

Uniwersytet Warszawski  
Wydział Nauk Ekonomicznych

Michał Ejdys  
Nr albumu: 201258

**Weryfikacja hipotezy efektywności rynku  
przy użyciu metod  
programowania genetycznego**

Praca magisterska  
na kierunku: Informatyka i Ekonometria

Praca wykonana pod kierunkiem  
dr. Tomasza Kopczewskiego  
z Zakładu Mikroekonomii  
WNE UW

Warszawa, sierpień 2007

*Oświadczenie kierującego pracą*

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

*Oświadczenie autora (autorów) pracy*

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa (magisterska) została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

## **Streszczenie**

Praca przedstawia historię kształtowania się koncepcji efektywności informacyjnej rynków kapitałowych, wraz z jej trzema formami: słabą, półsilną oraz silną. Opisuje jedną z odmian obliczeń ewolucyjnych (programowanie genetyczne) jako metodę modelowania ekonomicznego. Praca opisuje implementację ewolucyjnego modelowania ekonomicznego w języku R oraz przedstawia badanie efektywności warszawskiej Giełdy Papierów Wartościowych przy użyciu tego narzędzia.

## **Słowa kluczowe**

ewolucja, programowanie genetyczne, modelowanie ekonomiczne, ekonometria, efektywność informacyjna

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

Ekonomia (14300)

## **Klasyfikacja tematyczna**

## SPIS TREŚCI

WSTĘP .....	5
ROZDZIAŁ I. Efektywność informacyjna rynku .....	6
1.1. Historia koncepcji efektywności informacyjnej.....	6
1.1.1. Formy efektywności .....	7
1.1.2. Formalne definicje efektywności.....	8
1.1.3. Krytyka teorii .....	10
1.1.4. Sposoby sprawdzania efektywności rynku .....	11
1.2. Problemy badania efektywności .....	13
1.3. Przykładowe badania słabej efektywności rynku.....	16
1.3.1. Szukanie regularności w danych .....	17
1.3.2. Zastosowanie programowania genetycznego .....	18
1.4. Efektywność rynku w Polsce .....	20
ROZDZIAŁ II. Ewolucyjne modelowanie ekonomiczne .....	23
2.1. Obliczenia ewolucyjne.....	23
2.1.1. Biologiczne fundamenty .....	23
2.1.2. Historia obliczeń ewolucyjnych .....	25
2.1.3. Przegląd koncepcji obliczeń ewolucyjnych.....	26
2.1.4. Przykłady zastosowań.....	27
2.2. Algorytmy ewolucyjne a modelowanie ekonomiczne .....	31
2.3. Programowanie genetyczne.....	32
2.3.1. Reprezentacja funkcji jako drzewek .....	33
2.3.2. Generowanie nowych osobników .....	34
2.3.3. Cenzurowanie funkcji.....	35
2.3.4. Operacje genetyczne .....	35
2.3.5. Przeżycie najbardziej przystosowanego.....	38
2.3.6. Parametry programowania genetycznego.....	39
2.3.7. Schemat programowania genetycznego.....	40
2.4. Zastosowanie programowania genetycznego do modelowania ekonomicznego .....	40
2.4.1. Równowaga na rynku pieniądza .....	41
2.4.2. Modele wielorównaniowe .....	43
2.4.3. Inne zastosowania programowania genetycznego .....	44
ROZDZIAŁ III. Implementacja programowania genetycznego jako narzędzia modelowania ekonomicznego .....	46
3.1. Oprogramowanie otwarte.....	46
3.1.1. Definicja .....	46

3.1.2. Wady i zalety .....	47
3.1.3. Przykłady .....	49
3.2. Narzędzia ekonomistów .....	49
3.3. Szczegółowy opis R .....	52
3.3.1. Typy danych .....	52
3.3.2. W kierunku języka obiektowego .....	53
3.3.3. W kierunku języka funkcyjnego .....	55
3.4. Opis implementacji algorytmu .....	56
ROZDZIAŁ IV. Efektywność warszawskiej Giełdy Papierów Wartościowych .....	60
4.1. Przebieg badania .....	60
4.2. Przekrojowe wyniki modelowania .....	61
4.3. Analiza modeli otrzymanych w toku ewolucji .....	62
4.4. Porównanie jakościowe modeli pomiędzy okresami .....	67
4.5. Wnioski .....	68
PODSUMOWANIE .....	70
BIBLIOGRAFIA .....	70
ZESTAWIENIE SPISÓW .....	73

## WSTĘP

W latach sześćdziesiątych XX wieku amerykański ekonomista Eugeniusz Fama opracował pojęcie efektywności informacyjnej rynków kapitałowych. Podejście to równie szybko znalazło swoich zwolenników co przeciwników. Zwolennicy uważają, że efektywność to wyznacznik dojrzałości rynku. Przeciwnicy twierdzą, że to koncepcja czysto teoretyczna. Spór pomiędzy nimi trwa od kilkudziesięciu lat i obydwie strony wciąż dostarczają nowych ciekawych badań na temat efektywności. Koncepcja ta jest o tyle interesująca, że w przypadku rynku informacyjnie efektywnego, nie jest możliwe osiągnięcie ponadnormalnych zysków. Teoretycznie zatem traci sens m.in. inwestowanie na efektywnej informacyjnie giełdzie.

Celem niniejszej pracy jest przeprowadzenie weryfikacji efektywności informacyjnej warszawskiej Giełdy Papierów Wartościowych. Dodatkowym celem jest sprawdzenie na ile programowanie genetyczne (jedna z metod obliczeń ewolucyjnych) jest przydatne w pracy ekonomisty, w szczególności zajmującego się badaniem efektywności giełdy.

W pracy opisano szczegółowo koncepcję efektywności informacyjnej, czemu poświęcony został rozdział I. Zawiera on opis historii prac nad samą koncepcją oraz przedstawia przykładowe jej badania. Drugim filarem pracy są obliczenia ewolucyjne, którym poświęcony jest rozdział II. Opisuje on całą rodzinę zastosowań ewolucji w programowaniu oraz szczególną uwagę poświęca programowaniu genetycznemu, które stosuje się do modelowania ekonomicznego. Niestety, brak jest uniwersalnych narzędzi do modelowania przy użyciu ewolucji. Stąd w pracy dokonana i przedstawiona została własna implementacja modelowania przy użyciu programowania genetycznego. Na jej potrzeby użyty został pakiet statystyczny R. Jego opis, jak również szczegóły dokonanej implementacji znajdują się w rozdziale III. Ostatnią częścią pracy jest opis przeprowadzonego badania nad efektywnością informacyjną warszawskiej GPW, zawarty w rozdziale IV.

Bibliografię pracy stanowią trzy grupy źródeł: artykuły opisujące konkretne badania, prace przekrojowe oraz strony internetowe. Opisy badań posłużyły podsumowaniu dotychczasowych osiągnięć w weryfikowaniu efektywności rynków oraz przeanalizowaniu warsztatu narzędziowego. Najważniejsze prace w tej kategorii należą do Johna Kozy oraz Shu-Heng Chena. Prace przekrojowe umożliwiły omówienie polemiki wokół koncepcji efektywności rynków oraz historii obliczeń ewolucyjnych. Najciekawsze prace należą do Michała Grotowskiego i Krzysztofa Wyroby oraz Sebastiana Buczka. Zawierają one cenne informacje na temat badań nad efektywnością informacyjną polskich rynków kapitałowych. Źródła internetowe dostarczyły informacji na temat narzędzi statystycznych oraz otwartego oprogramowania.

## ROZDZIAŁ I

### Efektywność informacyjna rynku

W latach 60 ubiegłego stulecia sformułowano koncepcję efektywności informacyjnej rynku, mówiącą o sposobie kształtowania się cen w zależności od dostępnej jego uczestnikom informacji. Teoria ta szybko zyskała zwolenników, jednak równie szybko pojawiła się fala jej krytyki. Dyskusja pomiędzy zwolennikami a przeciwnikami koncepcji efektywności informacyjnej rynków wciąż się toczy. Z każdą pracą przybywa wiedzy na ten temat, jednak wciąż brakuje jednoznacznych odpowiedzi na fundamentalne pytania – czy i na ile rynki mogą być efektywne oraz które są. Z powodu licznych wątpliwości na temat, czy pełna efektywność na rynku w ogóle może mieć miejsce, często koncepcję tę nazywa się **hipotezą rynków efektywnych**.

W tym rozdziale przedstawiona zostanie historia badań nad efektywnością informacyjną rynków kapitałowych. Zaprezentowany zostanie proces doprecyzowania jej definicji, aż do obecnie obowiązującej postaci wraz z trzema jej odmianami – silnej, półsilnej i słabej. Zaprezentowane zostaną również najciekawsze kierunki badań związane z tym pojęciem. Omówione zostaną również przykładowe badania: Chena, ze względu na nowatorską metodę, jaką zastosował oraz badania nad rynkiem polskim.

#### 1.1. Historia koncepcji efektywności informacyjnej

Termin rynku efektywnego informacyjnie pojawił się w ekonomii w latach sześćdziesiątych XX wieku za sprawą amerykańskiego ekonomisty, Eugeniusza Famy (zob. [Beechey et al., 2000]). Zdefiniował on jako efektywny taki rynek, który szybko dostosowuje się do nowych informacji. Dwadzieścia lat później Fama zmodyfikował swoją definicję, stwierdzając, że szybkie dostosowanie to nie jedyny składnik badanego zjawiska. Nowa definicja mówiła, że rynek jest efektywny, jeżeli uwzględnia całą dostępną informację. Oznacza to, że rynki efektywne zachowują się w sposób racjonalny – istotne informacje nie są pomijane i nie ma systematycznego popełniania błędów przez inwestorów. Kolejne uwagi do definicji wniósł Jensen w 1978 roku. Zauważył on, że należy uwzględnić koszt dostępu do informacji. Zaproponował definicję słabej efektywności rynku, w myśl której rynek odzwierciedla każdą taką informację mającą na niego wpływ, której koszt pozyskania jest niższy niż jej krańcowa użyteczność.

Jednym z elementów, który sprawił, że koncepcja efektywności rynków kapitałowych

stała się popularna, były konsekwencje wpływające z ewentualnej efektywności danego rynku. Efektywność informacyjna może powodować (zob. [Grotowski, Wyroba, 2004]):

- brak wpływu struktury pasywów firmy na jej wartość
- brak wpływu polityki dywidend na wartość firmy
- bezużyteczność analizy technicznej i fundamentalnej przy konstruowaniu portfeli inwestycyjnych
- nieistnienie strategii inwestycyjnej przynoszącej ponadnormalne zyski.

### 1.1.1. Formy efektywności

Obecnie uważa się (zob. [Grotowski, Wyroba, 2004]), że rynek efektywny to taki, na którym na podstawie informacji danego typu nie da się osiągać ponadnormalnych zysków. W zależności od typu informacji wyróżnia się trzy formy efektywności: słabą, półsilną oraz silną.

**Słaba efektywność** najmniej zakłada o dostępnej informacji. Mówi ona, że rynek jest słabo efektywny, gdy wycena aktywów odzwierciedla większość dostępnych publicznie informacji. W tym przypadku na podstawie tylko historycznych informacji o kształtowaniu się cen nie jest możliwe ich sensowne prognozowanie. Analiza techniczna nie daje zatem żadnych rezultatów. Więcej o informacji zakłada kolejna forma – **efektywność półsilna**. Ma ona miejsce wtedy, gdy rynek odzwierciedla wszystkie publicznie dostępne informacje – jawne raporty o stanie finansów firm, ich strategię, wyniki, plany. Analiza fundamentalna nie umożliwia zatem w tym przypadku osiągnięcia nadzwyczajnych zysków. Najbardziej rygorystyczna jest **efektywność silna**. Występuje ona wówczas, gdy rynek odzwierciedla absolutnie wszystkie istniejące informacje (tak publiczne, jak i na przykład raporty wewnętrzne firm). Przy tej formie efektywności rynku dostęp nawet do specjalistycznej drogiej wiedzy nie gwarantuje osiągnięcia ponadnormalnych zysków, a więc praca maklerów i menedżerów funduszy inwestycyjnych nie ma sensu.

Na pytanie, czy rynki kapitałowe mogą być silnie efektywne, odpowiedzieli Grossman i Stiglitz (zob. [Grossman, Stiglitz, 1980]). Stwierdzili oni, iż silnie efektywny rynek to konstrukcja teoretyczna i dowiedli, że nie może ona istnieć w rzeczywistym świecie. Obserwacje Grossmana i Stiglitz'a nie powodują jednak, że badanie efektywności rynków traci sens. Istnieją bowiem informacje o niezerowym koszcie pozyskania. Jeżeli na rynku są niewykorzystane informacje, to pojęcie efektywności wiązać się będzie z relacją między kosztem ich zdobycia a zyskiem jakie mogą przynieść. Jeżeli dostępne są informacje, z których posiadania można osiągnąć zwrot przewyższający koszty ich pozyskania, to mamy do czynienia z rynkiem



nieefektywnym. W przeciwnym jednak razie możemy mówić o silnej efektywności rynku.

### 1.1.2. Formalne definicje efektywności

Efektywność informacyjna rynku ma również swoje formalne definicje w języku matematyki. W tym podrozdziale zaprezentowany zostanie nurt formalizacji zbudowany na propozycji Fama, który zdefiniował efektywność poprzez wartość oczekiwaną względem zbioru informacyjnego. Przedstawione zostanie również podejście ogólniejsze (w sensie użytego aparatu matematycznego) dotyczące słabej efektywności a wychodzące od teorii  $\sigma$ -algebr zbiorów informacyjnych.

### Zbiory informacyjne i wartości oczekiwane

Eugeniusz Fama w 1970 roku zaproponował formalizację pojęcia efektywności rynku (zob. [Buczek, 2005]). Sformułował on szereg równań opisujących proces stochastyczny, który według niego realizuje się na rynku efektywnym informacyjnie. Zaczął od poniższego równania:

$$E(\hat{p}_{j,t+1}|\Phi_t) = [1 + E(\hat{r}_{j,t+1}|\Phi_t)] p_{j,t}$$

gdzie:

$p_{j,t}$  – cena  $j$ -tej akcji w okresie  $t$

$\hat{p}_{j,t}$  – zmienna losowa opisująca cenę akcji

$\hat{r}_{j,t}$  – zmienna losowa opisująca stopy zwrotu z akcji  $j$  w okresie  $t$

$\Phi_t$  – zbiór informacji w pełni odzwierciedlony w cenach akcji w okresie  $t$ .

Na rynku efektywnym informacyjnie nie da się w oparciu o zbiór informacji  $\Phi_t$  zbudować strategii inwestycyjnej, która pozwoliłaby na osiągnięcie stopy zwrotu większej niż w stanie równowagi. Ewentualną premię z takiej strategii na rynku nieefektywnym można oznaczyć następująco:

$$x_{j,t+1} = p_{j,t+1} - E(p_{j,t+1}|\Phi_t)$$

Przy tym oznaczeniu warunek efektywności rynku, czyli nieosiągnięcie ponadnormalnych zysków, opisać można równaniem:

$$E(\hat{x}_{j,t+1}|\Phi_t) = 0$$

Szybko jednak odkryto, że tak zapisany zestaw równań jest tautologią (jako jeden z

pierwszych wykazał to S. LeRoy w 1976 roku). Pierwsza próba sformalizowania teorii Fama okazała się więc fiaskiem.

Jeszcze w 1976 roku Fama zaproponował inne podejście do matematycznego zapisu swojej teorii. Zbudował je na rozdzieleniu pojęcia wartości oczekiwanej przez rynek ( $E_m$ ) od teoretycznej wartości oczekiwanej ( $E$ ). Ta ostatnia oparta jest o cały zbiór informacyjny  $\Phi_t$ . Ta pierwsza może z kolei ograniczać się do jego podzbioru – tak jak inwestorzy mogą nie bazować na rynku nieefektywnym informacyjnie na wszystkich dostępnych informacjach, a jedynie na ich części. Fama przedstawił następujące równanie:

$$E(\hat{p}_{j,t}|\Phi_{t-1}) = E_m(\hat{p}_{j,t}|\Phi_{t-1})$$

Buczek z kolei zaproponował zmodyfikowanie tej definicji. W oparciu o obserwacje zachowań inwestorów można stwierdzić, że rynek nie reaguje natychmiast a jedynie – niezwłocznie. Uczestnicy rynku potrzebują zazwyczaj określonego czasu, aby przetworzyć nowe informacje i zareagować na nie. Stąd zamiana równości na zbieżność:

$$E_m(\hat{p}_{j,t}|\Phi_{t-1}) \rightarrow E(\hat{p}_{j,t}|\Phi_{t-1})$$

Zaproponowane przez Buczkę równanie asymptotyczne jest jednak bardzo nieściśle. Zbieżność wymaga bowiem określenia zmiennej i horyzontu zbiegania. Intencją było oddanie obserwacji, że ceny na rynku zbiegają do poziomu równowagi wpływającego z pełnej informacji. Oczywiście, w kolejnym okresie rynek ściera się z nowym zbiorem informacyjnym i nowym poziomem równowagi, do którego zaczyna dążyć. W praktyce zatem nie jest możliwe osiągnięcie równowagi, bo ona sama zmienia się wraz z  $t$ .

Dokonana zostanie teraz próba doprecyzowania propozycji Buczka. W pierwszym kroku należy przede wszystkim doprecyzować  $E_m$  jako:

$$E_m(\hat{p}_{j,t}|\Phi_{t-1}) = E(\hat{p}_{j,t}|\Phi_{t-1}^m)$$

Oczekiwania cen akcji formułowane przez rynek są tak naprawdę oczekiwaniami budowanymi w oparciu o niepełny zbiór informacji. Inwestor dysponuje jedynie podzbiorem pełnej wiedzy ( $\Phi_t^m \subseteq \Phi_t$ ). Ideą, którą próbował wyrazić Buczek, była rzecz następująca: jeżeli w danym momencie  $t$  zamrozimy zbiór wszystkich informacji, na podstawie których kształtują się ceny ( $\Phi_t = \bar{\Phi}$ ), to informacja, w oparciu o którą inwestorzy kształtują swoje oczekiwania co do cen będzie wraz z upływem czasu zbiegać do zbioru  $\bar{\Phi}$ . Tym samym zaś oczekiwana

przez uczestników rynku cena będzie zbiegać do ceny równowagi, odzwierciedlającej pełną informację. Powyższe poprawki ująć można w następujący zestaw równań:

$$\begin{aligned}
 E_m(\hat{p}_{j,t}|\Phi_{t-1}) &= E(\hat{p}_{j,t}|\Phi_{t-1}^m) \\
 \Phi_t^m &\subseteq \Phi_t \\
 \forall_t \quad \Phi_t = \bar{\Phi} &\Rightarrow \lim_{t \rightarrow \infty} \Phi_t^m = \bar{\Phi}
 \end{aligned}$$

### **$\sigma$ -ciała informacji**

Wartość oczekiwana, używana przez Famy w poprzednim podrozdziale, jest całką Lebesgue'a zmiennej losowej (zob. [Jakubowski, Sztencel, 2001]). Całka taka określona jest dla pewnej miary – miary prawdopodobieństwa. Miara jest funkcją określoną na przestrzeni mierzalnej, która jest  $\sigma$ -ciałem. To z kolei definiuje się następująco.

Podzbiór  $\Sigma$  zbioru potęgowego zbioru  $X$  jest  $\sigma$ -ciałem wtedy i tylko wtedy, gdy spełnia następujące trzy warunki:

- $\Sigma \neq \phi$
- $\forall E \in X \quad E \in \Sigma \Rightarrow X \setminus E \in \Sigma$  (jeżeli zbiór należy do  $\Sigma$ , to jego dopełnienie (w  $X$ ) też należy do  $\Sigma$ )
- $I \subset N \quad \forall_{i \in I} X_i \in \Sigma \Rightarrow \cup_{i \in I} X_i \in \Sigma$  (suma przeliczalnie wielu zbiorów z  $\Sigma$  też należy do  $\Sigma$ ).

Shu-Heng Chen (zob. [Chen, Yeh, 1996]) zaproponował odwołanie się w definicji efektywności informacyjnej bezpośrednio do  $\sigma$ -ciał (a nie do wartości oczekiwanych, jak w propozycji Famy). Mówi on o  $\sigma$ -ciałach informacji wygenerowanych przez historyczne kształtowanie się cen. Na rynku (słabo) efektywnym funkcja ceny powinna być niezależna od dowolnej funkcji Borelowskiej<sup>1</sup> określonej na  $\sigma$ -ciele historycznych informacji. Nie istnieje inny sposób zweryfikowania tak zdefiniowanej efektywności jak tylko poprzez sprawdzenie niezależności notowań od wszystkich funkcji Borelowskich. W praktyce jest to niewykonalne.

#### **1.1.3. Krytyka teorii**

Po początkowej fali entuzjazmu związanej z koncepcją rynków efektywnych według Famy (zob. [Buczek, 2005]), nadeszła fala refleksji i krytyki. Argumenty przeciw istnieniu

<sup>1</sup>Funkcja Borelowska to funkcja mierzalna na zbiorze Borelowskim, czyli zbiorze wszystkich zbiorów otwartych przestrzeni. Chenowi zdawało się chodzić zaledwie o funkcje określone na przedziałach  $(-\infty, t]$ , co odpowiadałoby funkcjom, które historycznym notowaniom cen aż do momentu  $t$  przypisują cenę w momencie  $t + 1$ .

efektywności na rynku skoncentrowane były (i wciąż są) wokół obserwowanych anomalii na rynkach kapitałowych. Przykładowe anomalie to:

- efekt stycznia – na początku roku ceny akcji zachowują się lepiej niż w innych miesiącach
- efekt przegrani-zwycięzcy – akcje spółek osiągających najgorsze wyniki przeważnie osiągają wysokie wzrosty w kolejnych okresach
- pierwsze oferty publiczne – wiążą się z wysokim wzrostem akcji spółek w okresie ich debiutu i wycofaniem się inwestorów w późniejszych okresach.

