Průměrný návazné výhry	2	návazné prohry	1



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.05.25 14:10	buy	1	1.00	1.16587	1.16379	1.16777		
2	2018.05.25 14:50	buy	2	1.00	1.16556	1.16348	1.16746		
3	2018.05.25 15:00	buy	3	1.00	1.16568	1.16360	1.16758		
4	2018.05.25 15:25	buy	4	1.00	1.16603	1.16395	1.16793		
5	2018.05.25 16:00	buy	5	1.00	1.16671	1.16463	1.16861		
6	2018.05.25 17:05	t/p	2	1.00	1.16746	1.16348	1.16746	190.00	10190.00
7	2018.05.25 17:05	t/p	3	1.00	1.16758	1.16360	1.16758	190.00	10380.00
8	2018.05.25 20:51	s/l	5	1.00	1.16463	1.16463	1.16861	-208.00	10172.00
9	2018.05.27 21:00	t/p	1	1.00	1.16777	1.16379	1.16777	190.00	10362.00
10	2018.05.27 21:00	t/p	4	1.00	1.16793	1.16395	1.16793	190.00	10552.00

# Strategy Tester Report tester (Build 1170)

Symbol EURUSD (Euro vs US Dollar)  
 Období 5 minut (M5) 2017.12.01 00:00 - 2018.11.30 21:59 (2018.05.25 - 2018.06.02)  
 Model Všechny cenové pohyby (nejpřesnější metoda založena na všech dostupných nižších časových intervalech)  
 Parametry TakeProfit=190; Lots=1; TrailingStop=30; StopLoss=190;

Testování sloupcových grafů 74670 Značky namodelovány 25117219 Modelování kvality 99.90%  
 Chyba neshodných grafů 0

Počáteční depozit	10000.00		Spread	Aktuální (18)	
Celkový čistý zisk	1826.70	Hrubý zisk	2658.70	Hrubá ztráta	-832.00
Ziskový faktor	3.20	Předpokládaný zisk	101.48		
Absolutní pokles	165.00	Maximální pokles	901.00 (8.35%)	Relativní pokles	8.35% (901.00)
Transakce celkem	18	Krátké pozice (výhra %)	0 (0.00%)	Dlouhé pozice (výhra %)	18 (77.78%)
		Ziskové obchody (% z celkové výše)	14 (77.78%)	Ztrátové obchody (% z celkové výše)	4 (22.22%)
		Největší ziskový obchod	190.00	ztrátový obchod	-208.00
		Průměrný ziskový obchod	189.91	ztrátový obchod	-208.00
		Maximálně návazné výhry (finanční zisk)	7 (1328.70)	návazné prohry (finanční ztráta)	1 (-208.00)
		Maximální návazný zisk (počet zisků)	1328.70 (7)	návazná ztráta (počet ztrát)	-208.00 (1)
		Průměrný návazné výhry	4	návazné prohry	1



#	Čas	Typ	Pokyn	Lotů	Cena	S / L	Realizovat zisk	Zisk	Zůstatek:
1	2018.05.25 14:10	buy	1	1.00	1.16587	1.16379	1.16777		
2	2018.05.25 14:50	buy	2	1.00	1.16556	1.16348	1.16746		
3	2018.05.25 15:00	buy	3	1.00	1.16568	1.16360	1.16758		
4	2018.05.25 15:25	buy	4	1.00	1.16603	1.16395	1.16793		
5	2018.05.25 16:00	buy	5	1.00	1.16671	1.16463	1.16861		
6	2018.05.25 17:05	t/p	2	1.00	1.16746	1.16348	1.16746	190.00	10190.00
7	2018.05.25 17:05	t/p	3	1.00	1.16758	1.16360	1.16758	190.00	10380.00
8	2018.05.25 20:51	s/l	5	1.00	1.16463	1.16463	1.16861	-208.00	10172.00
9	2018.05.27 21:00	t/p	1	1.00	1.16777	1.16379	1.16777	190.00	10362.00
10	2018.05.27 21:00	t/p	4	1.00	1.16793	1.16395	1.16793	190.00	10552.00
11	2018.05.27 23:45	buy	6	1.00	1.16872	1.16664	1.17062		
12	2018.05.28 00:00	buy	7	1.00	1.16904	1.16696	1.17094		
13	2018.05.28 01:09	t/p	6	1.00	1.17062	1.16664	1.17062	188.70	10740.70
14	2018.05.28 01:14	t/p	7	1.00	1.17094	1.16696	1.17094	190.00	10930.70
15	2018.05.29 09:50	buy	8	1.00	1.15197	1.14989	1.15387		
16	2018.05.29 09:57	t/p	8	1.00	1.15387	1.14989	1.15387	190.00	11120.70
17	2018.05.30 07:20	buy	9	1.00	1.15745	1.15537	1.15935		
18	2018.05.30 07:35	buy	10	1.00	1.15810	1.15602	1.16000		
19	2018.05.30 07:46	t/p	9	1.00	1.15935	1.15537	1.15935	190.00	11310.70
20	2018.05.30 07:55	t/p	10	1.00	1.16000	1.15602	1.16000	190.00	11500.70
21	2018.05.30 12:35	buy	11	1.00	1.16325	1.16117	1.16515		
22	2018.05.30 14:02	s/l	11	1.00	1.16117	1.16117	1.16515	-208.00	11292.70
23	2018.05.30 15:25	buy	12	1.00	1.16273	1.16065	1.16463		
24	2018.05.30 15:31	t/p	12	1.00	1.16463	1.16065	1.16463	190.00	11482.70
25	2018.05.31 13:55	buy	13	1.00	1.16594	1.16386	1.16784		
26	2018.05.31 14:25	buy	14	1.00	1.16621	1.16413	1.16811		
27	2018.05.31 14:36	t/p	13	1.00	1.16784	1.16386	1.16784	190.00	11672.70
28	2018.05.31 14:44	t/p	14	1.00	1.16811	1.16413	1.16811	190.00	11862.70
29	2018.05.31 14:50	buy	15	1.00	1.16745	1.16537	1.16935		
30	2018.05.31 15:09	s/l	15	1.00	1.16537	1.16537	1.16935	-208.00	11654.70
31	2018.05.31 15:30	buy	16	1.00	1.16611	1.16403	1.16801		