Większość prac krytyków koncepcji efektywności informacyjnej wiąże się z ideą finansów behawioralnych (zob. [Buczek, 2005]). Teoria ta opiera się na spostrzeżeniu, że ludzie nie działają w tak wyidealizowany sposób, jaki o nich zakłada wizja homo oeconomicus. Nie jest prawdą, że każdy z nas stosuje wyrafinowany warsztat matematyczny do podejmowania wszystkich ekonomicznych decyzji w ciągu dnia. Zawsze istnieją bowiem jakieś ograniczenia – np. czasowe lub technologiczne. Herbert Simon określił postępowanie człowieka mianem satysfakcjonowania<sup>2</sup>, oznaczającym poszukiwanie wystarczająco dobrych rezultatów – zadawalających i osiągalnych, ale niekoniecznie najlepszych istniejących. Dalej, Amos Tversky i Daniel Kahneman zdefiniowali trzy grupy heurystyk postępowania homo oeconomicusa: dostępność (odwoływanie się do podobnych doświadczeń z przeszłości), reprezentatywność (ocena na ile obserwowane zjawisko odpowiada większej całości, regule, populacji) oraz zakotwiczenie (przewidywanie przyszłych wartości na podstawie zasugerowanych z przeszłości wielkości). Wszystkie te trzy heurystyki zostały rzeczywiście zaobserwowane w trakcie badań na uczestnikach rynków kapitałowych i dowodzą, iż nie działają oni w sposób racjonalny ani efektywny. W konsekwencji, według zwolenników teorii behawioralnych, nie można oczekiwać od inwestorów pełnej racjonalności, a od rynku – efektywności.

#### **1.1.4. Sposoby sprawdzania efektywności rynku**

W tym podrozdziale przedstawione zostaną sposoby sprawdzania czy dany rynek jest efektywny dla każdej z trzech form efektywności.

##### **Słaba efektywność**

Sebastian Buczek [2005] podaje trzy sposoby weryfikacji słabej efektywności rynku. Podstawowy, wprost wiążący się z definicją, to sprawdzanie, czy analiza techniczna dostarcza wiarygodnych prognoz. W tym celu buduje się różne strategie analizy technicznej, m.in. śred-

<sup>2</sup>Simon nazwał to zjawisko *satisficing*, co jest skrzyżowaniem angielskich słów *satisfy* – zaspokoić oraz *suffice* – wystarczyć.

nie ruchome (proste, wykładnicze, adaptacyjne), średnie kroczące z filtrem, czy oscylatory. Następnie ocenia się jakość prognoz dostarczanych przez te metody. Jeżeli prognozy analizy technicznej są wiarygodne, rynek nie będzie słabo efektywny. Należy odnotować, iż do budowania prognoz można używać o wiele większego warsztatu niż tylko analizy technicznej. Przykłady takich badań znajdują się w rozdziale 1.3.

Innym sposobem weryfikacji słabej efektywności jest badanie, czy ciąg cen jest realizacją procesu błędzenia losowego. Do tego celu służą takie testy statystyczne, jak autoregresji, serii, pierwiastków jednostkowych, czy ilorazy wariancji. Jeżeli szereg cen nie jest procesem błędzenia losowego, rynek nie będzie słabo efektywny.

Kolejny sposób opiera się na badaniu rozkładu stóp procentowych w czasie – np. w rozkładzie na dni tygodnia, tygodnie miesiąca, czy miesiące w roku. W tym podejściu poszukuje się jakiejś regularności (jak np. efekt piątku). Jeżeli uda się zidentyfikować jakieś reguły, rynek nie będzie słabo efektywny.

### **Półsilna efektywność**

Weryfikację półsilnej efektywności rynku można przeprowadzić za pomocą testów strategii inwestycyjnych lub testów zdarzeń.

Na rynkach o półsilnej efektywności należy spodziewać się nieskuteczności strategii inwestycyjnych, opierających się na analizie fundamentalnej. Takie zabiegi jak na przykład zbudowanie portfela ze spółek dobranych według kapitalizacji rynkowej, wskaźnika ceny do wartości księgowej, czy ceny do zysku netto, nie powinny przynieść nadzwyczajnych zysków.

Testy zdarzeń z kolei polegają na zebraniu spółek, u których wystąpiło jakieś konkretne wydarzenie – np. nieoczekiwane zmiany wyników finansowych spółek, czy korygowanie prognoz finansowych. Metoda zakłada precyzyjne wyznaczenie momentu pojawienia się danej informacji oraz dokładne zbadanie zachowania się cen walorów spółki przed, w trakcie i po jej publikacji. Im szybciej nastąpi internalizacja tej informacji (uspokojenie się kursu), tym rynek jest bardziej efektywny w ujęciu półsilnym.

### **Silna efektywność**

Ten rodzaj efektywności jest najbardziej kontrowersyjny i najtrudniejszy w zbadaniu, jako że jego definicja odwołuje się do wiedzy specjalistycznej i informacji niedostępnych szerszemu gronu.

Jednym ze sposobów na jej badanie jest śledzenie wyników funduszy inwestycyjnych.

Zakłada się, że fundusze zatrudniają dobrze przeszkolonych ludzi o ogromnych umiejętnościach, którzy posiadają ponadprzeciętną wiedzę i mają dostęp do wielu informacji. W związku z tym można się spodziewać, że wyniki funduszy będą ponadprzeciętne. Jednak Buczek uważa, że nie jest w praktyce możliwe zweryfikowanie silnej efektywności rynku. W swojej pracy podważa stosowane dotychczas metody, w szczególności śledzenie wyników funduszy inwestycyjnych. Badania nad ich wynikami są bowiem *de facto* badaniami nad jakością pracy inwestorów poszczególnych funduszy.

Buczek zauważa również, że praca funduszy nie może się opierać o informacje niejawne, których wykorzystywanie byłoby prawdopodobnie niezgodne z prawem. Więcej na temat pracy funduszy inwestycyjnych znajduje się w rozdziale 1.2.

## **1.2. Problemy badania efektywności**

Beechey et al. [2000] dokonują przeglądu podejść do badania, czy dany rynek finansowy jest efektywny. Opisują najczęstsze problemy i wątpliwości badaczy. Podsumowanie ich obserwacji znajduje się poniżej.

### **Efektywność rynku a błędzenie losowe**

Ceny dóbr na rynkach efektywnych powinny zmieniać się losowo w odpowiedzi na nieoczekiwane informacje. Ceny aktywów mogą wykazywać pewne trendy, niemniej nawet wówczas fluktuacje wokół trendu powinny być czysto losowe. Stąd mówi się, że ceny aktywów na rynkach efektywnych zachowują się jak błędzenie losowe.

Uważa się, że hipoteza o błędzeniu losowym cen na rynkach efektywnych informacyjnie jest prawdziwa. Wprawdzie możliwość przewidzenia zachowania się cen istnieje, to jednak jest ona tak ograniczona i obciążona tak dużym błędem, że nie przynosi żadnej korzyści.

Kwestia błędzenia losowego kursów akcji jest intensywnie eksploatowana przez ekonomistów. Campbell, Lo i MacKinley zbadali w 1997 roku amerykański rynek akcji. Udało im się wyjaśnić zaledwie 12% zmienności kursów. Zaobserwowali jednak, że portfele małych giełd są lepiej przewidywalne niż większych.

### **Szybkość uwzględniania nowych informacji**

Badania Famy z lat sześćdziesiątych XX wieku pokazały, że ceny akcji gwałtownie reagują na pojawienie się nowych informacji (np. fuzje, zmiany w strategii wypłat dywidend), jednak już po kilku dniach powracają do swych zwykłych zachowań. Warto nadmienić, iż to

odkrycie przysporzyło koncepcji efektywności rynków dużego zainteresowania wśród ekonomistów.

Jednocześnie zaobserwowano (prace Balla i Browna z 1968 roku, Chana i in. z 1996 roku, Famy z 1998 roku), że ceny reagują na informacje o niespodziewanym zysku lub stracie firmy z pewnym charakterystycznym opóźnieniem.

### **Możliwość przewidywania przyszłych zysków**

W myśl hipotezy rynków efektywnych informacyjnie, wszelkie informacje na temat aktywów powinny być już uwzględnione w ich wycenie. Żadna publicznie dostępna informacja nie powinna umożliwiać osiągnięcia ponadnormalnych zysków. Istnieje jednak kilka anomalii, które mogą temu przeczyć.

Jednym z przykładów takiego zjawiska jest analiza techniczna, czyli wynajdywanie w notowaniach historycznych pewnych regularności i przedłużanie ich na przyszłość, dokonując w ten sposób prognoz. Istnieje dużo tanich i prostych metod analizy technicznej. Można je łatwo stosować. Jak pokazują liczne badania na rynku walutowym dolara amerykańskiego (np. Levicha z 1993 roku, Osleta z 1995 roku, Neely'ego z 1997 roku), umożliwiają one osiągnięcie systematycznych zysków. Na giełdach wyniki nie są już jednoznacznie pozytywne. Badania, np. Allena i Karjalainena z 1999 roku, pokazują, że analiza techniczna nie umożliwia uzyskania przewagi w inwestowaniu. Można uznać, iż wytłumaczeniem tego, że analiza techniczna nie przynosi istotnych korzyści jest sam fakt jej istnienia. Skoro korzysta z publicznie dostępnych informacji, wiedza na temat jej metod jest również powszechna, jest ona dość popularna wśród analityków rynku. A skoro większość z niej korzysta i podejmuje decyzje bazując na wynikach analizy technicznej, to kształtujące się na rynku ceny są odzwierciedleniem wszystkich informacji i tym samym analiza techniczna nie daje możliwości osiągnięcia ponadnormalnych zysków.

Innym zjawiskiem stawiającymi pod znakiem zapytania hipotezę o efektywności rynków jest np. zależność między rozmiarem firmy czy rynku a ich wynikami (obserwuje się, że im mniejsza firma tym większa jej rentowność).

Obronić koncepcję efektywności rynków przed wspomnianymi zjawiskami można dokonując kilku obserwacji. Przede wszystkim – korzyści wynikające z sugerowania się zaobserwowanymi prawidłowościami są i tak niewielkie: przewidywalność nie jest zbyt duża, a zyski niewysokie. Po drugie – uzasadnieniem może być zjawisko podkradania danych (ang. *data-snooping*). Przeprowadzone zostało badanie (zob. [Lo, MacKinlay, 1990]), z którego wynika, że większość prac naukowych w Stanach Zjednoczonych opiera się na tym samym

zestawie danych. W związku z tym niektóre obserwacje mogą być po prostu pochodnymi specyficznych właściwości tego konkretnego zestawu danych. Aby uwiarygodnić czynione obserwacje, należałoby weryfikować obserwacje na różnych próbkach.

### **Wyniki funduszy inwestycyjnych**

Fundusze inwestycyjnie aktywnie inwestują kapitał. Zatrudniają dobrze wyszkolonych, doświadczonych menedżerów, którzy za swoje wynagrodzenie starają się być lepsi od innych – dotrzeć do ukrytych informacji, przewidzieć zachowanie się rynku na podstawie własnego doświadczenia i intuicji. W związku z tym są doskonałym materiałem do rozważań w temacie efektywności rynków, zwłaszcza w kontekście wpływu drogiej eksperckiej wiedzy (opłacenie menedżerów) na ponadnormalne zyski (na jakie liczą klienci funduszy inwestycyjnych).

Beechey i inni dokonują szeregu interesujących obserwacji na ten temat. Przede wszystkim, oceniając jednocześnie wszystkie fundusze i uwzględniając koszty ich prowadzenia, nie osiągają one nadzwyczajnych zysków. Po odjęciu od ich rentowności kosztów wynagrodzenia menedżerów pozostają zyski netto zbliżone do osiąganych przez przeciętnych indywidualnych pasywnych inwestorów. Co gorsza, w latach pięćdziesiątych i sześćdziesiątych XX wieku (badania Sharpe'a z 1966 roku i Jensena z 1968 roku) zyskowność netto funduszy była gorsza niż proste pasywne strategie inwestycyjne. W latach 80 zaobserwowaną znaczną poprawę efektywności funduszy, nie na tyle jednak dużą, żeby netto wypadały one lepiej.

Należy jednak zauważyć, że niektóre fundusze, rozpatrywane indywidualnie, są mimo wszystko zdolne do systematycznego osiągania lepszych wyników niż konkurencja. Badania (np. Chevaliera z 1996 roku) pokazują, że młodzi, lepiej wykształceni menedżerowie są zdolni do osiągania lepszych wyników. Pozostaje jednak pewną zagadką, czemu gorsze fundusze inwestycyjne utrzymują się na rynku. Możliwym uzasadnieniem mogą być trudności, jakie mają ich klienci, w ocenie, które fundusze są lepsze, a które gorsze.

Podsumowując należy stwierdzić, że istnienie i wyniki funduszy inwestycyjnych pozostaje w zgodzie z hipotezą rynków efektywnych informacyjnie. Wiedza, pozwalająca na osiągnięcie ponadnormalnych zysków jest kosztowna i jej opłacenie niweluje te nadzwyczajne zyski.

### **Konsekwentne złe wyceny**

Kolejną kwestią godną rozważenia w kontekście hipotezy rynków efektywnych informacyjnie jest sposób kształtowania się cen, a dokładniej – czy długookresowo aktywa wyceniane



są spójnie z dostępną informacją.

Warto zauważyć, iż fakt, że menedżerowie funduszy inwestycyjnych nie osiągają systematycznych ponadnormalnych zysków świadczyć może o tym, że generalnie aktywa są dobrze wyceniane przez rynki. Nie jest to jednak dowód. W rzeczywistości zdarza się, że aktywa są systematycznie źle wyceniane. Zidentyfikować można trzy tego przykłady.

Pierwszy związany jest z wyceną funduszy zamkniętych. Obserwuje się (m.in. badania Lee, Shleifera i Thaler z 1990 roku), że jednostki uczestnictwa w funduszach zamkniętych są systematycznie różne od ceny wynikającej z koszyka aktywów, w jakie one same inwestują. W początku działalności wyceniane są powyżej, a po pewnym czasie – poniżej wartości zbudowanego portfela. Jednym wyjaśnieniem jest odbicie w tej różnicy wycen oczekiwań co do przyszłych zysków z funduszu.

Drugi dotyczy warunków ekstremalnych – kryzysów. Zazwyczaj wówczas, gdy ceny akcji rosną, inwestorzy szybko starają się dokupić więcej akcji, w nadziei, że ich ceny będą wciąż rosły a oni zdążą zrealizować zyski przed załamaniem się rynku (tzw. bąble spekulacyjne). Badania licznych momentów krytycznych na giełdach pokazują, iż możliwe są nawet długie (kilkumiesięczne) okresy, w których wycena akcji jest niezwiązana z racjonalną ich oceną.

Trzeci, ostatni przykład dotyczy rynków walutowych. W zasadzie nie istnieje dobre wyjaśnienie zachowania się kursów wymiany walut w długim i krótkim okresie. Są one bardzo płynne, zmienne. Są one w związku z tym niewłaściwie wyceniane.

### **1.3. Przykładowe badania słabej efektywności rynku**

Sebastian Buczek wysnuwa hipotezę, że dojrzałe rynki finansowe powinny być efektywne informacyjnie w ujęciu słabym. Nawet prawdziwość tej tezy nie odbiera sensu badaniu słabej efektywności. Jednocześnie bowiem bada się stopień dojrzałości danego rynku.

W tym podrozdziale przedstawione zostaną dwa szczególne badania, stawiające sobie za cel zbadanie słabej efektywności rynków. Ich szczególność polega na użytych metodach. Pierwsze z nich opiera się na zastosowaniu pojęcia złożoności stochastycznej. Drugie próbuje zbudować model opisujący rynek akcji, stosując do tego metody programowania genetycznego. W podrozdziale przedstawiony zostanie również przegląd prac na temat słabej efektywności warszawskiej giełdy.

### 1.3.1. Szukanie regularności w danych

Pierwszym badaniem, które zostanie omówione, jest ciekawa analiza efektywności 13 światowych giełd (zob. [Shmilovici et al., 2003]). Autorzy zdecydowali się wykorzystać kryterium regularności danych (MDL, ang. *Maximum Description Length*) do zweryfikowania hipotezy efektywności rynku w postaci słabej. Kryterium to (zob. [Grünwald, 2005]) wywodzi się z obserwacji, że w danych są regularności a poznawanie tych regularności odpowiada procesowi uczenia się. Im więcej regularności uda się odkryć w danych, tym lepiej je można je zrozumieć i tym zwięźlej można je reprezentować. W oryginalnej postaci (zaproponowanej w 1965 roku przez Kolmogorova) MDL danych określana była jako długość najkrótszego programu komputerowego, który potrafi wypisać te dane. Najprostszym będzie zawsze program zawierający w sobie w sposób jawny wszystkie dane i je wypisujący. Sprytniejsze programy będą wykorzystywały regularności w danych, przez to będą krótsze. Programy, o których mówił Kolmogorov, są po prostu programami kompresującymi dane. W 1996 roku Rissanen dokonał przededefiniowania kryterium. Określił on złożoność stochastyczną (ang. *stochastic complexity*) jako miarę na przestrzeni dostępnych modeli. Miara ta jest tym większa (gorsza) im dłuższa jest reprezentacja danych wejściowych przez najlepszy z dostępnych modeli. Zatem, im większa złożoność stochastyczna, tym bardziej dane są nieregularne, losowe, trudniejsze do zrozumienia i opisanie, trudniej jest uchwycić prawidłowości nimi rządzące. Zespół Shmilovici'ego spodziewał się, że im większa efektywność rynku, tym bardziej powinien on zachowywać się losowo i tym większa powinna być jego złożoność stochastyczna.

Autorzy badania zajęli się zbadaniem szeregów czasowych 13 giełd światowych (m.in. Dow-Jones, FTSE, i DAX). Badanie stochastycznej złożoności prowadzili dla podserii długości 50, 75 i 100 dziennych obserwacji stóp zwrotu. Dla każdej długości i dla każdego ciągu liczyli złożoność stochastyczną. Następnie wyniki były uśredniane dla danej długości ciągu.

Po zestawieniu wyników okazało się, że aż 10 z 13 giełd wykazuje ponadprzeciętną<sup>3</sup> kompresowalność danych. Do grupy tej należą przede wszystkim małe (w ocenie autorów) giełdy, takie jak np. BRSP (Brazylia), SSMI (Szwajcaria), czy BFX (Belgia). Wysoka kompresowalność oznacza niską złożoność stochastyczną i tym samym może być interpretowana jako nieefektywność tych giełd. Z kolei duże rynki (jak np. holenderski AEX czy hongkongski HSI) wykazują bardzo wysoką złożoność stochastyczną, co może świadczyć o ich efektywności.

---

<sup>3</sup>Za punkt odniesienia autorzy przyjęli złożoność stochastyczną realizacji procesu błędzenia losowego.

### 1.3.2. Zastosowanie programowania genetycznego

W tym podrozdziale zaprezentowane zostanie badanie używające do modelowania metod programowania genetycznego (zob. [Chen, Yeh, 1996]). Pracę ta będzie punktem wyjścia do nowego badania, zaprezentowanego w rozdziale IV.

Programowanie genetyczne<sup>4</sup> zostało użyte do konstruowania modeli opisujących ceny akcji. Tradycyjne modele liniowe nie dają na tym polu wyników lepszych niż procesy błędzenia losowego. Programowanie genetyczne jest jednak zdolne do produkowania modeli o praktycznie dowolnej formie funkcyjnej, nie tylko liniowych. Autorzy przywołują badania sugerujące, iż zależności na rynku akcji mogą być nieliniowe (m.in. prace Savita z 1988 i 1989 roku, Hinicha i Pattersona z 1989 roku, Hsieha z 1989 roku). Na tej podstawie ugruntowany był ich optymizm poznawczy i cel pracy – zbudowanie modelu osiągającego lepsze wyniki niż błędzenie losowe.

Przedmiotem badania były giełdy Tajwanu i Stanów Zjednoczonych. Badacze mieli dostęp do dziennych stóp zwrotu z tajwańskiego indeksu TAIEX (ang. *Taiwan Stock Price Index*) oraz amerykańskiego indeksu S&P 500 (ang. *Standard and Poor*) dla okresu od 1. maja 1974 roku do 27. stycznia 1997 roku. Przekłada się to na 6677 obserwacji z rynku tajwańskiego i 5831 dla amerykańskiego. Do modelowania użyto jednak jedynie fragmentu danych – 50 obserwacji. Wybór akurat takiej długości szeregu został dokonany arbitralnie i wytłumaczony, że wprawdzie istnieją przesłanki na rzecz tezy, że długość szeregu może mieć znaczenie dla jakości modelu budowanego na jego podstawie, to jednak ich badanie ma charakter czysto pogładowy (główny cel to ilustracja użycia nowej metody). Autorzy poświęcili dość dużo uwagi na odpowiedni wybór szeregu 50 obserwacji ze wszystkich dostępnych. Uznając, iż wybór może mieć ogromne znaczenie, zwłaszcza przy przyjętej niewielkiej długości, postanowili użyć kryterium stochastycznej złożoności Rissanena, które zostało omówione w rozdziale 1.3.1.

Na podstawie kryterium MDL, autorzy wybrali szereg 27.11.1990-30.01.1991 dla TAIEX i 06.05.1977-19.07.1977 dla S&P 500. Do wybranych szeregów zastosowano okres *post-sample* długości 5 obserwacji, które posłużyły weryfikacji własności prognostycznych modelu. Autorzy chcieli przekonać się, czy modele budowane przez algorytm ewolucyjny są w stanie dawać lepsze wyniki niż proces błędzenia losowego. Podstawowym kryterium oceny było dla nich kryterium MAPE (ang. *Mean Absolute Percentage Error*, czyli średni bezwzględny błąd procentowy). Dla procesu błędzenia losowego wynosi on 1. Autorzy policzyli również MAPE

---

<sup>4</sup>Szczegółowy opis programowania genetycznego znajduje się w rozdziale 2.3.

dla modeli liniowych autoregresyjnych (proces AR dla od 1 do 10 opóźnień). Okazało się, że wszystkie te modele osiągają MAPE powyżej 1, często dość znacznie od tej liczby odbiegając. Algorytm ewolucyjny pracuje nad populacją modeli. W każdym kroku symuluje się operacje ewolucyjne, zmieniając populację. Autorzy ustalili wielkość populacji na 500, dzięki czemu w każdym kroku działania algorytmu mieli zbudowanych 500 modeli. Oczywiście wraz z postępem ewolucji, modele te stają się na ogół coraz lepsze.

Autorzy badania za jeden z celów obrali zastosowanie miary mogącej określić stopień prawdziwości hipotezy na danym rynku. Niezadowoleni z wcześniejszych badań innych naukowców, które to praktycznie nigdy nie dawały jednoznacznej odpowiedzi, postanowili wprowadzić miarę, której wartości można by porównywać pomiędzy rynkami i okresami. Temu właśnie celowi posłużyło wprowadzenie dwóch miar:

- $\pi_1(n)$  jako prawdopodobieństwo, że algorytm ewolucyjny w  $n$ -tym pokoleniu będzie lepszy niż proces błędzenia losowego. Oznacza to po prostu liczbę modeli z tych istniejących w pokoleniu  $n$ , których MAPE dla okresu *in-sample* jest mniejsze niż 1
- $\pi_2(n)$  – analogicznie jak  $\pi_1(n)$ , z tym że dla okresu *post-sample*.

Miary te mówią w sposób obiektywny o stopniu nieprzewidywalności danego rynku. Im mniejsza ich wartość, tym trudniejsze konstruowanie prognoz na rynku.

Podsumowanie wyników znajduje się w tabeli 1. Należy przede wszystkim zauważyć, że wyniki są dużo lepsze niż procesu autoregresyjnego (powyżej 1) i błędzenia losowego (1). Są tego dwie konsekwencje. Po pierwsze być może istnieje jednak sposób modelowania giełdy. Po drugie – prezentowane indeksy w wybranym okresie mogą nie być rynkami efektywnymi informacyjnie. Kolejnym wnioskiem, wypływającym z zebranych wyników jest to, że algorytm ewolucyjny jest bardzo podatny na problem zbytowego dopasowania do danych. W miarę bowiem kolejnych pokoleń, MAPE na okresie *in-sample* jest coraz lepsze. Jednak dla okresów kontrolnych (*post-sample*) – coraz gorsze. Modele budowane przez algorytm ewolucyjny mają zatem tendencję do wyuczania się danych testowych. Skutkuje to dobrym dopasowaniem do nich, lecz ograniczonymi możliwościami przewidywania. Ostatnim wnioskiem zespołu Chena była konstatacja, że zastosowanie programowania genetycznego w modelowaniu ekonomicznym, a giełd w szczególności, ma ogromny potencjał i warto jest stosowania. Sugestia ta zostanie wykorzystana w rozdziale IV, gdzie znajduje się opis nowego badania – próby zastosowania programowania genetycznego do weryfikacji hipotezy o efektywności rynku na warszawskiej giełdzie.

Tabela 1. Zestawienie wartości miar  $\pi$  dla indeksów giełdy tajwańskiej i amerykańskiej wg badań Chena i Yeha

Symbol miary	Wartości [%] miar dla indeksów	
	TAIEX	S&P 500
$\pi_2(50)$	48,6	56,9
$\pi_2(100)$	41,7	56,9
$\pi_2(150)$	37,5	47,2
$\pi_2(200)$	34,7	40,3

Źródło: Opracowanie własne na podstawie [Chen, Yeh, 1996].

#### 1.4. Efektywność rynku w Polsce

Nim dokonane zostanie nowe badanie na temat efektywności polskiego rynku, należy dokonać przeglądu dotychczasowych badań w tym temacie. Poniżej zaprezentowane zostaną po krótko najważniejsze badania.

##### Efektywność informacyjna polskiego rynku walutowego

Grotowski, Wyroba [2004] dokonali badania empirycznego polskiego rynku walutowego. Sprawdzili oni, czy kursy walut mogą być realizacją błędzenia losowego, by w ten sposób zweryfikować efektywność informacyjną tych rynków. Na wstępie badania zauważyli, że możliwość weryfikowania efektywności informacyjnej polskiego rynku kapitałowego dopiero się rozwija. Wynika to z jego stosunkowo młodego wieku. Badania, na które się powołują, nie dostarczyły przypadków wyraźnej nieefektywności, nie dały jednak również jednoznacznej odpowiedzi na pytanie o stopień ich efektywności.

Autorzy zajęli się szeregami czasowymi dziennych notowań logarytmów kursów dolara amerykańskiego do złotówki (USD/PLN) i marki niemieckiej do złotówki (DEM/PLN) za okres od stycznia 1996 roku do grudnia 2001 roku. Do szeregów zastosowano test ilorazu wariancji oraz testy pierwiastków jednostkowych. Test pierwiastków jednostkowych nie dostarczył żadnych argumentów przeciwko hipotezie zerowej stwierdzającej, że wspomniane notowania kursów walutowych są realizacją procesu błędzenia losowego. Nie podważył on zatem hipotezy o efektywności informacyjnej tych rynków walutowych. Również test ilorazu wariancji dla kursu USD/PLN skłonił autorów do uznania, że model błędzenia losowego dobrze opisuje te notowania. Jednak podobna analiza w odniesieniu do kursu DEM/PLN skłania do odrzucenia hipotezy o realizacji przez niego procesu błędzenia losowego. Badaczom nie udało się jednak znaleźć przyczyn tego zjawiska.

## Słaba efektywność warszawskiej Giełdy Papierów Wartościowych

Buczek [2005] dokonał w swojej pracy przeglądu badań nad efektywnością polskiego rynku kapitałowego. Poniżej zaprezentowane zostaną najważniejsze omówione przez niego badania oraz wnioski.

Najważniejsze badania, które próbują zweryfikować słabą efektywność warszawskiej giełdy poprzez sprawdzenie skuteczności strategii analizy technicznej, zostały przeprowadzone przez J. Czekaja, M. Wosia i J. Żarnowskiego w 2001. Badania te prześledziły strategie średnich ruchomych, kroczących, oscylatory i inne na warszawskiej giełdzie w latach 1994-2000. Udało im się zbudować dość dobre strategie, tj. takie, których poziom ryzyka (mierzony odchyleniem standardowym) był niższy niż referencyjnej strategii (inwestowanie w WIG), a oczekiwane stopy zwrotu – wyższe. Statystyczna istotność tych wyników nie była jednak wysoka. Wybrany specyficzny okres (burzliwa druga połowa dekady lat dziewięćdziesiątych XX wieku) sprawia, że nie można z tych obserwacji wprost wyciągnąć wniosku o nieefektywności polskiej giełdy. Dodatkowy argument przeciw związany był z wykorzystaniem analizy technicznej – autorom udało się ustalić optymalne parametry dla poszczególnych metod, jednak jest to możliwe jedynie *ex post*.

Kolejny sposób badania słabej efektywności to sprawdzenie, czy notowania są realizacją procesu błędzenia losowego. Służą do tego testy autoregresji, serii, obecności pierwiastków jednostkowych, ilorazy wariancji oraz testy teorii chaosu. W licznych badaniach Szyszki oraz Czekaja, Wosia i Żarnowskiego, potwierdza się wizja notowań akcji jako realizacji procesu błędzenia losowego. Wątpliwości pojawiają się jedynie na przełomie 1993 i 1994 roku, co tłumaczy się ówczesnym bardzo silnym trendem wzrostowym. Ogólnie zatem, z punktu widzenia błędzenia losowego, hipoteza o efektywności warszawskiej giełdy zdaje się być prawdziwa.

Adam Szyszka przeprowadził również badania układu stóp zwrot w czasie. Przeanalizował on przeciętne tygodniowe oraz dzienne zmiany stóp w latach 1994-99. Udało mu się zidentyfikować efekt pierwszego tygodnia miesiąca, kiedy zaobserwował największe stopy wzrostu oraz efekt trzeciego tygodnia, kiedy większość spółek odnotowywała spadki. Zależności te wyszły wprawdzie statystycznie istotne, ale badania Buczka nie potwierdziły tych obserwacji dla lat 2001-2004. Kolejne badania Szyszki pozwoliły mu zaobserwować efekt poniedziałku (wyróżne wzrosty). Warto odnotować, iż efekt poniedziałku realizował się w tych badaniach dokładnie odwrotnie niż w przypadku giełdy nowojorskiej, gdzie początek tygodnia oznacza spadki. Zależność ta okazała się statystycznie istotna. Jednak również w

tym wypadku analiza późniejszego okresu przez Buczka nie potwierdziła tych obserwacji. W dodatku, sama obserwacja nie przedstawia mimo wszystko zbyt dużej wartości dla inwestujących – strategia kupowania w piątek i sprzedawania w poniedziałek może nie przynosić zysków pozwalających choćby opłacić koszty transakcji.

Według Buczka z dotychczasowych badań wynika słaba efektywność informacyjna warszawskiej giełdy. Co więcej, zaobserwowane na początku jej istnienia pewne anomalie (jak efekt pierwszego tygodnia miesiąca) zanikają, co świadczy o dojrzewaniu polskiego rynku kapitałowego.

## ROZDZIAŁ II

### Ewolucyjne modelowanie ekonomiczne

Modelowanie ekonomiczne kojarzy się przede wszystkim z regresją liniową. Oczywiście, nowoczesna ekonometria to dużo więcej niż ten prosty model, jednak od niego wszystko się zaczęło. To on był podstawą do zbudowania wszystkich innych. Modele mają jedną wspólną ważną cechę – wymagają od badacza wykazania się własną wiedzą na temat zjawiska. Ma to swoje dobre i złe strony. Naukowiec może opierając się na własnej wiedzy i doświadczeniu wybrać najwłaściwszy (według niego) dla danego problemu model. Z drugiej jednak strony musi *a priori* zdecydować się na określoną formę funkcyjną, ograniczając istotnie możliwości modelowania.

W rozdziale tym zaprezentowane zostanie stosunkowo młode i zdecydowanie odmienne podejście do modelowania – wykorzystanie metod ewolucyjnych. Zostanie szczegółowo omówiona koncepcja obliczeń ewolucyjnych: ich biologicznej motywacji, historii i obecnie stosowanych podejść. Następnie opisane zostanie jak obliczenia ewolucyjne mogą zostać wykorzystane w modelowaniu ekonomicznym. Ponieważ najwłaściwsze do tego jest programowanie genetyczne, zostanie ono szczegółowo opisane. W ostatniej części tego rozdziału omówione będą przykładowe badania, jakie przeprowadzono z wykorzystaniem metod programowania genetycznego.

#### 2.1. Obliczenia ewolucyjne

##### 2.1.1. Biologiczne fundamenty

Wśród biologicznych fundamentów leżących u podstaw obliczeń ewolucyjnych wyróżnić należy teorię ewolucji Karola Darwina oraz genetykę.

**Teoria ewolucji Darwina** podaje wyjaśnienie różnorodności biologicznej oraz odkrywa sterujące nią mechanizmy. Główną rolę w procesie ewolucji odgrywa mechanizm selekcji (zob. [Eiben, Smith, 2003]). W ograniczonym środowisku (tak przestrzennie jak i w zasoby) może przebywać ograniczona liczba osobników danego gatunku. Z drugiej jednak strony mamy do czynienia z biologicznym instynktem rozmnażania, mającym na celu zapewnienie istnienia gatunkowi. W tym momencie do głosu dochodzi właśnie mechanizm selekcji, który powoduje, że przetrwają jedynie najlepiej przystosowani do danego środowiska (ang. *survival of the fittest*).

„Przeżycie najbardziej przystosowanego” wymaga jednak dodatkowego komentarza,



gdyż – mimo swej ogromnej popularności – jest często źle interpretowane. Należy pamiętać, iż przeżywają nie najlepsi jak to się często mówi, lecz **względnie dobrzy**, jako że siłę napędową ewolucji stanowi różnorodność. Dodatkowo, to nie przeżycie osobnika jest jego ewolucyjnym celem, lecz **pozostawienie potomstwa**. Ewolucja nie kara śmiercią, lecz nagradza możliwością reprodukcji.

Drugim kluczowym elementem teorii Darwina są ciągłe zmiany i różnorodność. Zaobserwował on, że kolejne pokolenia osobników różnią się od poprzednich. Stwierdził, że w momencie reprodukcji zachodzą losowe, niewielkie zmiany. Selekcja naturalna powoduje, że premiiowane przeżyciem są jedynie najkorzystniejsze zmiany. Dzięki temu gatunki są w stanie przystosować się do zmieniającego się środowiska.

Warto w tym miejscu nadmienić, iż to nie Karol Darwin a Herbert Spencer był autorem i popularyzatorem terminów „ewolucja” oraz „przeżycie najbardziej przystosowanego” (zob. [Kwaśnicki, 1996]). To on jako pierwszy użył tych określeń. Darwin początkowo stosował nazwę „teoria doboru naturalnego” i dopiero za radą Wallace’a zaadaptował Spence-rowskie określenie.

**Genetyka** pozwoliła na posunięcie się krok dalej – przejście od podejścia makro do mikro (zob. [Eiben, Smith, 2003]). Genetyka stwierdza, że jednostka ma charakter dualny. Z jednej strony identyfikuje ją **genotyp** (czyli zestaw genów – funkcjonalnych jednostek dziedziczenia), z drugiej zaś strony – efekt ekspresji genów, czyli **fenotyp**. Rozróżnienie to pozwoliło na lepsze zrozumienie mechanizmów stojących za obserwowanymi przez Darwina zmianami pomiędzy kolejnymi pokoleniami danego gatunku. Lepsze dostosowanie do środowiska jest bowiem wynikiem zmian w genotypie jednostki. Zmiany te mogą być powodowane albo przez mutacje (czyli losowe, niewielkie modyfikacje genów), albo przez rekombinacje (np. poprzez chromosomowy *crossing-over* w czasie reprodukcji). O różnorodności biologicznej i przeżyciu decydują zatem geny jednostek. Richard Dawkins (zob. [Dawkins, 1996]), podążając tym rozumowaniem, wysnuwa wręcz hipotezę, że osobnik (w szczególności człowiek) jest jedynie narzędziem, służącym genowi do jego reprodukcji, powielenia, przetrwania.

Podstawowymi bytami świata ewolucji naturalnej są: środowisko, osobnik oraz dopasowanie. Tabela 2 pokazuje związek tych obiektów z fundamentami podejścia ewolucyjnego do programowania.

Ilustracją zaprezentowanych powiązań jest następujący przykład: problem (odpowiednik środowiska) znalezienia w grafie najkrótszej ścieżki łączącej dwa zadane punkty (np. znalezienie najkrótszej drogi z Warszawy do Amsterdamu). W tym problemie zanurza się kantydatów na rozwiązanie (tak jak w środowisku funkcjonują osobniki). Każda ścieżka przez

Tabela 2. Powiązanie między teorią ewolucji a programowaniem ewolucyjnym

Pojęcia teorii ewolucji	Pojęcia programowania ewolucyjnego
Środowisko	Problem
Osobnik	Kandydat na rozwiązanie
Dopasowanie	Jakość rozwiązania

Źródło: [Dawkins, 1996].

graf pomiędzy zadanymi punktami może być kandydatem na rozwiązanie (np. Warszawa-Gdańsk-Praga-Paryż-Amsterdam). Za wyznacznik jakości rozwiązania można przyjąć długość danej ścieżki. Im krótsza ścieżka, tym lepszy jest dany kandydat (i lepsze jego dopasowanie do środowiska) – ścieżka Warszawa-Genewa-Amsterdam (2650 km<sup>5</sup>) będzie lepsza niż Warszawa-Rzym-Paryż-Amsterdam (4000 km).

Idea programowania ewolucyjnego polega na symulowaniu działania naturalnej ewolucji (mutacja, reprodukcja, selekcja) na zbiorze kandydatów na rozwiązanie. Spodziewamy się, że z pokolenia na pokolenie w zbiorze będą pojawiać się coraz lepsze rozwiązania. Celem działania może być albo znalezienie **optymalnego** rozwiązania (o ile umiemy to stwierdzić) lub rozwiązania **wystarczająco dobrego** (np. krótszego niż 2000 km, kontynuując przykład o szukaniu drogi).

Zdefiniowanie operacji ewolucyjnych (mutacja, reprodukcja) zależy od konkretnego problemu, a przede wszystkim postaci rozwiązań. W naszym przykładzie można by zaproponować mutację jako zamianę losowego miasta na ścieżce na inne (sąsiednie), np.: Warszawa-**Genewa**-Amsterdam → Warszawa-**Zurych**-Amsterdam. Z kolei reprodukcję przez *crossing-over* możemy zdefiniować poprzez algorytm:

- 1) wylosować po jednym punkcie na każdej z dwóch ścieżek biorących udział w reprodukcji
- 2) rozerwać ścieżki w wylosowanych punktach
- 3) skleić ścieżki łącząc początek pierwszej z końcem drugiej i początek drugiej z końcem pierwszej.

Działanie opisanego algorytmu na przykładowych danych przedstawione zostało na rysunku 1.

### 2.1.2. Historia obliczeń ewolucyjnych

Historia (zob. [Eiben, Smith, 2003]) zastosowania podejścia ewolucyjnego do rozwiązywania problemów sięga jeszcze czasów przedkomputerowych. Już w 1948 roku Alan Tu-

<sup>5</sup>Źródło: <http://maps.google.com/>.

Rys. 1. Schemat przykładowej realizacji przebiegu operacji *crossing-over* dla dwóch ścieżek pomiędzy Warszawą a Amsterdamem



Źródło: Opracowanie własne.

ring zaproponował podejście „genetycznego lub ewolucyjnego poszukiwania”. Początki praktycznego stosowania tego podejścia przypadają jednak na lata 60. XX wieku, kiedy to powstały trzy sposoby realizacji tej koncepcji: **programowanie ewolucyjne**<sup>6</sup> (ang. *evolutionary programming*), **algorytmy genetyczne** (ang. *genetic algorithm*) oraz **strategie ewolucyjne** (ang. *evolution strategies*). W latach 90 uznano, iż wspomniane trzy koncepcje są tak naprawdę różnymi odmianami jednej, którą nazwano **obliczeniami ewolucyjnymi** (ang. *evolutionary computing*). Mniej więcej w tym samym czasie powstał kolejny nurt nazwany **programowaniem genetycznym** (ang. *genetic programming*).

Obecnie całą dziedzinę nazywa się **obliczeniami ewolucyjnymi** (ang. *evolutionary computing*), natomiast algorytmy w niej używane – **algorytmami ewolucyjnymi** (ang. *evolutionary algorithms*). Poszczególne, wymienione powyżej, algorytmy traktuje się jako poddziedziny będące po prostu różnymi wariantami algorytmu. Obliczenia ewolucyjne zalicza się do młodej dziedziny informatyki – sztucznej inteligencji.

Wymienione koncepcje zostaną po krótko scharakteryzowane w kolejnym podrozdziale. Szczególna uwaga poświęcona zostanie programowaniu genetycznemu, którego opis znajduje się w osobnym podrozdziale – 2.3.

### 2.1.3. Przegląd koncepcji obliczeń ewolucyjnych

Poniżej zaprezentowane zostaną różne realizacje idei obliczeń ewolucyjnych.

**Algorytmy genetyczne** są najbardziej ogólną grupą algorytmów ewolucyjnych, której podstawy stworzył Holland. Zawiera ona algorytmy umożliwiające rozwiązywanie proble-

<sup>6</sup>Polską terminologię zaczerpnięto z [Kwaśnicka, 2002].

mów optymalizacyjnych. Najczęściej wykorzystywane są do znajdowania ekstremów funkcji wielu zmiennych. W tzw. kanonicznej wersji osobniki reprezentowane są jako ciągi bitów (interpretowane np. jako wartości kolejnych zmiennych w zagadnieniu szukania ekstremum). Podejście to charakteryzuje możliwość bezpośredniej interpretacji osobników (odczytania fenotypu z genotypu – np. wartości zmiennych z ich reprezentacji binarnej).

**Programowanie ewolucyjne** jest podejściem zaproponowanym przez Lawrence'a Fogla i polegającym na symulowaniu ewolucji na automatach skończonych. Celem ich działania jest przewidzenie kolejnego symbolu na wejściu. Miarą jakości jest dokładność prognozy. Zmiany osobników następują jedynie w drodze mutacji (nie ma krzyżowania). Cechą charakterystyczną tego podejścia jest selekcja. Odbywa się ona poprzez turnieje: z całej populacji wybierane jest losowo  $n$  osobników, a z nich wybierany do następnego pokolenia jest jeden – najlepszy.

Z kolei istotą **strategii ewolucyjnych**, zapoczątkowanych przez Rechenberga we wczesnych latach 60, jest konkurowanie pomiędzy rodzicami a potomkami. W przeciwieństwie do innych podejść, gdzie rola rodziców ogranicza się do wyprodukowania potomków, tu i rodzice i ich dzieci konkurują ze sobą i wygrywają najlepsi spośród nich wszystkich. Dzięki temu dobre rozwiązania mogą utrzymywać się w populacji przez wiele pokoleń. Podejście to, podobnie jak programowanie ewolucyjne, najczęściej wykorzystywane jest do rozwiązywania problemów optymalizacyjnych.

Historycznie najmłodszą i jednocześnie – z punktu widzenia tej pracy – najciekawszą odmianą algorytmów ewolucyjnych jest **programowanie genetyczne**. Tutaj osobnikiem jest cały program komputerowy, funkcja, czy też model. Celem ewolucji jest odnalezienie najskuteczniejszego programu. Do najciekawszych zastosowań tej odmiany algorytmów należy ewoluowanie strategii gier (np. gry w szachy). W podejściu tym, osobniki reprezentowani są zazwyczaj przez struktury drzewiaste – można w ten sposób z łatwością przedstawić zarówno algorytm, jak i funkcję matematyczną. Ogromną zaletą takiej reprezentacji jest łatwość określenia operacji krzyżowania i mutacji. Programowanie genetyczne zostanie szerzej omówione w rozdziale 2.3.

#### **2.1.4. Przykłady zastosowań**

Zastosowania obliczeń ewolucyjnych mają szerokie spektrum – to m.in. sterowanie robotem, konstruowanie skrzydeł samolotów i rozwiązywanie równań całkowych. W tym podrozdziale zostanie dokonany przegląd zastosowań z perspektywy ich systemowej klasyfikacji (zob. [Eiben, Smith, 2003]).

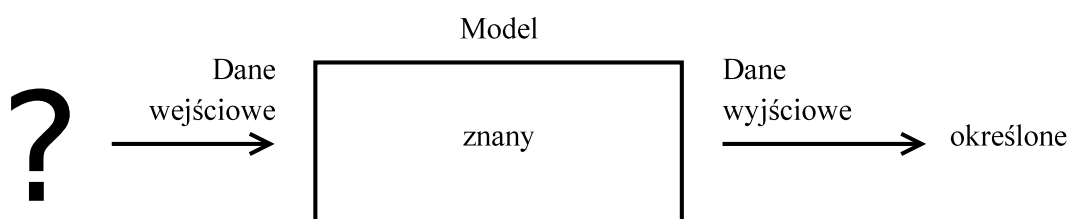
Należy zauważyć, iż z punktu widzenia analizy systemowej w każdym działającym

systemie można wyodrębnić trzy podstawowe komponenty: dane wejściowe, strukturę modelu oraz dane wyjściowe. Problemy, jakie stawia się algorytmom, to obliczenie jednego z nich, przy zadanych pozostałych.

## Optymalizacja

Pierwszą grupę problemów stanowią problemy natury optymalizacyjnej. W tej kategorii znany jest model oraz określone są pożądane dane wyjściowe. Zadanie stanowi obliczenie odpowiednich danych wejściowych, czego schemat zaprezentowany został na rysunku 2.

Rys. 2. Schemat problemów optymalizacyjnych



Źródło: [Eiben, Smith, 2003].

Przykładowym problemem w tej kategorii jest TSP (ang. *Travelling Salesperson Problem*<sup>7</sup>). Dla zadanego zbioru miast i odległości pomiędzy nimi należy obliczyć najkrótszą trasę przejazdu dla sprzedawcy, która pozwoli mu każde miasto odwiedzić dokładnie raz. Wejściem dla tego problemu jest ścieżka. Modelem jest obliczenie długości tej trasy (algorytm jest znany). Wyjściem systemu jest obliczona długość. Celem jest znalezienie takiego wejścia, żeby wyjście było minimalne.

Przykładem pomyślnego zastosowania algorytmu ewolucyjnego do problemu optymalizacyjnego jest układanie planu na uniwersytecie. Polega ono na takim przydziale sal oraz terminów odbywania się zajęć, aby dostosować się do czasu, jakim dysponują wykładowcy, nie powodować kolizji (lokalowych i personalnych) oraz dodatkowo zminimalizować nieciągłości czasowe w planie (tzw. okienka). Ewolucyjne rozwiązanie<sup>8</sup> tego zagadnienia zaprezentowane

<sup>7</sup>Jeszcze kilka lat temu problem nosił miano *Travelling Salesman Problem*. Jednak poprawność polityczna spowodowała, iż obecnie używa się nazwy wykorzystującej neutralne płciowo słowo *salesperson* – osoba sprzedająca, w miejsce męskoosobowego sprzedawcy (ang. *salesman*).

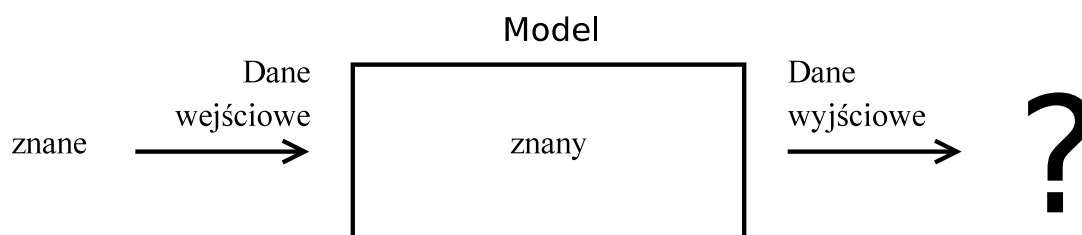
<sup>8</sup>Algorytm zastosowano do przygotowania planu sesji egzaminacyjnej na rok akademicki 1994/1995 dla 8000 studentów i 805 egzaminów. Wymaganiami względem planu było: żaden student nie może mieć dwóch egzaminów o tym samym czasie, na jeden termin egzaminu nie może przypaść więcej niż 1550 studentów (liczba siedzeń w sali) oraz jeśli studentowi wypadają dwa egzaminy jednego dnia, to musi mieć między nimi rozsądną długości odstęp.

zostało na Uniwersytecie Nottingham w Wielkiej Brytanii (zob. [Burke et al., 1995]).

## Symulacje

Problemy symulacyjne polegają na obliczeniu danych wyjściowych, przy zadanych wejściowych oraz znanym modelu nimi rządzącym. Schemat tego podejścia widoczny jest na rysunku 3.

Rys. 3. Schemat problemów symulacyjnych



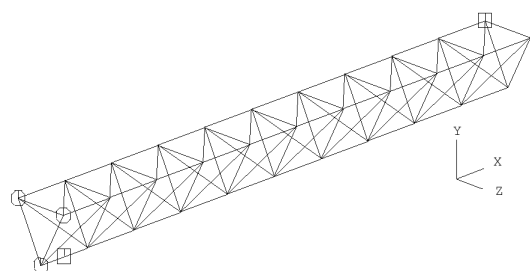
Źródło: [Eiben, Smith, 2003].

Przykładem może być symulowanie działania skomplikowanych wielorównaniowych układów elektrycznych. Prościej i taniej może być zaprojektować algorytm symulujący dany układ niż zbudowanie samego układu, by empirycznie sprawdzić jego właściwości. Podejście to jest szczególnie przydatne przy prototypowaniu (inżynierowie mogą tanio zweryfikować działanie projektu np. skrzydła samolotu).

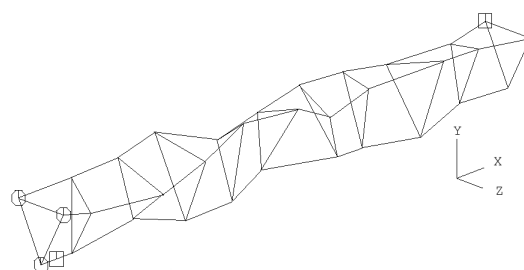
Podejście to sprawdziło się doskonale przy projektowaniu ramienia satelity (zob. [Keane, 1996]). Zadaniem było zaprojektowanie konstrukcji maksymalnie odpornej na wstrząsy tak, aby satelita mogła przetrwać zderzenie ze śmieciami kosmicznymi. Algorytm ewolucyjny znalazł projekt o 20000% lepszy niż tradycyjnie stosowany. Na rysunku 4 widać efekt pracy algorytmu.

Niezwykłą cechą rozwiązania znalezionej przez algorytm ewolucyjny jest jego nieintuicyjność. Umysł ludzki poszukuje rozwiązań na utartych ścieżkach, ciężko mu wybić się ze schematu. W związku z tym inżynierowie poszukujący optymalnej konstrukcji sprawdzali jedynie proste symetryczne konstrukcje. Algorytm ewolucyjny jest wolny od tych ograniczeń. Swobodnie sprawdza wszystkie poprawne rozwiązania, poszukując najlepszego. Jak pokazuje ten przykład, wykraczanie poza granice ludzkich schematów może się opłacać.

Rys. 4. Obraz efektów pracy algorytmu ewolucyjnego nad projektowaniem ramienia satelity na tle wyjściowego projektu



(a) Wyjściowy projekt ramienia satelity



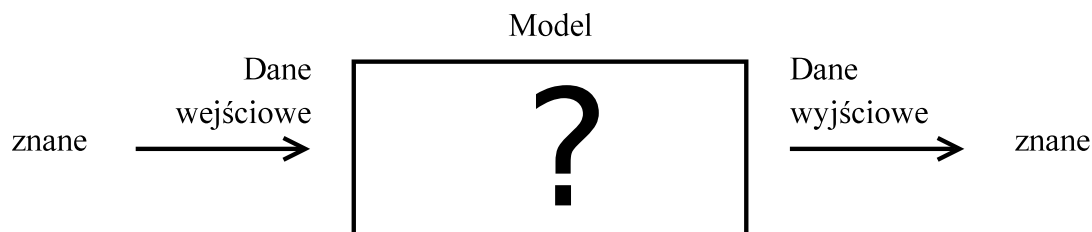
(b) Projekt znaleziony przez algorytm ewolucyjny

Źródło: [Keane, 1996].

## Modelowanie

W tym podejściu poszukiwany jest model wyjaśniający przejście od zadanych danych wejściowych do znanych wyjściowych. Schemat tego podejścia widoczny jest na rysunku 5.

Rys. 5. Schemat problemów modelowania



Źródło: [Eiben, Smith, 2003].

Ciekawym przykładem problemu modelowania jest zadanie znalezienia algorytmu cofania ciężarówka z naczepą pod rampę do załadunku (zob. [Koza, 1994]). Świat prowadzenia ciężarówka jest 4-wymiarowy: 2 wymiary to lokalizacja naczepy, kolejny to kąt pomiędzy pojazdem a rampą, ostatni – kąt skrętu samej naczepy. Wartości początkowe tych czterech wymiarów są jednocześnie danymi wejściowymi systemu. Poszukiwanym modelem jest algorytm przekształcający te dane w najkrótszą trajektorię jazdy ciężarówka prowadzącą do określonego celu. Efektem pracy programowania genetycznego, które to autorzy zastosowali w swoim podejściu, jest skuteczny algorytm prowadzenia ciężarówka. Warto dodać, iż problem ten nie ma rozwiązania analitycznego.

Podobnie konstruować można modele postępowania robotów, wykonujących różne zadania (przenoszenie pudełek, omijanie przeszkód) i wiele innych podobnych zadań. Inną grupę problemów stanowi natomiast poszukiwanie funkcji najlepiej dopasowanej do zadanych danych. Wejściem systemu są wówczas zmienne niezależne, wyjściem – zmienne zależne, a (poszukiwanym) modelem – funkcja. Jest to ewolucyjna alternatywa dla klasycznej ekonometrii. Zagadnienie to, kluczowe dla tej pracy, zostanie szerzej omówione w kolejnym podrozdziale.

## 2.2. Algorytmy ewolucyjne a modelowanie ekonomiczne

Jak zostało zauważone w poprzednim rozdziale, algorytmy ewolucyjne mogą znaleźć zastosowanie w modelowaniu ekonomicznym jako narzędzie do odnajdywania prawidłowości rządzących danymi. Mogą posłużyć do skonstruowania funkcji przybliżającej dane, mogą dostarczyć modelu. Istotną zaletą ewolucyjnego podejścia do modelowania jest znaczne ograniczenie wiedzy *a priori*. W ekonometrii to analityk decyduje o formie funkcyjnej modelu, sugerując się testami statystycznymi, wiedzą ekonomiczną na temat modelowanego zjawiska oraz – przede wszystkim – własnym doświadczeniem. W podejściu ewolucyjnym to właśnie ewolucja steruje wyborem najbardziej odpowiedniej funkcji. Proces modelowania jest więc w pełni sterowany danymi, bez wiedzy z zewnątrz. Jak uczy przykład konstrukcji ramienia satelity (zob. rozdz. 2.1.4.), wiedza ta może oddalać od najlepszych rozwiązań.

Nie sposób omówić ewolucyjnego podejścia do modelowania ekonomicznego bez przyjrzenia się innej dziedzinie badań w ramach ekonomii, na którą ewolucja miała ogromny wpływ, a mianowicie na **ekonomię ewolucyjną**.

Początki ekonomii ewolucyjnej odnajduje się w pracach Herberta Spencera (zob. [Kwaśnicki, 1996]), który rozwijał swoją teorię równoległe do Darwina. Spencer jednak definiował procesy ewolucyjne odmiennie. Dla niego różnorodność była celem ewolucji, a nie jej narzędziem. Utożsamiał on ewolucję z ciągłym postępowaniem, wzrostem efektywności w kierunku pewnego idealnego stanu, z przechodzeniem od homogeniczności do heterogeniczności. Ważnym założeniem Spencera był przy tym fakt, że cel (harmonia) jest osiągalny.

Kolejnymi myślicielami, zajmującymi się ekonomią ewolucyjną byli Carl Menger, Alfred Marshall i Thorstein Veblen. Żaden z nich jednak nie przedstawił konkretnego programu ekonomii ewolucyjnej. Dopiero Joseph Schumpeter zaproponował pierwsze dojrzałe sformułowanie zadania analizy ekonomicznej w duchu ewolucyjnym. Upatrywał on przyczyn zmian gospodarczych w procesach wewnętrznych, wbrew dominującemu przekonaniu o zewnętrznym ich pochodzeniu. Co więcej, stwierdził on, że każdy system gospodarczy podlega ewolucji –



rozwija się niespokojnie, gwałtownie, w sposób przypominający serię eksplozji.

Współczesny głos ekonomii ewolucyjnej należy do Richarda Nelsona (Columbia University) i Sidneya Wintera (University of Pennsylvania). Ich badania koncentrują się na innowacjach i rutynach w przedsiębiorstwach. Firmy konkurują ze sobą w poszukiwaniu innowacji, operując zestawem rutyn, czyli zwyczajowymi wzorcami zachowań, obejmującymi m.in. technologię produkcji, czy wszelkiego rodzaju wewnętrzne procedury. Zmiana rutyny firmy może być jej odpowiedzią na zmianę warunków, w jakich funkcjonuje i jest to bezpośrednia analogia do Darwinowskiego procesu ewolucji.

Współczesny rozwój ekonomii ewolucyjnej to w dużym stopniu konstruktywna realizacja krytyki ekonomii neoklasycznej (zob. [Kwaśnicki, 2001]). Główny zarzut to po prostu nieadekwatność opisu rzeczywistości ekonomii neoklasycznej, która – inspirowana mechaniką – obserwuje statyczny wycinek rzeczywistości i stara się zamrozić jak najwięcej parametrów. Tymczasem ewolucja to dążenie do optymalności, a nie optymalność. Ewolucja to wykorzystanie lokalnej wiedzy, a nie dążenie do pozyskania pełnej wiedzy. Wreszcie ewolucja to rozpatrywanie całej populacji wraz ze wszystkimi interakcjami pomiędzy jej członkami, a nie opisywanie reprezentanta. Według Kwaśnickiego (zob. [Kwaśnicki, 1996, 2001]) wciąż poszerza się grono zwolenników ewolucyjnego podejścia do ekonomii.

Jak widać, ewolucyjne modelowanie ekonomiczne jest niezależne od ekonomii ewolucyjnej, jest raczej rozwinięciem modelowania, dostarczeniem alternatyw dla klasycznych modeli regresyjnych. Badania w jego ramach prowadzone są głównie przez informatyków (z Johnem Kożą jako liderem ewolucyjnego modelowania ekonomicznego), a nie przez ekonomistów. Przyczyną tego stanu rzeczy może być fakt, że programowanie genetyczne wymaga dobrych umiejętności programistycznych. Brak jest niestety gotowych narzędzi do przeprowadzania modelowania ekonomicznego przy użyciu programowania genetycznego.

### **2.3. Programowanie genetyczne**

Jak zostało to zarysowane w rozdz. 2.1.3. programowanie genetyczne jest poddawaniem procesowi ewolucji całych programów, strategii, funkcji, a osobnicy reprezentowani są zazwyczaj przez struktury drzewiaste. W modelowaniu ekonomicznym poszukuje się najwłaściwszej funkcji.

W tym podrozdziale opisany zostanie zatem szczegółowo wszystko to, co jest potrzebne, żeby korzystać z programowania genetycznego, a więc reprezentacje funkcji w postaci drzewek funkcji elementarnych oraz operacje genetycznych na nich.

### 2.3.1. Reprezentacja funkcji jako drzewek

Ideą „udrzewienia” funkcji jest reprezentowanie wywołania funkcji jako korzenia drzewa, a jej poszczególnych parametrów – jako podwęzłów. Koncepcja ta zostanie omówiona na następującym przykładzie:

$$\sin(\log(1/y) * ((2 * x) + \cos(3)))$$

W powyższym przykładzie występują:

- stałe (1, 2, 3)
- zmienne ( $x$ ,  $y$ )
- funkcje (sin, cos, log, \* – mnożenie, + – dodawanie, / – dzielenie).

Dla przejrzystości rozumowania wszystkie obiekty wchodzące w skład funkcji będą traktowane jako funkcje elementarne. Stałe i zmienne będą szczególnym ich przypadkiem – funkcjami 0-argumentowymi. Funkcje te mają wartość stałej (zmiennej), którą reprezentują. Pozostałe funkcje będą grupowane ze względu na ich arność. Dla uproszczenia operacje dodawania, odejmowania, mnożenia i dzielenia będą traktowane jako funkcje 2-argumentowe. Warto zauważyć, że wieloskładnikowe (wieloczynnikowe) operacje dodawania (mnożenia) można zawsze reprezentować jako złożenia 2-argumentowych wersji tych operacji, co gwarantuje nam prawo łączności<sup>9</sup> tak mnożenia jak i dodawania.

Po uwzględnieniu powyższych założeń lista funkcji elementarnych w powyższym przykładzie wygląda następująco:

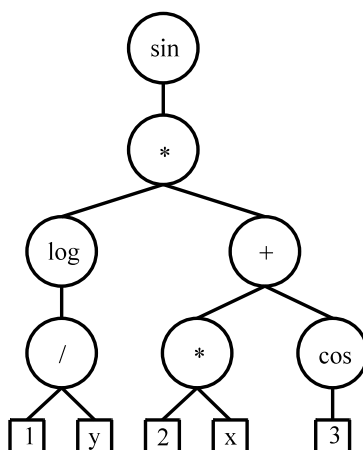
- funkcje 0-argumentowe (1, 2, 3,  $x$ ,  $y$ )
- funkcje 1-argumentowe (sin, cos, log)
- funkcje 2-argumentowe (\*, +, /).

Konstrukcję drzewa rozpoczyna się od najbardziej zewnętrznej funkcji. W zaproponowanym przykładzie jest nią sin. Ma ona jeden argument ( $\log(1/y) * ((2 * x) + \cos(3))$ ) – 2-argumentowe działanie mnożenia, którego czynnikami są  $\log(1/y)$  oraz  $(2 * x) + \cos(3)$ . Argumenty te rozkłada się analogicznie. Efektem operacji jest drzewko widoczne na rysunku 6.

Należy zauważyć, iż liśćmi w drzewie funkcji mogą być tylko funkcje 0-argumentowe. Z drugiej zaś strony funkcje te nie mogą wystąpić w roli węzłów.

<sup>9</sup>Prawo łączności dla działania  $\Delta$  mówi, iż:  $a\Delta b\Delta c = (a\Delta b)\Delta c = a\Delta(b\Delta c)$ .

Rys. 6. Schemat drzewiastej reprezentacji funkcji  $\sin(\log(1/y) * ((2 * x) + \cos(3)))$



Źródło: Opracowanie własne.

### 2.3.2. Generowanie nowych osobników

#### Generowanie losowych funkcji

Tworzenie nowych osobników populacji odbywa się poprzez losowe generowanie drzewek funkcji. W tym celu losowana jest funkcja będąca korzeniem drzewa. W losowaniu tym prawdopodobieństwo wyboru każdej funkcji nie musi być jednakowe. Można np. dyskryminować przy losowaniu korzenia funkcje 0-argumentowe, aby zmniejszyć w populacji udział stałych i zmiennych, a zwiększyć – funkcji. Po wybraniu funkcji należy wygenerować dla niej odpowiednią liczbę argumentów. Każdy z nich będzie poddrzewem i jest generowany w sposób analogiczny do całego drzewa.

Przy schodzeniu włąb drzewa można zmieniać prawdopodobieństwo wyboru funkcji 0-argumentowej (liścia drzewa), np. stopniowo je zwiększając aż do 1 przy założonym maksymalnej głębokości drzewa. W ten sposób można zapewnić preferencję dla większych (głębszych) drzewek.

#### Generowanie stałych

Osobnego omówienia wymaga zagadnienie stałych w generowanych funkcjach. Za każdym razem, gdy wyżej opisany algorytm decyduje o umieszczeniu w drzewku stałej (tj. wybiera 0-argumentową funkcję niebędącą zmienną), generowana jest pseudolosowa liczba z określonego zakresu. Liczba ta jest następnie umieszczana w drzewie funkcji.

Należy zauważyć, iż działając na stałą inną funkcją (tj. umieszczając stałą jako dziecko

węzła funkcji wieloargumentowej), mogą powstawać stałe o innych wartościach, np.

$$1 \rightarrow \sin(1) \simeq 0,841$$

Dodatkowo, dzięki operacjom genetycznym w trakcie ewolucji, wartości te mogą się zmieniać. Oba te mechanizmy powodują, że wylosowanie stałych w momencie generowania osobnika nie ogranicza w niczym różnorodności populacji.

### 2.3.3. Cenzurowanie funkcji

Stosując obliczenia należy dbać o przestrzeganie dziedzin funkcji elementarnych. Na przykład operacja dzielenia jest zdefiniowana na zbiorze  $\mathbf{R} \times \mathbf{R} \setminus \{0\}$ . Należy zatem zadbać, żeby drugim argumentem tej funkcji nie było 0, dla tej wartości wynik funkcji jest bowiem nieokreślony. Podobną trudność mamy z funkcją logarytmiczną, której dziedziną jest przedział  $(0, \infty)$ .

Problem można łatwo rozwiązać stosując w miejsce dzielenia i logarytmowania „ocenzurowane” wersje tych funkcji. Zdefiniować można (zob. [Koza, 1991]) następujące funkcje:

$$\begin{aligned} \% (x, y) &= \begin{cases} 0 & \text{jeżeli } y = 0 \\ x/y & \text{wpp.} \end{cases} \\ \text{RLOG}(x) &= \begin{cases} 0 & \text{jeżeli } x = 0 \\ \log |x| & \text{wpp.} \end{cases} \end{aligned}$$

Dziedziną tak zdefiniowanej funkcji  $\%$  jest  $\mathbf{R} \times \mathbf{R}$ , a funkcji RLOG –  $\mathbf{R}$ .

### 2.3.4. Operacje genetyczne

#### Migracja

Najprostszą operacją, jaka może zostać przeprowadzona na populacji jest migracja. Jej ideą jest symulowanie dołączenia do populacji osobnika z innego środowiska. W praktyce realizuje się to generując nowego osobnika i dołączając go do populacji.

## Mutacja

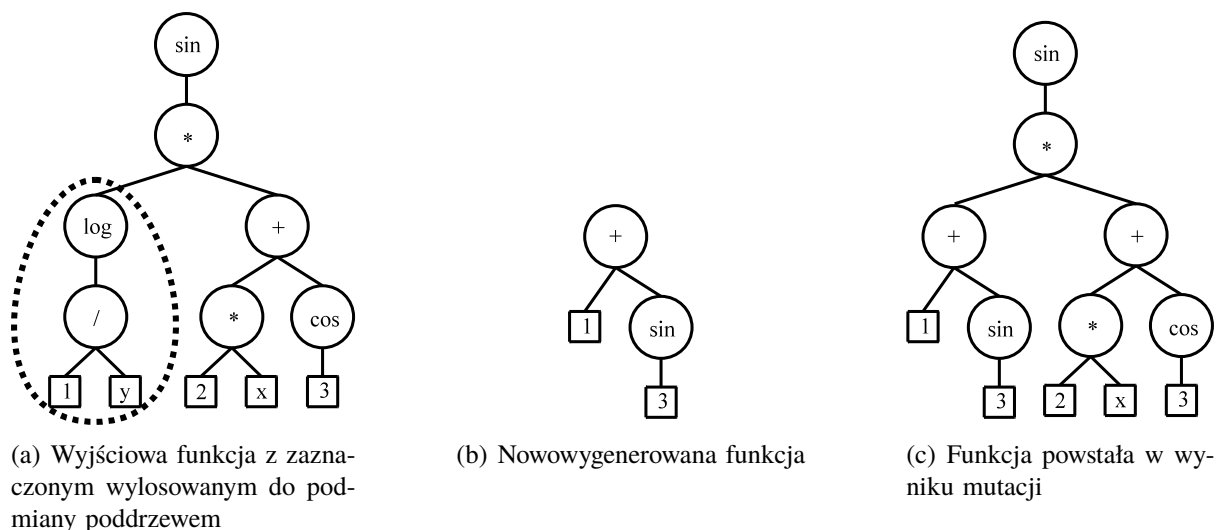
Mutacja polega na losowej zmianie w strukturze drzewa. Zrealizować ją można jako zamianę losowo wybranego węzła drzewa nowogenerowanym poddrzewem. Poniżej opisany jest przykład takiej operacji.

Przeanalizowana zostanie operacja mutacji dla przykładu z poprzedniego podrozdziału ( $\sin(\log(1/y) * ((2 * x) + \cos(3)))$ ). Mutacja wybiera losowo węzeł do zmiany (np. ten z funkcją logarytmiczną). Niezależnie od istniejącego drzewa, generowana jest nowa funkcja, zgodnie z opisem z poprzedniego podrozdziału. Niech funkcją tą będzie  $1 + \sin(3)$ . Wyjściowa funkcja, po wstawieniu w miejsce logarytmu nowej funkcji wyglądać będzie następująco:

$$\sin((1 + \sin(3)) * ((2 * x) + \cos(3)))$$

Wizualizacja opisanego procesu znajduje się na rysunku 7.

Rys. 7. Schematyczna prezentacja przykładowej realizacji procesu mutacji



Źródło: Opracowanie własne.

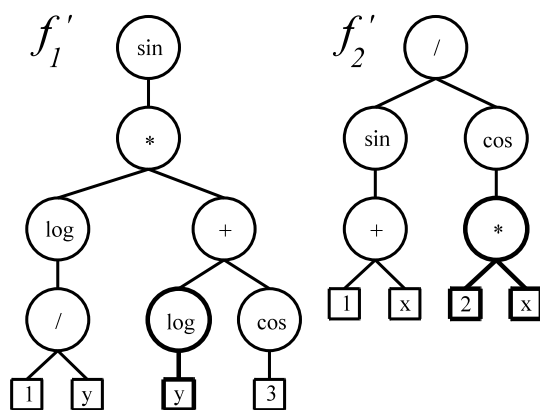
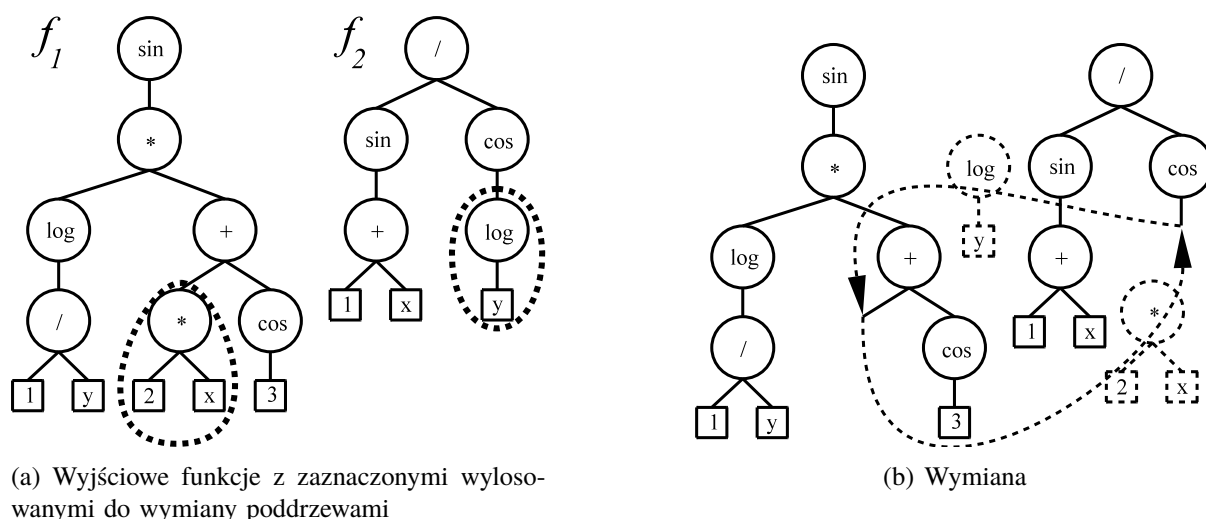
Jeżeli osobnik powstały w wyniku mutacji jest większy (w sensie wysokości drzewa) niż przyjęte maksimum, operacja mutacji jest przerywana i jej rezultatem jest wyjściowy osobnik.

## Krzyżowanie

W procesie krzyżowania dwa osobniki wymieniają się materiałem genetycznym tworząc dwa nowe osobniki. W języku drzew funkcji oznaczać to będzie, że mając dwa drzewa funkcji, w każdym wylosowane zostanie poddrzewo, a następnie poddrzewa te zostaną wymienione pomiędzy funkcjami.

Przykładowa realizacja krzyżowania zostanie omówiona na dwóch funkcjach:  $f_1 = \sin(\log(1/y) * ((2*x) + \cos(3)))$  oraz  $f_2 = (\sin(1+x)) / \cos(\log(y))$ . Losowy wybór poddrzew do skrzyżowania mógł paść na poddrzewo  $2*x$  w  $f_1$  oraz  $\log(y)$  w  $f_2$ . Po wzajemnej wymianie poddrzewami, wyjściowe funkcje przyjmą postać  $f'_1 = \sin(\log(1/y) * ((\log(y)) + \cos(3)))$  oraz  $f'_2 = (\sin(1+x)) / \cos(2*x)$ . Wizualizacja tego procesu znajduje się na rysunku 8.

Rys. 8. Schematyczna prezentacja przykładowej realizacji procesu krzyżowania



(c) Funkcje powstałe w wyniku krzyżowania

Źródło: Opracowanie własne.

Funkcje  $f'_1$  oraz  $f'_2$  powstałe w ten sposób to potomkowie funkcji  $f_1$  oraz  $f_2$ . Podob-

nie jak w przypadku mutacji, jeżeli w wyniku krzyżowania powstaną osobniki większe niż przyjęte maksimum, operacja ta jest anulowana.

### 2.3.5. Przeżycie najbardziej przystosowanego

Do realizacji ewolucyjnego mechanizmu selekcji oraz idei „przeżycia najbardziej przystosowanego” potrzebna jest miara tego przystosowania. Dla modeli ekonomicznych do dyspozycji mamy m.in. takie miary jak błąd średniokwadratowy (SSE), czy średni bezwzględny błąd procentowy (MAPE).

Przeprowadzone zostało badanie (zob. [Makridakis, 1993]), którego wnioski sugerują posługiwanie się zmodyfikowanym MAPE do oceny dopasowania modelu. Argumentuje się, że używane kryterium powinno być proste, niewrażliwe na outlierów, zawsze obliczalne (tj. bez dzielenia przez 0), stabilne (np. bez dzielenia przez bardzo małe i bardzo duże liczby) oraz wyrażone w wartościach względnych. Makridakis w swojej pracy ocenia różne kryteria w dwóch podejściach: teoretycznym (podsumowując zalety i wady w myśl wymienionych wyżej cech) oraz praktycznym (oceniając jakość prognoz z modeli dopasowanych różnymi kryteriami). Ostatecznie przekonuje, że miarą najlepiej spełniającą postawione kryteria jest MAPE.

W oparciu o wnioski Makridakisa w pracy przyjęto następującą miarę dopasowania:

$$r(i) = \sum_{j=1}^M \left| \frac{f_i(x_j^{(1)}, x_j^{(2)}, \dots) - y_j}{f_i(x_j^{(1)}, x_j^{(2)}, \dots)} \right|$$

gdzie:

- $r(i)$  – miara dopasowania  $i$ -tego osobnika
- $f_i$  – funkcja, jaką reprezentuje ten osobnik
- $x^{(k)}$  –  $k$ -ta zmienna
- $x_j^{(k)}$  –  $j$ -ta obserwacja  $k$ -tej zmiennej
- $y$  – zmienna objaśniana
- $M$  – liczba obserwacji.

Należy zauważyć, że  $r(i) \in [0, \infty)$  i im niższa jest wartość tej miary, tym lepiej „przystosowany” dany osobnik.

Dalej, podążając za [Koza, 1992], miarę dopasowania poddaje się kalibracji, definiując:

$$a(i) = \frac{1}{1 + r(i)}$$

Skalibrowana miara zawiera się w przedziale  $[0, 1]$  i ma tę właściwość, że lepsi osobnicy osiągają większą wartość. Ostatnim krokiem jest normalizacja (przy oznaczeniu przez  $N$  wielkości populacji):

$$n(i) = \frac{a(i)}{\sum_{k=1}^N a(k)}$$

W ten sposób  $n(i) \in [0, 1]$  oraz dodatkowo  $\sum_{i=1}^N n(i) = 1$ . Dzięki wszystkim powyższym zabiegom uzyskana została zatem miara prawdopodobieństwa na zbiorze osobników, która wyższe prawdopodobieństwo przypisuje osobnikom lepiej przystosowanym. Miara ta będzie używana do wybierania osobników do reprodukcji i przetrwania.

### 2.3.6. Parametry programowania genetycznego

Aby móc zrealizować algorytm genetyczny, należy ustalić parametry wymienione w tabeli 3.

Tabela 3. Zestawienie parametrów programowania genetycznego

Parametry	Opisy
Wielkość populacji	Liczba osobników w populacji
Liczba pokoleń	Maksymalna liczba iteracji (pokoleń), jako może zostać przeprowadzona
Funkcje elementarne	Zestaw stałych, zmiennych i wieloargumentowych funkcji elementarnych
Inicjalna głębokość drzewa	Dopuszczalna głębokość drzewa, które może zostać wygenerowane (przy generowaniu populacji i migracji)
Maksymalna głębokość drzewa	Dopuszczalna głębokość drzewa, które może istnieć (ograniczenie istotne przy mutacji i krzyżowaniu)
Odsetek krzyżowania	Odsetek populacji, jaki powstaje w wyniku krzyżowania
Odsetek migracji	Odsetek populacji, jaki powstaje w wyniku migracji
Odsetek mutacji	Odsetek populacji, jaki powstaje w wyniku mutacji
Odsetek reprodukcji	Odsetek populacji, jaki powstaje w wyniku reprodukcji

Źródło: Opracowanie własne.

Zadanie znalezienia najlepszego dla danego problemu zestawu parametrów jest niezwykle trudne. Wymaga ono od badacza doświadczenia oraz pracy. Najprostszym sposobem jest zastosowanie metody prób i błędów – uruchamianie algorytmu z różnymi parametrami i ocenianie ich na podstawie jakości modelu znalezionego przez algorytm. Ponieważ każdy przebieg algorytmu ewolucyjnego jest niedeterministyczny, dla danego zestawu parametrów należy przeprowadzić wiele prób a wyniki uśrednić. Z tego względu bardzo dobrym pomysłem wydaje się opieranie (o ile to tylko możliwe) nowych badań na parametrach wypracowanych



przez innych badaczy przy pracy z problemami podobnej natury.

### **2.3.7. Schemat programowania genetycznego**

Poniżej zaprezentowany został schemat postępowania w ewolucyjnym modelowaniu ekonomicznym, będący podsumowaniem operacji zaprezentowanych w poprzednich sekcjach:

- 1) przygotować populację początkową, generując  $M$  losowych osobników
- 2) wyliczyć miary dopasowania dla populacji –  $r(i)$
- 3) wybrać najlepiej dopasowanego osobnika oraz – jeżeli jest on lepszy niż najlepszy dotychczas znaleziony – zapamiętać go
- 4) jeżeli spełnione zostały warunki końca (osiągnięcie celu wyrażonego miarą dopasowania lub przekroczenie maksymalnej liczby pokoleń), przerwać, podając najlepszego napotkanego osobnika jako odpowiedź
- 5) wyliczyć rozkład prawdopodobieństwa –  $n(i)$
- 6) ustaloną liczbę razy wykonać krzyżowanie, składające się z następujących kroków:
  - wybrać dwóch osobników zgodnie z określoną miarą prawdopodobieństwa
  - wykonać krzyżowanie
  - do populacji potomków dołączyć powstałych w ten sposób 2 osobników
- 7) ustaloną liczbę razy wykonać reprodukcję (wybrać jednego osobnika zgodnie z określoną miarą prawdopodobieństwa i dołączyć go do populacji potomków)
- 8) ustaloną liczbę razy wykonać migrację (wygenerować nowego osobnika i dołączyć go do populacji potomków)
- 9) ustaloną liczbę razy wykonać mutację (wybrać losowo osobnika, przeprowadzić na nim mutację i nowopowstałego osobnika dołączyć do populacji potomków)
- 10) powtórzyć algorytm od punktu 2 dla populacji potomków.

### **2.4. Zastosowanie programowania genetycznego do modelowania ekonomicznego**

Mimo iż samo programowanie genetyczne jest stosunkowo młodą dziedziną, a jego zastosowanie w modelowaniu ekonomicznym ma bardzo krótką historię, opublikowanych zostało już wiele ciekawych i obiecujących badań. W tym rozdziale omówione zostanie kilka najciekawszych prac.

### 2.4.1. Równowaga na rynku pieniądza

Interesujący przykład użycia programowania genetycznego do modelowania ekonomicznego podaje John Koza (zob. [Koza, 1991]). Zajmuje się on empiryczną weryfikacją równania z ilościowej teorii pieniądza, opisującego równowagę na rynku pieniądza:

$$P = \frac{MV}{Q}$$

gdzie:

- $P$  – poziom cen
- $V$  – prędkość obiegu pieniądza
- $M$  – ilość pieniądza w obiegu
- $Q$  – wolumen transakcji kupna-sprzedaży.

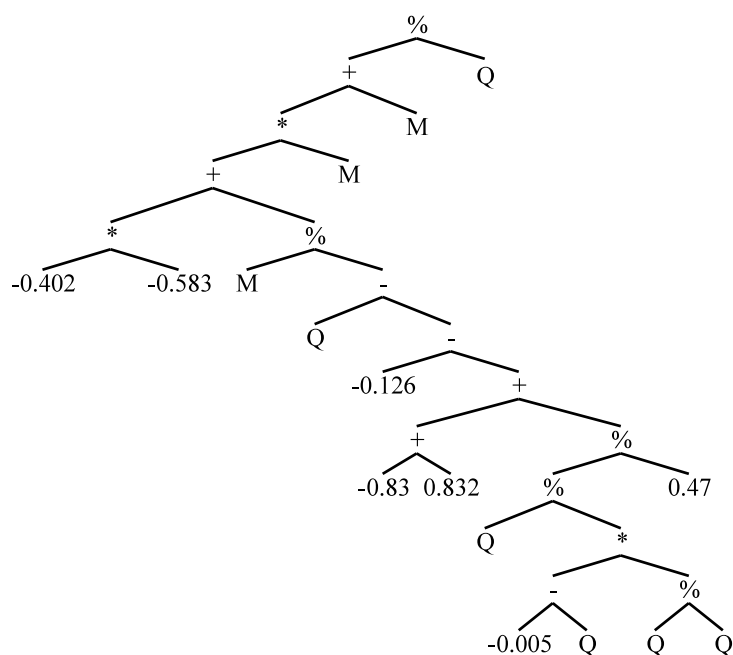
Celem badania Kozy było wykrycie zależności pomiędzy wspomnianymi czterema wartościami. W tym celu zebrał następujące kwartalne dane za lata 1959-1988 dla Stanów Zjednoczonych: PKB ( $Q$ ), deflator PKB ( $P$ ) oraz zasoby pieniądza w gospodarce ( $M2$ ). Dodatkowo przyjął na podstawie badań innych autorów długookresową prędkość obrotu pieniądza w gospodarce ( $V=1,6527$ ). Aby nie sugerować się teoretycznym równaniem, którego weryfikacja była celem, Koza dodał zmienną niezwiązaną z problemem ( $r$  – stopę procentową 3-miesięcznych bonów skarbowych). Zbiór zmiennych miał postać:  $T = \{Q, M, r\}$ . Dodatkową funkcją 0-argumentową był generator stałych. Za zmienną objaśnianą przyjęto deflator PKB ( $P$ ). Również zbiór funkcji elementarnych nie ograniczał się do tych użytych w równaniu równowagi na rynku pieniądza. Wyglądał on następująco:  $F = \{+, -, *, \%, \exp, \text{RLOG}\}$ .

Warto odnotować, że takie podejście do problemu minimalizuje wiedzę *a priori* wkładaną do modelu, co jest główną wytyczną filozofii programowania genetycznego.

Modelowanie wykonano w oparciu o populację 500 osobników, ewolucją sterowała miara błędu średniokwadratowego. Zastosowano metodę *in-sample* na 2/3 próbki. Już po zaledwie 15 pokoleniach udało się uzyskać osobnika (funkcję), którego średniokwadratowy błąd dopasowania wyniósł 0,009272 a  $R^2 = 0,99$ . Znalezione najlepsze rozwiązanie zaprezentowane jest na rysunku 9.

Znalezione drzewko reprezentuje następującą funkcję (dla uproszczenia zapisu funk-

Rys. 9. Obraz najlepszego rozwiązania ewolucyjnego modelowania równowagi na rynku pieniądza



Źródło: Opracowanie własne na podstawie [Koza, 1991].

cja % została zastąpiona przez kreskę ułamkową):

$$\bar{P} = \frac{\left( (-0,402) * (-0,583) + \frac{M}{Q - \left( -0,126 - \left( (-0,83 + 0,832) + \frac{\frac{Q}{(0,005 - Q) * \frac{Q}{Q}}}{0,47} \right) \right)} \right) * M + M}{Q}$$

Ostatnim, najprostszym i zarazem najbardziej spektakularnym krokiem pracy Kozy, było ręczne uproszczenie funkcji, za pomocą zwykłych operacji arytmetycznych. Okazuje się, że:

$$\bar{P} = \frac{M * 1,634}{Q}$$

Należy zauważyć, iż jest to realizacja równania  $P = \frac{MV}{Q}$  dla  $V = 1,634$ . Należy wspomnieć, że wyliczona przez badaczy wartość prędkości obrotu pieniądza wynosiła 1,6527. To, co otrzymano zatem z algorytmu ewolucyjnego, to potwierdzenie równania równowagi na rynku pieniądza z przybliżoną wartością prędkości obrotu pieniądza.

Otrzymany model dobrze zachowuje się przy prognozowaniu, o czym przekonują wyniki zebrane w tabeli 4.

Tabela 4. Zestawienie wartości miar dopasowania w modelu równowagi na rynku pieniądza wg badań Kozy

Zakres danych	Miary dopasowania	
	R <sup>2</sup>	SSE
Okres treningowy (1-80)	0,997949	0,009272
Okres kontrolny (81-120)	0,990614	0,066116
Cały okres (1-120)	0,993480	0,075388

Źródło: Opracowanie własne na podstawie [Kozy, 1991].

Warto dodać, iż model  $P = \frac{MV}{Q}$  z  $V = 1,6527$  (czyli wartością podaną przez badaczy) osiąga błąd średniokwadratowy na poziomie 0,077193 oraz  $R^2 = 0,993320$ . Rozpatrując cały okres można zatem stwierdzić, że w „naturalny” ewolucyjny sposób otrzymaliśmy równie dobry model co teoretyczny. Dodatkowo, dzięki podziałowi wszystkich danych na okres treningowy i kontrolny dowiedziona, że zbudowany model nie wpadł w pułapkę „wyuczenia się” danych, tj. internalizacji wszystkich przypadków z danych i tym samym świetnego do nich dopasowania, ale jednocześnie osiągnięcia fatalnych rezultatów na nowych danych.

#### 2.4.2. Modele wielorównaniowe

Programowanie genetyczne nie ogranicza się jedynie to pojedynczych równań. Można z powodzeniem stosować jego metody do modeli wielorównaniowych. Interesujący przykład takiego badania prezentują naukowcy z Uniwersytetu Erazma w Rotterdamie (zob. [Duyvesteyn, Kaymak, 2005]). Modelowaniu poddali oni następujący układ:

$$\begin{aligned}
 GDP_d &= GDP - T \\
 I &= f_I(GDP, r) \\
 C &= f_C(GDP_d) \\
 GDP &= C + I,
 \end{aligned}$$

gdzie:

$GDP_d$  – PKB do dyspozycji

$T$  – podatki

$I$  – inwestycje

$C$  – konsumpcja

$r$  – stopa procentowa.

Modelowanie polegało na znalezieniu funkcji  $f_I$  oraz  $f_C$ . Wyniki działania algorytmu

ewolucyjnego dla tak postawionego problemu były następujące:

$$f_I(GDP, r) = 0,2433 * GDP + 3,09162009 * \sqrt{r} * e^{\sqrt{r}} * \sqrt[4]{GDP} * (r^2 + GDP)$$

$$f_C(GDP) = 0,0885 + GDP_d + \frac{1}{GDP_d}$$

Dla porównania autorzy skonstruowali również model liniowy klasycznymi metodami z następującymi wynikami:

$$f_I(GDP, r) = 14,92 + 0,20 * GDP - 57,69 * r$$

$$f_C(GDP) = -5,14 + 1,06 * GDP_d$$

Jak widać, zależności znalezione przez algorytm ewolucyjny są skomplikowane i nieinterpretowalne ekonomicznie, co jest niewątpliwą wadą podejścia ewolucyjnego. Również wyniki w przypadku tego badania nie są imponujące – jakość dopasowania obydwu modeli (liniowego i ewolucyjnego) jest porównywalna.

Należy jednak zauważyć, iż w tym badaniu autorzy nie próbowali znaleźć zestawu parametrów (zob. tabela 3) optymalnego dla tego problemu. Jest to niezwykle trudnym zadaniem, wymagającym od badacza dużego doświadczenia oraz ogromu pracy.

### 2.4.3. Inne zastosowania programowania genetycznego

Siłą programowania genetycznego jest fakt, że nie ogranicza się ono jedynie do ewoluowania funkcji liczb rzeczywistych. Wszystkie opisane w tym rozdziale operacje, jak i cała ewolucja, będą działać dla dowolnego prawidłowego zestawu funkcji elementarnych. Przykładowe inne zastosowania programowania genetycznego to:

- modelowanie bramek logicznych (z zestawem funkcji logicznych (OR, AND, NOT) i sygnałami na wejściu jako zmiennymi)
- planowanie, np. problem załadunku, czyli skonstruowania planu przejścia od zestawu określonych elementów w określonej kolejności do docelowego stanu tych elementów w innej określonej kolejności. Funkcjami elementarnymi w tym zadaniu są operacje sprawdzania jaki element znajduje się w jakim miejscu, jak również rozkazy przeniesienia określonych elementów. Efektem działania algorytmu jest program wykonujący postawione zadanie
- rozwiązywanie równań różniczkowych (np.  $\frac{\partial y}{\partial x} + y \cos x = 0$ ), całkowych (np.  $y(t) - 1 + 2 \int_{r=0}^{r=t} \cos(t-r)y(r)dr = 0$ ), jak również równań funkcyjnych (np.  $f(2x) - 1 + 2 \sin^2 x = 0$ )

- rozwiązywanie zwykłych równań, gdzie funkcjami elementarnymi są jedynie stałe, a budowanie drzewek służy ewoluowaniu różnych liczb za pomocą działań elementarnych na liczbach losowych.

## ROZDZIAŁ III

### Implementacja programowania genetycznego jako narzędzia modelowania ekonomicznego

Głównym celem niniejszej pracy jest zweryfikowanie efektywności informacyjnej warszawskiej giełdy. Narzędziem do tego będzie modelowanie metodami programowania genetycznego. Z racji jednak na brak gotowych programów do takiego modelowania, pracę należy zacząć od opracowania własnej realizacji koncepcji programowania genetycznego. Rozdział ten poświęcony został omówieniu tej implementacji.

Najważniejszą decyzją, jaką należy podjąć przy projektowaniu programu, jest wybór języka programowania oraz ewentualnie środowiska docelowego. Dla informatyka naturalnym rozwiązaniem będzie stworzenie zupełnie nowego programu w języku programowania takim jak Java, C, czy Pascal. Alternatywnym rozwiązaniem jest stworzenie biblioteki współpracującej z istniejącym pakietem statystycznym. Napisanie samodzielnego programu ma tę przewagę, że będzie z pewnością dużo wydajniejsze. Ma jednak zasadniczą wadę: języki programowania typu C czy Java są bardzo mało popularne wśród ekonomistów. Należy zatem, za cenę być może straty wydajności, skłonić się ku stworzeniu biblioteki do pakietu statystycznego, aby opracowane narzędzie mogło być również wykorzystywane w przyszłości przez innych.

Przygotowania do stworzenia własnej biblioteki należy rozpocząć od zapoznania się z koncepcją oprogramowania o otwartym kodzie (ang. *open source*). Po tym wstępie, dokonany zostanie przegląd wybranych pakietów statystycznych oraz dokonany zostanie wybór środowiska dla tworzonej biblioteki. Następnie wybrane środowisko zostanie zaprezentowane szczegółowo. W ostatnim podrozdziale omówiona zostanie architektura i sposób realizacji przygotowanej biblioteki.

#### 3.1. Oprogramowanie otwarte

##### 3.1.1. Definicja

Aby program można było określić mianem **otwartego oprogramowania** (ang. *open source*), licencja, na podstawie której jest udostępniany, musi spełniać szereg warunków. Podaje je Open Source Initiative<sup>10</sup>:

- darmowa dystrybucja

---

<sup>10</sup><http://www.opensource.org/>

- dostępność kodu źródłowego (program musi być dystrybuowany wraz z kodem źródłowym)
- prace pochodne (licencja musi umożliwiać dokonywanie modyfikacji i wymuszać dystrybuowanie programów pochodnych na tej samej licencji)
- integralność kodu autora (licencja może zabraniać dystrybucji kodu źródłowego tylko jeżeli daje możliwość rozpowszechniania łątek (ang. *path*); przepis ten służy ochronie dorobku i reputacji autora oryginalnego programu)
- zakaz dyskryminacji osób (licencja nie może różnicować praw ze względu na osoby)
- zakaz dyskryminacji dziedzin zastosowania (licencja nie może ograniczać dziedzin zastosowania dla programu)
- dystrybucja licencji (licencja musi być zawsze dołączona do oprogramowania, również przy dokonywaniu jego redystrybucji)
- ogólność licencji (treść licencji nie może zależeć od konkretnego programu bądź jego przynależności do większej grupy – program wyjęty z większej całości (pakietu) wciąż objęty jest tą samą licencją)
- zakaz ograniczania innego oprogramowania (licencja nie może wpływać na inne współdystrybuowane programy – w tym samym pakiecie, na tym samym nośniku, itp.)
- niezależność licencji od technologii (licencja nie może być powiązana z konkretnym środkiem dystrybucji, tj. nie można zakładać, że oprogramowanie zawsze będzie rozpowszechniane za pomocą stron internetowych, gdzie przed pobraniem można wymusić zgodę na warunki licencji).

### 3.1.2. Wady i zalety

Dyskusja na temat wad i zalet otwartego oprogramowania toczy się od bardzo dawna. Raymond [1999] w ciekawy sposób obrazuje różnice między tymi dwoma podejściami. Oprogramowanie komercyjne przypomina budowę katedry – to misterna praca grupy czarodziej, zespołów, przeważnie pracujących w izolacji. Z kolei atmosfera pracy w modelu *open-source* bardziej przypomina bazar, gdzie panuje pewien pozornie chaotyczny gwar, gdzie każdy bierze żywiłowo udział w dyskusji, dorzuca coś od siebie. Gworny bazar ma swoje wady i zalety (zob. [Gunton, 2005]):

- wady otwartego oprogramowania:
  - dewaluacja pracy deweloperów. Skoro bowiem tak wiele tak rozbudowanych programów dostępnych jest za darmo do ściągnięcia z Internetu, użytkownicy przestają cenić pracę programistów
  - dąbotaż wolnego rynku. Producenci oprogramowania komercyjnego mają problemy ze



- sprzedają swoich produktów
- arogancka społeczność. Wokół programów *open-sourcowych* skupione są rozbudowane społeczności, w dużej części zbudowane z tzw. specjalistów-entuzjastów (ang. *geeks*), którzy traktują bezwzględnie osoby zadające niekoniecznie mądre pytania. Z jednej strony *open-sourcowi* twórcy pragną, by ich programy trafiły pod strzechy, ale z drugiej – gdy przeciętny użytkownik zadaje podstawowe pytania, odnoszą się do niego opryskliwie
  - nieskoordynowany indywidualizm. Projekty *open source* rodzą się przeważnie w umyśle jednego programisty, który odczuwa brak jakiegoś narzędzia, więc je tworzy. W ten sposób powstaje sporo narzędzi pokrywających się w funkcjonalności, brakuje koordynacji i współpracy
  - wybiórczość w działaniu. Gunton zauważa, że spektakularne funkcjonalności społeczności przeważnie realizuje szybko i z zapałem, a poprawki nieciekawych fragmentów kodu odpychane są w czasie
  - ciągle przepisywanie. Programistom sprawia dużo satysfakcji przepisywanie istniejących fragmentów programów, np. gdy muszą przejąć od kogoś kod. To, niestety, nie posuwa do przodu pracy
  - nieprzystępna dokumentacja. Opisy programów są często tworzone przez specjalistów dla specjalistów. Skrótowe i w języku nieprzystępnym dla przeciętnego użytkownika
  - zalety otwartego oprogramowania:
    - jest wolne. Licencje otwartego oprogramowania gwarantują, że żadna firma nie przywłaszczy go sobie, nie zamknie
    - wsparcie społeczności. Społeczność (użytkowników i twórców) zbudowana wokół projektu jest zwykle chętna do pomocy i rozwiązywania problemów
    - duży wybór gotowych rozwiązań. Jeżeli użytkownik ma jakiś problem do rozwiązania, istnieje duże prawdopodobieństwo, że ktoś już go rozwiązał. Co więcej, prawdopodobnie problem rozwiązano już na kilka sposobów
    - niezawodność i bezpieczeństwo. Dzięki użytkownikom-entuzjastom otwarte oprogramowanie jest zazwyczaj dobrze przetestowane
    - możliwość modyfikacji. Jeżeli program nie spełnia oczekiwań użytkownika, może on go sam dostosować, dzięki swobodnemu dostępowi do kodów źródłowych
    - motywuje producentów komercyjnych. Wolne projekty „podnoszą poprzeczkę” – sprawiają, iż producenci komercyjnych produktów muszą ciągle oferować nowe dodatkowe rzeczy, aby zachęcić klientów do zakupu ich produktu.

Podsumowując, Gunton przekonuje, iż nie da się jednoznacznie odpowiedzieć na pytanie, czy wolne oprogramowanie jest dobre czy złe. Twierdzi, iż wszystko zależy od punktu widzenia (przyzwyczajenia i wiedza użytkownika, wielkość firmy, dziedzina zastosowań) oraz – wpływającego czasu. Zauważa jednak, iż wolne oprogramowanie z pewnością uderza w jedno fundamentalne prawo – prawo do zarabiania na życie. Uważa, iż jest to jedyny powód, dla którego otwarte oprogramowanie może budzić wątpliwości.

### **3.1.3. Przykłady**

Poniżej znajduje się lista przykładowych projektów prowadzonych obecnie na licencjach typu *open-source*:

- Apache Software Foundation – producent popularnego serwera WWW (httpd)
- Debian, Fedora Project, FreeBSD, NetBSD, OpenBSD, OpenSuse – darmowe dystrybucje systemu Linux oraz Unix
- Eclipse Foundation – wydawca środowiska programistycznego Eclipse
- GIMP – producent profesjonalnego oprogramowania graficznego, przez wielu uważany za odpowiednik Photoshopa firmy Adobe
- GNU – system operacyjny bazujący na Unixie, a właściwie bardzo bogata kolekcja różnego rodzaju oprogramowania, m.in. kompilatora języka C (GNU C compiler), zaawansowanego edytora tekstów (emacs)
- Java – obiektowy język programowania oraz środowisko czasu wykonania, która umożliwia tworzenie programów działających na każdej platformie środowiskowej i sprzętowej, na której przygotowana jest tzw. wirtualna maszyna Javy
- Mozilla Foundation – wydawca m.in. przeglądarki Firefox
- MySQL – baza danych
- OpenOffice.org – zaawansowany pakiet biurowy (edytor tekstów, prezentacje, arkusz kalkulacyjny, baza danych, edytor graficzny)
- PHP – skryptowy język dynamicznie generowanych stron internetowych
- Python – język programowania obiektowego.

### **3.2. Narzędzia ekonomistów**

Na rynku dostępnych jest kilkanaście dużych pakietów ułatwiających pracę w analizie statystycznej i ekonometrycznej danych. Należą do nich m.in. EViews, GAUSS, gretl, Mathematica, R, SAS, Stata, Statistica, SPSS. Nie jest celem tej pracy szczegółowe omówienie

wszystkich pakietów. Poniżej zaprezentowana zostanie krótka charakterystyka dwóch najbardziej popularnych darmowych (R i gretl) oraz dwóch komercyjnych pakietów (EViews i SAS). W opisach wskazane zostanie ich pochodzenie oraz wady i zalety. Pozwoli to wybrać najlepsze środowisko do tworzonej implementacji ewolucyjnego modelowania ekonomicznego.

## **Pakiet R**

R jest pakietem obliczeniowym dla statystyków. Praca z nim odbywa się głównie w trybie tekstowym za pomocą tzw. linii poleceń. Istnieją jednak interfejsy graficzne (np. JGR napisany w Javie). Pracę ułatwiają również wtyczki do popularnych edytorów tekstów (Emacs, Kate, Tinn, Vim, WinEdt) oraz środowiska programistycznego Eclipse. Pakiet R stworzony został przez Rossa Ihaka oraz Roberta Gentlemana na Uniwersytecie w Auckland (Nowa Zelandia). Obecnie projekt prowadzi grupa *R Development Core Team*. Pakiet ten dostępny jest nie tylko w wersji dla systemu operacyjnego Windows, ale również Mac OS X oraz kilku dystrybucji Linuxa i Unixa.

Dużą zaletą pakietu R jest bogata biblioteka kodu tworzonego przez użytkowników. R posiada również użyteczne procedury rysowania wykresów, wizualizacji danych oraz eksportu efektów pracy do różnych formatów. Ułatwia również stosowanie symboli matematycznych. Bardzo istotną cechą pakietu, znacznie przyczyniającą się do jego sukcesu, jest jego darmowość oraz otwartość kodu. Dyskusja na temat zalet i wad oprogramowania z otwartym kodem źródłowym znajduje się w rozdziale 3.1.

## **Program gretl**

Pakiet gretl uważany jest za darmową alternatywę dla EViewsa. Jego nazwa to skrót od *GNU Regression, Econometrics and Time-series Library*, czyli Biblioteka GNU do regresji, ekonometrii i szeregów czasowych. Ojcem programu jest Allin Cottrell z Uniwersytetu w Wake Forest (Stany Zjednoczone). Obecnie program zyskuje popularność i grono jego twórców poszerza się. Pakiet gretl dostępny jest w wersji Linuxowej oraz Windowsowej i dla Mac OS. Programem można sterować przy użyciu linii poleceń lub interfejsu graficznego. Program oferuje możliwość generowania wykresów, do czego wykorzystuje zewnętrzny program gnuplot. Program prowadzony jest na zasadzie otwartego kodu źródłowego.

Jak sami autorzy gretla przyznają na stronie internetowej, gretl nie jest narzędziem idealnym dla statystyków, celuje głównie w ekonometryków. Aby jednak wypełnić statystyczne braki, przystosowany został do współpracy z pakietem R.

## **Pakiet EViews**

EViews jest pakietem statystycznym dedykowanym analizie ekonometrycznej. Pracuje w trybie graficznym, opierając się na podejściu wskaż-i-wybierz (ang. *point-and-click*). Jego częścią jest jednak również język programowania. Pakiet ten dostępny jest jedynie dla użytkowników systemu Windows.

Projekt prowadzi firma Quantitative Micro Software. Program ten jest płatny. Koszt pojedynczej licencji akademickiej to około 500 dolarów amerykańskich, a komercyjnej – 1000.

## **Pakiet SAS**

SAS, produkt firmy SAS Institute, jest bardzo rozbudowanym narzędziem do zarządzania zbiorami danych, przygotowywania raportów i wykresów oraz analizy statystycznej. Pakiet wspiera również przeprowadzanie złożonych obliczeń na zdalnych komputerach. Praca z SASem odbywa się głównie za pomocą linii poleceń. Do operowania na danych wykorzystywany jest język zapytań SQL (ang. *Structured Query Language*), tzw. język czwartej generacji (4GL). System SAS jest produktem komercyjnym. Dostępny jest na platformę Windows, IBM mainframe, Unix oraz OpenVMS.

Do głównych wad SASa należy:

- ogromny koszt zakupu licencji
- ubogie możliwości przygotowywania wykresów i wizualizacji danych
- opóźnienie w implementacji różnych metod statystycznych w porównaniu do konkurencji.

## **Wybór środowiska**

W przekonaniu autora pracy, najkorzystniejsze dla środowiska naukowego okazać się może wybranie pakietu R jako platformy dla biblioteki do ewolucyjnego modelowania ekonomicznego. Przemawiają za tym dwa argumenty. Po pierwsze, R cieszy się rosnącą popularnością w środowiskach akademickich (zob. np. [Fox, Andersen, 2005] i [Jackman, 2003]). Ten wybór daje więc szansę na największe przysłużenie się innym. Po drugie, R jest otwartym oprogramowaniem, co oznacza przede wszystkim, że praktycznie każdy może je za darmo uzyskać, zainstalować i wykorzystywać. Jeżeli więc nawet potencjalny użytkownik proponowanej biblioteki nie korzystał z R, to bez problemu mógłby zacząć. Kwestia finansowa nie byłaby barierą. W dodatku R jest łatwym intuicyjnym językiem, którego można się szybko nauczyć.

### 3.3. Szczegółowy opis R

R jest językiem programowania zorientowanym na dane. Stanowi on darmową (*open-sourcową*) implementację dialektu języka S. Stąd jego alternatywna nazwa – GNU S.

Język S został opracowany w celu zatarcia różnicy między programistą a użytkownikiem. Na jego bazie powstał komercyjny system S-PLUS. Środowisko języka S powstało z myślą o interaktywnej analizie danych, gdzie naukowiec wydaje kolejne polecenie, oglądając pośrednie wyniki. Jest dobrym systemem do szybkiego tworzenia aplikacji statystycznych. R jest bliźniakiem S. Większość programów może być w prosty sposób przetłumaczona z jednego języka do drugiego.

Głównymi zarzutami w stosunku do języków S i R jest ich powolność i pamięciożerność. Autorzy środowiska R przekonują jednak, że w nowoczesnych komputerach na użytek zwykłej codziennej analizy nawet dużych ilości danych, wielkość pamięci operacyjnej nie jest żadnym ograniczeniem. Również moc komputerów stale rośnie, złożoność obliczeniowa stosowanych metod statystycznych natomiast nie zmienia się. W związku z tym z biegiem czasu waga wspomnianych wad zmniejsza się.

Ogromną zaletą środowiska R jest otwarte repozytorium pakietów (CRAN, ang. *Comprehensive R Archive Network*). To bogata kolekcja bibliotek stworzonych przez użytkowników systemu z całego świata, wraz z szczegółową dokumentacją i przykładami zastosowania. Każdy ma możliwość umieścić swoją bibliotekę w tym repozytorium i uczynić ją dostępną dla innych. Dzięki interfejsom systemu R w wielu językach programowania, dodatkowe biblioteki mogą być pisane nie tylko w języku R, ale również w językach Java, C i Fortran.

W kolejnych podrozdziałach zaprezentowane zostaną najciekawsze (z punktu widzenia przygotowanej biblioteki) cechy języka R.

#### 3.3.1. Typy danych

##### Wektory i operacje wektorowe

R operuje na nazwanych strukturach danych (ang. *data structures*). Najprostszą taką strukturą jest wektor liczbowy, czyli uporządkowany zbiór liczb. Poniżej zaprezentowana została instrukcja języka R, która wiąże wektor 5 liczb ze zmienną o nazwie  $x$ <sup>11</sup>.

<sup>11</sup>Interakcje ze środowiskiem R (wydawane komendy oraz wydruki programu) wyróżniane są w tekście czcionką maszynową. Linijki zaczynające się od > (znaku zachęty środowiska R) oznaczają komendy wydane systemowi, pozostałe – jego odpowiedzi. Czcionka ta używana jest również do wyróżnienia nazw funkcji i zmiennych.

```
> x <- c(15, 24, 33, 42, 51)
```

Wszystkie operacje języka R to operacje na wektorach. Jeżeli w wyrażeniu znajduje się pojedyncza liczba, jest ona traktowana jako wektor długości jeden z tą liczbą jako jedynym elementem. Dzięki temu, wszystkie operacje na zestawach danych są bardzo proste. Na przykład obliczenie wariancji dla próbki w wektorze  $x$  można wykonać w sposób prezentowany poniżej<sup>12</sup>.

```
> mean(x*x) - mean(x)^2
```

`mean` jest wbudowaną funkcją, która dla zadanego wektora liczb zwraca ich średnią, a  $^2$  oznacza potęgowanie. Należy przy tym pamiętać, iż `*` jest mnożeniem liczb wektorów – jego wynikiem jest wektor, który na  $i$ -tej pozycji ma iloczyn  $i$ -tych liczb z obu czynników. Nie należy mylić tej operacji z mnożeniem wektorowym (oznaczanym w R przez `% * %`).

### Pozostałe typy danych

W R znajdziemy również, obok wektorów, następujące typy danych:

- macierze (ang. *matrices*), będące *de facto* wielowymiarowymi wektorami
- listy (ang. *lists*), będące ogólniejszą postacią wektora – uporządkowanymi zbiorami elementów o niekoniecznie jednakowych typach. Poszczególne elementy mogą być nazwane. Dzięki temu, listy są wygodnym sposobem na zwracanie złożonych wyników bądź przekazywanie zestawu parametrów
- funkcje (ang. *functions*), będące zbiorem instrukcji w języku R na określonych parametrach wejściowych
- cechy (ang. *factors*), czyli zestawy atrybutów klasyfikujących poszczególne dane w próbce (np. wg płci). R dostarcza specjalnych operacji na cechach, ułatwiających klasyfikację próbki ze względu na wartości cechy
- ramki danych (ang. *data frames*), będące zestawami danych – kolekcją nazwanych macierzy (np. ramka danych pracownicy z wektorami płac, godzin pracy, podatku).

### 3.3.2. W kierunku języka obiektowego

Język R wspiera paradygmat programowania obiektowego. Warto przede wszystkim odnotować, że z każdą zmienną w R można skojarzyć atrybuty. Ze zmienną można powiązać pod dowolną nazwą dowolną wartość. To doskonały mechanizm do przechowywania meta

<sup>12</sup>Należy zauważyć, iż podany sposób nie jest jedyny i ma charakter jedynie poglądowy. Istnieje prostszy – wystarczy skorzystać z wbudowanej funkcji `var`.

informacji o zmiennej. Środowisko wykorzystuje go do przechowywania informacji o klasie obiektu przypisanego zmiennej.

Obiektowość w R jest bardzo luźna. R jest językiem zdecydowanie imperatywnym, wzbogaconym o mechanizm wyboru odpowiedniej wersji implementacji funkcji w zależności od atrybutu `class` ustawionego zmiennej.

Przykładem wykorzystania obiektowości w R jest funkcja `summary`, która dostarcza zwarte podsumowanie obliczonego modelu. Różne sposoby modelowania, mogą produkować modele o różnych cechach i charakterystykach (specyficzne wartości funkcji dopasowania, zapamiętane parametry modelu, itp.). Wraz z każdą funkcją modelującą (np. `lm` – dla modeli liniowych, `glm` – dla uogólnionych modeli liniowych) przygotowano również specjalne wersje funkcji `summary` (nazywające się, odpowiednio, `summary.lm` i `summary.glm`). R sam dba o to, aby wywołać odpowiednią dla podanego obiektu implementację metody. Poniżej przeanalizowana jest przykładowa interakcja z programem R, która obrazuje obiektowość R:

1) przygotowanie danych

```
> x <- c(1,2,3,4,5,6)
> y <- c(1,2,3,4,5,6)
```

2) zbudowanie modelu regresji liniowej

```
> model <- lm (x~y)
```

3) sprawdzenie klasy zmiennej `model`

```
> attr(model, "class")
[1] "lm"
```

Odpowiedź oznacza, że zmiennej `model` pod atrybutem o nazwie `class` przypisano wartość (napis) `lm`.

4) żądanie podsumowania.

```
> summary(model)
Call:
lm(formula = x ~ y, data = list(x = x, y = y))
Residuals:
      Min       1Q   Median       3Q      Max
-5.661e-16 -1.157e-16  4.273e-17  2.153e-16  4.167e-16
...

```

Ostatni punkt wymaga dodatkowego komentarza. Gdy R napotyka wywołanie takie jak `summary(model)`, sprawdza najpierw jaka klasa przypisana jest zmiennej `model`. W tym wypadku jest to `lm`. R poszukuje zatem definicji funkcji o nazwie `summary.lm`. Jeżeli taka istnieje (a istnieje), wywoływane jest `summary.lm(model)`. W podobny sposób zdefiniowanych jest wiele innych funkcji wbudowanych w R, np. `print` do drukowania obiektu, czy `plot` do jego wizualizacji na wykresie.

Opisane tu mechanizmy obiektowe są doskonałym ułatwieniem w pisaniu programów i codziennej pracy ze środowiskiem R. Nie są jednak tak pełną realizacją tego paradygmatu, jak języki w pełni obiektowe, np. Java czy C++. Nie występuje w R dziedziczenie, polimorfizm, przekazywanie komunikatów między obiektami. Nie ma w R również pojęcia klasy jako bytu określającego zestaw operacji i atrybutów. Wspomniane komponenty to uznane (zob. [Armstrong, 2006]) fundamentalne składowe języków obiektowych.

### 3.3.3. W kierunku języka funkcyjnego

Możliwość definiowania i przechowywania funkcji jako zmiennych nie jest jedyną cechą właściwą paradygmatowi funkcyjnemu, jaką można odnaleźć w R. Inną, bardzo istotną, jest możliwość manipulowania wyrażeniami zapisanymi w tym języku.

Użytkownik może poprosić o zapamiętanie przetworzonej (skompilowanej) wersji wyrażenia pod jakąś zmienną. Służy do tego operacja `quote`, która zwraca drzewko wyrażenia w postaci zagnieżdżonej listy. Przykłady działania tej operacji przedstawione zostały w tabeli 5.

Tabela 5. Przykłady wywołania funkcji `quote` w języku R

Wyrażenia	Wyniki operacji <code>quote</code>
1	1
1+2	<code>list('+', 1, 2)</code>
1+2*3	<code>list('+', 1, list('*', 2, 3))</code>
1+2*3-mean(x)	<code>list('+', 1, list('-', list('*', 2, 3), list('mean', x)))</code>

Źródło: Opracowanie własne.

Każde wyrażenie to albo stała lub zmienna (1, 2, 3, x) albo wywołanie funkcji wieloargumentowej, które reprezentowane jest jako lista. Pierwszy jej element stanowi dowiązanie do wywoływanej funkcji (+, \*, mean) a kolejne – wszystkie argumenty funkcji. Argumenty te są wyrażeniami, które definiuje się w ten sam sposób, tj. są albo stałą lub zmienną, albo wywołaniem funkcji.



Gdy całe wyrażenie zostanie zapamiętane pod jakąś zmienną, można dowolnie manipulować jego treścią, zmieniając wywoływane funkcje, czy wartości liczbowe albo zmienne. Należy jedynie dbać o spójność arności funkcji z liczbą przypisanych argumentów. Poniżej zaprezentowany został przykład, zmieniający wyrażenie  $\sin(1+2*3)$  w  $\cos(1+2-x)$ .

```
> x <- 5
> w <- quote(sin(1+2*3))
> w[[1]] <- as.name("cos")
> w[[2]][[3]][[1]] <- as.name("-")
> w[[2]][[3]][[3]] <- x
> w
cos(1 + (2 - 1))
```

Procedura `as.name` wyszukuje i zwraca funkcję o zadanej nazwie. Operator podwójnych kwadratowych nawiasów umożliwia odwołanie do danego elementu listy. I tak, tuż po przypisaniu na `w` wyjściowego wyrażenia mamy: `w[[1]] = sin`, `w[[2]] = 1+2*3`, `w[[2]][[1]] = +`, `w[[2]][[2]] = 1`, `w[[2]][[3]] = 2*3`.

Operacją odwrotną do `quote` jest `eval`, która oblicza wartość zapamiętanego wyrażenia (jednoelementowy wektor liczbowy).

```
> eval(w)
[1] -0,4161468
```

Zauważmy, że możliwość swobodnego manipulowania wyrażeniami jest idealnym środkiem do realizacji koncepcji programowania genetycznego. Operacje genetyczne bezpośrednio tłumaczą się na operacje na elementach list – reprezentacji wyrażeń.

### 3.4. Opis implementacji algorytmu

Opisane powyżej cechy języka R ułatwiły dokonanie własnej implementacji programowania genetycznego. Przede wszystkim, procedura `quote` dała możliwość łatwego przekształcania wyrażeń w kolekcje obiektów. Te miały postać zagnieżdżonych list, które można bezpośrednio odwzorować na drzewka funkcji. Funkcyjność R umożliwiła łatwe operowanie na elementach tych drzewek – funkcjach elementarnych. Język ten daje możliwość modyfikowania wywołań funkcji w tak reprezentowanych wyrażeniach.

Dzięki wspomnianemu warsztatowi funkcyjnemu implementacja operacji genetycznych dokonana została w sposób intuicyjny. Opisane bowiem w rozdziale 2.3.4. teoretyczne operacje na abstrakcyjnych drzewkach tłumaczą się bezpośrednio na operacje na zagnieżdżonych

listach. I tak, mutacja polega na losowym wyborze poziomu zagnieżdżenia w liście oraz podmiany jednego z jej elementów (odpowiadających argumentom) na nowowygenerowaną listę. Podobnie krzyżowanie realizuje się poprzez losowy wybór poziomu zagnieżdżenia oraz argumentu w dwóch osobnikach i zamianę wskazanych obiektów pomiędzy osobnikami. Funkcyjność języka R pomogła również w wygodnym zdefiniowaniu ocenianych wersji funkcji (opisanych w rozdziale 2.3.3.). Każda z nich została zaprogramowana i zapamiętana pod osobną zmienną. Dzięki temu mogła być łatwo wykorzystana przy konstruowaniu drzewek.

Dostosowanie języka R do pracy na dużych ilościach danych doskonale przysłużyło się uproszczeniu wszelkich operacji obliczeniowych, zwłaszcza w kontekście obliczania miary dopasowania, omówionej w rozdziale 2.3.5. Kryterium MAPE dla pojedynczego osobnika można w programie wyliczyć przy użyciu jednej prostej instrukcji:

$$\text{mean}(\text{abs}(\text{eval}(I)/V-1))$$

gdzie:

- I – osobnik (zagnieżdżona lista wywołań funkcji elementarnych)
- V – wektor wartości.

W implementacji programowania genetycznego w środowisku R przydatna okazała się również obiektowość tego języka. Zaproponowana procedura zwraca użytkownikowi model w postaci obiektu. Implementacja dostarcza również podstawowych metod, którymi na tym obiekcie można operować: `summary` do wyświetlania podsumowania modelu oraz `predict` do dokonywania prognoz.

Wszystkie operacje samego modelowania ewolucyjnego przy użyciu programowania genetycznego zamknięte zostały w jedną definicję funkcji, która nazwana została `em`. Przyjmuje ona trzy parametry wejściowe: zmienne, wartości oraz parametry. **Zmienne** (`vars`) są wektorem wartości (jeżeli w modelu występuje tylko jedna zmienna) lub listą wektorów (jeżeli występuje więcej zmiennych). **Wartości** (`vals`) to wektor wartości zmiennej objaśnianej. **Parametry** (`control`) z kolei są opcjonalnym argumentem funkcji, za pośrednictwem którego można sterować parametrami programowania genetycznego. Parametry te zostały zestawione w tabeli 6.

Praca algorytmu sterowana jest jeszcze jednym nie wymienionym w tabeli parametrem, którego nie można kontrolować bezpośrednio. Jest to odsetek osobników pokolenia, jakie powstają w drodze reprodukcji. Parametr ten wynika z wartości parametrów `XOVER_FRACTION`, `MIGRATION_FRACTION` oraz `MUTATION_FRACTION` – razem z nimi

Tabela 6. Zestawienie parametrów konfiguracyjnych procedury em wraz z wartościami domyślnymi i opisami

Nazwy parametrów	Wartości domyślne	Opisy
N	500	wielkość populacji
INI_DEPTH	5	maksymalna wysokość nowotworzonych drzew funkcji
MAX_DEPTH	15	maksymalna wysokość drzew funkcji
CONST_RANGE	(-1, 1)	zakres, z którego losowane są stałe
XOVER_FRACTION	0,6	odsetek osobników pokolenia, jakie powstają w drodze krzyżowania
MIGRATION_FRACTION	0,1	odsetek osobników pokolenia, jakie powstają w drodze migracji
MUTATION_FRACTION	0,2	odsetek osobników pokolenia, podlegających mutacji
MAX_ITERATIONS	200	maksymalna liczba pokoleń w ewolucji
TERMINAL_FITNESS	0,0	docelowa wartość funkcji dopasowania
FITNESS_FN	MAPE	funkcja dopasowania (możliwe wartości: RSQ ( $R^2$ ), SSE i MAPE)
LAGS	10	liczba opóźnień – liczba wprowadzanych do modelu zmiennych opóźnionych

Źródło: Opracowanie własne.

musi się sumować do 1.

Algorytm zawarty w powstałej procedurze em jest realizacją schematu opisanego w rozdziale 2.3.7. Wykonuje on następujące kroki:

- 1) sprawdzenie parametrów wejściowych procedury
- 2) zainicjowanie puli funkcji elementarnych: wzbogacenie podstawowej puli funkcji (+, -, \*, %, cos, sin, exp, RLOG) o zmienne z parametru vars oraz o żadaną liczbę ich opóźnień (parametr LAGS)
- 3) zainicjowanie pierwszego pokolenia (N nowych osobników)
- 4) obliczenie wartości (wybranej) funkcji dopasowania, normalizacja wartości
- 5) zapamiętanie najlepszego osobnika (o ile jest lepszy od najlepszego dotychczas znalezionego)
- 6) sprawdzenie celu (liczba iteracji – MAX\_ITERATIONS i dopasowanie najlepszego osobnika – TERMINAL\_FITNESS) – jeżeli jest spełniony, przejście do punktu 9
- 7) przygotowanie nowego pokolenia poprzez krzyżowanie, mutację, reprodukcję i migrację osobników zgodnie z zadanymi parametrami
- 8) przejście do punktu 4

9) wydrukowanie najlepszego napotkanego osobnika oraz zwrócenie użytkownikowi modelu.

Dokonana implementacja została przetestowana na przykładowym procesie generowania danych. Wygenerowano dwie zmienne ( $x$  i  $y$ ) jako iloczyn kartezjański zbiorów  $[1, 20] \times [1, 10]$ , uzyskując w ten sposób 200 par:  $\{(1, 1), (1, 2), (1, 3) \dots (1, 10), (2, 1), (2, 2) \dots (20, 10)\}$ . Na zmiennych tych zdefiniowano i wyliczono wartości funkcji  $x * y + e^{x/y}$ . Procedurę em uruchomiono dla tak zdefiniowanych zmiennych oraz wartości. Już po zaledwie 7 pokoleniach algorytm odnalazł właściwą funkcję, która posłużyła do wygenerowania tych danych.

## ROZDZIAŁ IV

### Efektywność warszawskiej Giełdy Papierów Wartościowych

Poprzednie rozdziały wykazały, że programowanie genetyczne może być interesującym narzędziem modelowania genetycznego. Badania ekonomistów pokazują, że polem modelowania o szczególnym potencjale jest giełda. W niniejszym rozdziale zaprezentowane zostanie nowe badanie warszawskiej Giełdy Papierów Wartościowych. Punktem wyjścia dla niego jest **hipoteza efektywności rynku**, która została szczegółowo omówiona w rozdziale I. Metodologią badania jest **ewolucyjne modelowanie ekonomiczne**, opisane w rozdziale II. Z kolei narzędziem opisywanego badania jest **własna implementacja** ewolucyjnej koncepcji, zaprezentowana w rozdziale III.

Bezpośrednią inspiracją dla prezentowanego badania była praca Chena, opisana szczegółowo w rozdziale 1.3.2. Jednakże tamto badanie miało za cel jedynie przetestowanie koncepcji programowania genetycznego w praktyce, bez interpretacji ekonomicznej. Prezentowane w tym rozdziale badanie jest rozszerzeniem wspomnianej pracy. Chen zbadał jeden wybrany okres tajwańskiej giełdy i amerykańskiego indeksu S&P 500. Jego celem było udowodnienie, że programowanie genetyczne jest dobrym narzędziem. Zbudowany przez niego metodami ewolucyjnymi model okazał się lepszy od modeli autoregresyjnych. Badanie oparte zostało na wnioskach Chena, że programowanie genetyczne może być dobrym narzędziem modelowania ekonomicznego, zwłaszcza na polu giełdy i efektywności informacyjnej.

#### 4.1. Przebieg badania

Modelowaniu poddane zostały wartości indeksu WIG. Zostały one zaczerpnięte z portalu <http://www.bossa.pl/>. Dzielne notowania indeksu zostały przekształcone do postaci logarytmicznych stóp zwrotu według wzoru zaprezentowanego poniżej.

$$R_t = \ln \frac{I_t}{I_{t-1}}$$

gdzie:

$I_t$  – notowanie indeksu WIG z okresu  $t$

$R_t$  – logarytmiczna stopa zwrotu z indeksu WIG.

Do badania wzięte zostały dane za okres od 1994.11.24 do 2007.07.17, w sumie 3155 obserwacji. Szereg ten został podzielony na 63 ciągi po 50 obserwacji. Każdy taki ciąg stanowił okres *in-sample* dla modelowania, a kolejne 5 obserwacji – *post-sample*. Długość

poszczególnych okresów zostało dobrana tak jak we wspomnianym badaniu Chena.

Dla każdej próbki 50 obserwacji uruchomiony został algorytm genetyczny (opisany w rozdziale III), który wyłaniał w drodze ewolucji 5 najlepszych modeli. Modele te były używane do formułowania prognoz na 5 okresów w przyszłość. Wyniki prognozy konfrontowane były z rzeczywistymi danymi i na tej podstawie wyliczana była wartość średniego procentowego błędu (MAPE). Do modelowania użyto zestawu domyślnych parametrów programu, zaprezentowanych w tabeli 6 na stronie 58. Niestety, pomimo swojej liczności (500 osobników), populacja bardzo szybko wytracała zdolność poprawiania jakości osobników. Próby zmian parametrów, takich jak wielkość zjawiska mutacji, czy krzyżowania, nie przynosiły żadnych rezultatów. Fakt ten jest przesłanką do stwierdzenia, że dane giełdy warszawskiej cechują się dużą nieregularnością.

Dodatkowo, dla każdego z 63 ciągów zbudowano model referencyjny. Do tego celu użyty został model autoregresyjny. Model ten nie jest aż tak zaawansowany jak np. ARCH i GARCH, jednak ma nad nimi pewną przewagę. Po pierwsze, modele autoregresyjne są bardzo proste w konstrukcji. Po drugie, w modelach (G)ARCH wybór odpowiedniej ich postaci zależy wyłącznie od umiejętności badacza, który nie ma do dyspozycji żadnych testów potwierdzających jego decyzję. W przypadku modeli autoregresyjnych, wybór najwłaściwszej ich postaci może zostać dokonany na podstawie kryteriów informacyjnych (np. Akaike'a). I tak, w pracy dla każdego okresu z osobna wybrano liczbę opóźnień do modelu wg kryterium informacyjnego Akaike'a. Model ten posłużył do skonstruowania prognozy, z której wyliczona została wartość MAPE. Wyznaczenie jakości prognoz modelu autoregresyjnego umożliwiło ocenę prognoz modeli ewolucyjnych<sup>13</sup>.

#### **4.2. Przekrojowe wyniki modelowania**

Wyniki modelowania zostały podsumowane w tabeli 7, która prezentuje najważniejsze charakterystyki. W niej oraz w kolejnych tabelach i rysunkach zastosowano następujące dwa skróty: PG na oznaczenie modeli programowania genetycznego oraz AR dla modeli autoregresyjnych.

Należy zauważyć, że modele programowania genetycznego osiągają znacząco lepsze rezultaty (tj. średnia wartość MAPE prognozy) niż modele autoregresyjne. Jednak średnia jakość prognozy pozostaje na poziomie procesu błędzenia losowego (1). Te sumaryczne dane sugerują, że modele autoregresyjne są dużo bardziej nieprzewidywalne. Ich MAPE zawiera

---

<sup>13</sup>Skonstruowanie modeli oraz prognoz dla wszystkich okresów zajęło średniej klasy domowemu komputerowi prawie 2 doby.

Tabela 7. Charakterystyki miary MAPE dla prognoz wykonanych przy użyciu modeli programowania genetycznego i autoregresyjnych

Charakterystyki MAPE	Modele	
	PG	AR
Średnia wartość	1,045	1,353
Odchylenie standardowe	0,270	0,753
Minimum	0,704	0,515
Maksimum	2,984	4,305

Źródło: Opracowanie własne.

się w znacząco większym przedziale wartości niż dla modeli programowania genetycznego, również średnia i odchylenie standardowe są większe. Szczegółowe wartości MAPE dla poszczególnych modeli zostały opisane w tabeli 8 oraz zwizualizowane na rysunku 10.

Rysunek potwierdza wnioski wyciągnięte z charakterystyk – wartości MAPE dla modeli programowania genetycznego wydają się być dużo bardziej skupione wokół swojej średniej, która jest niższa niż dla modeli autoregresyjnych. Dokładna analiza tabeli wyników pokazuje, iż modele programowania genetycznego okazują się lepsze od autoregresyjnych w 38 przypadkach, a gorsze – w 25. W przypadkach, gdy modele programowania genetycznego okazywały się lepsze, były one lepsze o średnio 0,36 punktu MAPE, gdy były gorsze, to średnio o 0,06 punktu.

Należy również zauważyć, że jakość prognoz, zarówno modeli programowania genetycznego jak i autoregresyjnych, nie wykazuje wyraźnych różnic w swojej zmienności w czasie. Ich MAPE wydaje się być jednostajnie zdywersyfikowane w czasie.

Podsumowując należy stwierdzić, że modele programowania genetycznego osiągają bez wątpienia przeciętnie dużo lepsze rezultaty niż modele autoregresyjne. Ich jakość jednakże nie jest zadawalająca (MAPE średnio lekko powyżej 1).

#### 4.3. Analiza modeli otrzymanych w toku ewolucji

W tym podrozdziale omówione zostaną modele (5 najlepszych w myśl kryterium MAPE dla okresu *in-sample*) uzyskane dla wybranego, ostatniego okresu. Dla czytelności zapisu, dłuższe równania modeli zostały podzielone na fragmenty i zapisane jako osobne równania.

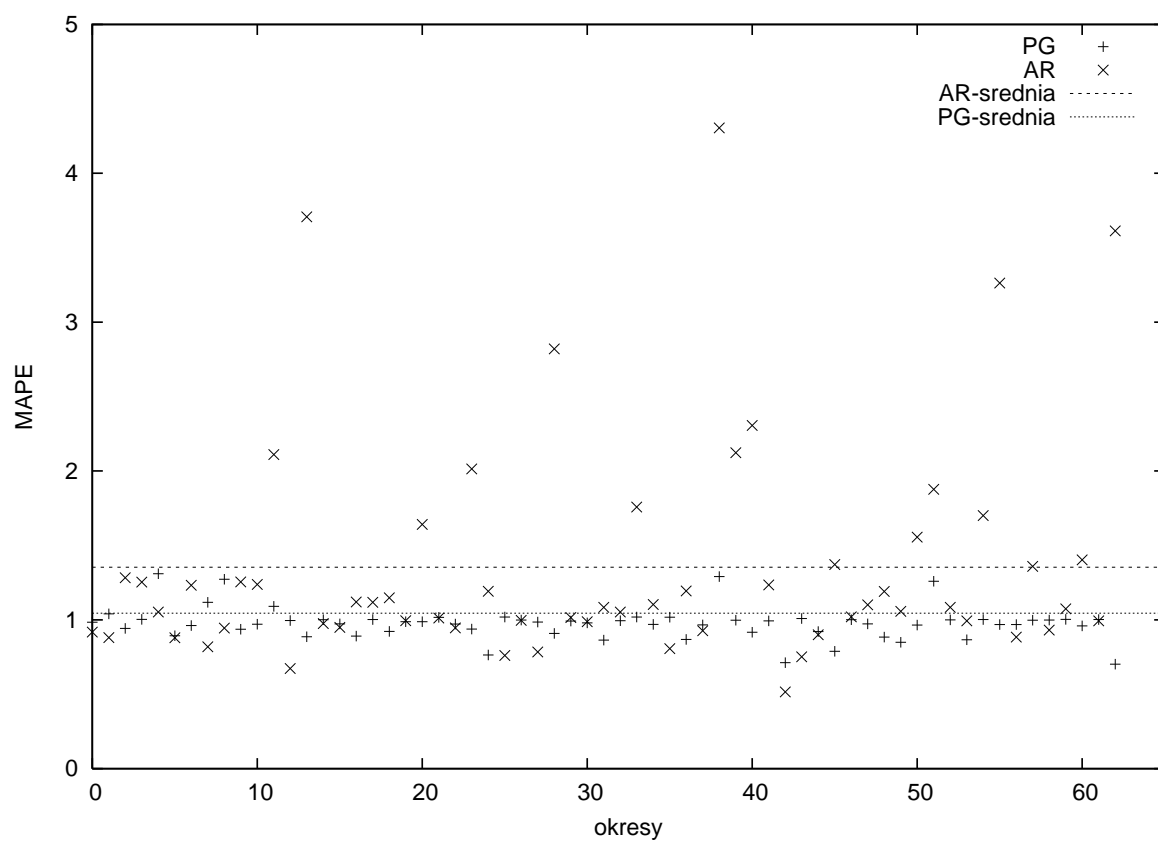
Tabela 8. Zestawienie wartości MAPE prognoz z modeli programowania genetycznego oraz autoregresyjnych dla poszczególnych okresów dla warszawskiej giełdy

Lp.	Początki okresów	Wartości MAPE dla modeli		31.	2000.11.28	1,000488	0,9872609
		PG	AR				
1.	1994.11.24	1,040052	0,9177668	32.	2001.02.09	0,873360	1,0833340
2.	1995.02.06	1,056631	0,8801674	33.	2001.04.24	1,033856	1,0527519
3.	1995.04.18	0,999973	1,2829736	34.	2001.07.06	1,054809	1,7580306
4.	1995.07.04	1,154904	1,2528478	35.	2001.09.17	0,999634	1,1031769
5.	1995.09.14	1,309696	1,0527842	36.	2001.11.27	1,082876	0,8064261
6.	1995.11.24	0,894906	0,8794164	37.	2002.02.11	0,934721	1,1942037
7.	1996.02.07	1,087051	1,2315016	38.	2002.04.24	0,975932	0,9261253
8.	1996.04.17	1,190742	0,8186617	39.	2002.07.08	2,984283	4,3052338
9.	1996.07.03	1,329112	0,9443797	40.	2002.09.17	1,013101	2,1221517
10.	1996.09.12	0,977088	1,2541749	41.	2002.11.18	1,006082	2,3062612
11.	1996.11.25	0,996747	1,2375832	42.	2003.02.12	1,004724	1,2343355
12.	1997.02.07	1,115722	2,1098329	43.	2003.04.25	0,712250	0,5153317
13.	1997.04.22	1,002058	0,6732592	44.	2003.07.08	1,022195	0,7512363
14.	1997.07.04	0,928453	3,7073025	45.	2003.09.17	0,992118	0,9001126
15.	1997.09.15	1,003160	0,9747706	46.	2003.11.27	0,788770	1,3716236
16.	1997.11.26	0,973162	0,9488228	47.	2004.02.11	1,014171	1,0192102
17.	1998.02.11	1,009651	1,1195495	48.	2004.04.23	1,018737	1,1007532
18.	1998.04.27	1,007482	1,1172529	49.	2004.07.06	0,955088	1,1899311
19.	1998.07.09	0,961577	1,1478338	50.	2004.09.14	0,935321	1,0570779
20.	1998.09.17	1,022367	0,9935331	51.	2004.11.25	1,024757	1,5553537
21.	1998.11.27	1,020696	1,6413939	52.	2005.02.03	1,337589	1,8768658
22.	1999.02.10	1,067933	1,0114455	53.	2005.04.19	1,006891	1,0841750
23.	1999.04.23	1,007833	0,9455546	54.	2005.06.30	0,878726	0,9932015
24.	1999.07.07	0,982899	2,0142276	55.	2005.09.09	1,003104	1,7001952
25.	1999.09.15	0,988124	1,1918010	56.	2005.11.22	1,042648	3,2631202
26.	1999.11.29	1,039712	0,7602366	57.	2006.02.11	0,982330	0,8834764
27.	2000.02.10	0,999205	0,9967074	58.	2006.04.12	1,050362	1,3590996
28.	2000.04.20	0,985093	0,7839855	59.	2006.06.28	1,013490	0,9314638
29.	2000.07.07	1,090692	2,8200631	60.	2006.09.07	1,045483	1,0733042
30.	2000.09.18	1,039377	1,0158220	61.	2006.11.17	1,008808	1,4032994
				62.	2007.01.31	1,021468	0,9932487
				63.	2007.04.13	0,704403	3,6126064

Źródło: Opracowanie własne.



Rys. 10. Zmienność wartości MAPE w zależności od czasu dla modeli ewolucyjnych i auto-regresyjnych dla warszawskiej giełdy



Źródło: Opracowanie własne.

Model nr 1, MAPE okresu treningowego 0,8601127, testowego – 1,097272.

$$\begin{aligned}
 M_1 &= 0,082R_6e^{R_8-0,584} \log M_1^{(1)} \\
 M_1^{(1)} &= R_4R_5R_9 \sin M_1^{(2)} - \frac{0,0495}{R_1} - \log \sin 0,186 + \log(R_2 - R_8) + R_2R_5 - R_{10} + R_3 \\
 M_1^{(2)} &= \frac{R_2(R_3 - R_6 - R_9 - R_{10})(R_2 - 0,492)R_3e^{R_6}}{\sin -0,386} + (e^{R_6} + 0,201 + R_3) \left( R_1 + R_6 + \frac{R_3}{R_{10}} \right)
 \end{aligned}$$

Model nr 2, MAPE okresu treningowego 0,8601127, testowego – 1,097272. Model ten różni się od poprzedniego jedynie drugim czynnikiem w liczniku pierwszego ułamka w trzecim równaniu. Element ten został wyróżniony pogrubieniem.

$$\begin{aligned}
 M_2 &= 0,082R_6e^{R_8-0,584} \log M_2^{(1)} \\
 M_2^{(1)} &= R_4R_5R_9 \sin M_2^{(2)} - \frac{0,0495}{R_1} - \log \sin 0,186 + \log(R_2 - R_8) + R_2R_5 - R_{10} + R_3 \\
 M_2^{(2)} &= \frac{R_2(R_3 - R_6 - R_9 - R_{10})(R_2 - \mathbf{R_7})R_3e^{R_6}}{\sin -0,386} + (e^{R_6} + 0,201 + R_3) \left( R_1 + R_6 + \frac{R_3}{R_{10}} \right)
 \end{aligned}$$

Model nr 3, MAPE okresu treningowego 0,8920879, testowego – 0,6905776.

$$M_3 = (-0,885R_6 + R_2R_9 + R_6) \cos R_2$$

Model nr 4, MAPE okresu treningowego 0,900782, testowego – 0,7044034.

$$M_4 = (-0,885R_6 + R_2\mathbf{R_5} + R_6) \cos R_2$$

Model nr 5, MAPE okresu treningowego 0,9072524, testowego – 0,970894.

$$M_5 = \frac{R_6}{R_2 + (R_{10} + R_9) - \log R_1}$$

Należy zauważyć, że 5 najlepszych modeli uzyskanych w procesie ewolucji cechuje bardzo niska różnorodność. Zidentyfikować można trzy postaci modeli: realizacją jednej są modele 1 i 2, drugiej – 3 i 4, a trzeciej – 5. Prawdopodobnie wiąże się to ze specyficznym kształtem tak zwanego krajobrazu możliwych modeli. Krajobraz to takie uporządkowanie w przestrzeni wszystkich modeli, że osobniki podobne do siebie (np. różniące się jedną zmienną)

znajdują się w swoim bezpośrednim sąsiedztwie<sup>14</sup>. Zaobserwowane zjawisko wyjaśnić można w kontekście krajobrazu ewolucyjnego na dwa sposoby: może on być albo bardzo skomplikowany, pełen lokalnych ekstremów, albo spokojny, z kilkoma wyrazistymi ekstremami. W pierwszym przypadku bardzo trudno jest poruszać się algorytmom ewolucyjnym (i ogólnie – sztucznej inteligencji) po tak bogatym krajobrazie w poszukiwaniu najlepszego rozwiązania. Wymaga to zdolności do porzucania lokalnych ekstremów na rzecz poszukiwania globalnego. Z kolei w drugim przypadku ewolucja szybko wykrywa te ekstrema i intensywnie pracuje wokół nich, poszukując najlepszego modelu. Zważywszy na dość wysokie MAPE dla opisanych modeli, należy skłonić się ku drugiemu wyjaśnieniu.

Kolejna obserwacja to skomplikowana, nieinterpretowalna postać modeli. Objawia się to na dwa sposoby. Po pierwsze, występuje duże wyrafinowanie w zestawieniu trudnych ekonomicznie funkcji (jak np.  $\cos$ , czy  $\log$ ). Uniemożliwia to nadanie równaniom ekonomicznej interpretacji. Można je oceniać jedynie według kryteriów MAPE. Należy odnotować, iż jest to cechą charakterystyczną podejścia ewolucyjnego, które nie trzyma się żadnych narzuconych z góry form funkcyjnych modeli. Po drugie, skonstruowane przez ewolucję modele są bardzo zachłanne w używane zmienne. Odwołują się do wielu z dostępnych 10 opóźnień, w kolejności modeli z: 9, 10, 2, 3 i 5 opóźnień. W porównaniu z referencyjnymi modelami autoregresyjnymi, które średnio wykorzystywały 2-3 opóźnienia, modele ewolucyjne są zdecydowanie bardziej złożone również pod tym względem. Warto zauważyć, iż to z kolei nie jest cechą konieczną algorytmów ewolucyjnych. W przypadku równania równowagi na rynku pieniądza, opisanego w rozdziale 2.4.1., ewolucja była w stanie wyeliminować z puli funkcji elementarnych zmienne niepotrzebne do opisu danych. W przypadku giełdy natomiast należy wnioskować, iż brak jest wyrazistych regularności w danych i programowanie genetyczne stara się uchwycić każdy najdrobniejszy choćby ślad zależności.

Należy również zauważyć, że dla zaprezentowanych modeli MAPE okresu *in-sample* jest nieskorelowane z wartością tej miary dla okresu *post-sample*. Badania Chena opisane w rozdziale 1.3.2. pokazały, iż istnieje taka niebezpieczna pułapka przy modelowaniu ewolucyjnym. W jego badaniach, z upływem pokoleń, pojawiały się coraz lepsze modele w kontekście okresu treningowego i jednocześnie coraz gorsze w kontekście okresu testowego. Analiza

<sup>14</sup>Najprościej krajobraz programowania genetycznego można wyobrazić sobie przy poszukiwaniu wartości funkcji  $n$ -argumentowej. Wówczas krajobraz jest przestrzenią  $n$ -wymiarową, gdzie na każdej osi są wszystkie możliwe wartości danego argumentu funkcji. Krajobraz to zbiór wartości funkcji na całej dziedzinie. W przypadku modelowania samych funkcji, należy sobie wyobrazić przestrzeń krajobrazu jako nieskończenie wymiarową przestrzeń funkcji. Na każdym z wymiarów wartościami są funkcje elementarne (a więc również stałe i zmienne). Kolejne wymiary przestrzeni odpowiadają kolejnym poziomom zagnieżdżenia funkcji. Punktem tej przestrzeni jest zatem jedna funkcja (model). Wartością natomiast – MAPE dla danego modelu

wyników badania warszawskiej giełdy nie ukazuje takich wyraźnych zależności. Wprawdzie najlepszy model na okresie treningowym (pierwszy) jest jednocześnie najgorszym w podanym zestawie w okresie testowym, to jednak pogorszenie dopasowania w okresie treningowym nie zawsze powoduje poprawę właściwości prognostycznych modelu (np. różnica pomiędzy modelem 3 a 4).

#### 4.4. Porównanie jakościowe modeli pomiędzy okresami

W niniejszym rozdziale omówione zostaną modele z różnych okresów – po jednym najlepszym (w myśl MAPE okresu treningowego) z każdego z 5 ostatnich okresów, a więc obejmujących łącznie czas od 2006.06.28 do 2007.07.17. Modele te zostały zaprezentowane poniżej w kolejności chronologicznej.

Model z okresu 59, MAPE okresu treningowego 0,9014816, testowego – 1.015069.

$$M_{59} = R_{10} - \frac{R_9 R_2}{R_4 - R_{10}} * (\sin R_{10} - R_4 + R_9) \log R_5$$

Model z okresu 60, MAPE okresu treningowego 0,9378184, testowego – 0,9919576.

$$M_{60} = \frac{(R_3 + 0,487)(R_1 + R_9) \frac{\log R_5}{R_3 - R_{10}} - R_6 + 0,946}{\frac{R_2 + R_2 R_8 \frac{R_7 + R_1 + R_6 - R_{10} - R_5}{\sin R_8} - R_5}{R_6 R_7 R_{10}}} - M_{60}^{(1)}$$

$$M_{60}^{(1)} = \frac{R_6 R_8}{R_9 e^{R_9} \frac{e^{R_{10} - R_9 + R_5 - 0,777}}{R_7 R_6} + \log R_2 - M_{60}^{(2)} - M_{60}^{(3)}}$$

$$M_{60}^{(2)} = R_{10} \left( R_3 - \cos \left( \frac{R_1 R_2 R_6 R_{10} e^{R_{10} - R_1}}{e^{R_7} + R_3 - R_6} \right) \right)$$

$$M_{60}^{(3)} = \cos(R_1) \sin \left( \frac{R_8 R_9}{e^{R_9}} + 0,467 \right) \frac{R_6 R_8 - R_7 + R_1}{e^{\sin R_1}} R_1$$

Model z okresu 61, MAPE okresu treningowego 0,9246226, testowego – 0,8135894.

$$M_{61} = \frac{(-0,937 \sin(R_6 R_9 + 0,889 - R_7) - R_{10} + R_1) R_6}{\sin(\cos(R_6 + R_7)) \left( \cos \left( \frac{R_2}{\cos R_8} \right) + \frac{\log(R_3 R_5 R_{10})}{e^{R_4}} \right)}$$

Model z okresu 62, MAPE okresu treningowego 0,930596, testowego – 1,043945.

$$\begin{aligned}
 M_{62} &= \frac{M_{62}^{(1)} - 0,643 + R_6 - R_4 - R_1}{\log\left(\frac{R_3 R_2}{R_1}\right) - \frac{R_5 R_3 \sin R_4}{e^{\sin R_5}} + R_{10} - \log R_2 - M_{62}^{(2)} - M_{62}^{(3)}} \\
 M_{62}^{(1)} &= R_2 R_3 R_4 R_8 R_9 \frac{R_8 - \frac{R_{10} \log \sin R_3}{(0,672 - R_{10}) R_4 (R_9 + R_2)}}{R_5 R_5} \left( R_4 + R_2 R_6 - R_{10} + \frac{\frac{R_3}{R_{10}} - \frac{R_3 R_7 R_8 R_6}{R_4 R_7 - R_4 R_8}}{\frac{R_9}{R_1 - R_3 + R_5}} \right) \\
 M_{62}^{(2)} &= \frac{(R_5 + R_2) R_4 (R_8 + \sin R_4) \cos R_1}{e^{\cos(R_3 R_9) + \sin R_3 - R_5}} \\
 M_{62}^{(3)} &= \frac{(R_9 + 0,895)(R_3 - R_6 - \log R_4 + R_3 R_5)}{\cos \sin R_4}
 \end{aligned}$$

Model z okresu 63, MAPE okresu treningowego 0,8601127, testowego – 1,097272.

$$\begin{aligned}
 M_{63} &= 0,082 R_6 e^{R_8 - 0,584} \log M_{63}^{(1)} \\
 M_{63}^{(1)} &= R_4 R_5 R_9 \sin M_{63}^{(2)} - \frac{0,0495}{R_1} - \log \sin 0,186 + \log(R_2 - R_8) + R_2 R_5 - R_{10} + R_3 \\
 M_{63}^{(2)} &= \frac{R_2 (R_3 - R_6 - R_9 - R_{10})(R_2 - 0,492) R_3 e^{R_6}}{\sin -0,386} + (e^{R_6} + 0,201 + R_3) \left( R_1 + R_6 + \frac{R_3}{R_{10}} \right)
 \end{aligned}$$

Zaprezentowane modele cechuje ogromna różnorodność. Nie jest możliwe wskazanie żadnych nawet rodzin modeli, każdy z nich jest zupełnie innej postaci. W szczególności dwa najprostsze modele (dla okresów: 59 i 61) również nie wykazują cech wspólnych, choćby ze względu na liczbę wykorzystywanych opóźnień (odpowiednio: 5 i 10). Dwa z przytoczonych modeli wykorzystują wszystkie dostępne opóźnienia, dwa kolejne – 9, najprostszy natomiast – 5. Ta duża różnorodność modeli oznaczać może, że rodzaj prawidłowości w danych zmienia się bardzo dynamicznie w czasie (zważywszy krótkie okresy poszczególnych próbek). Nasuwa to sugestię, że może być warto przetestować również mniejsze i większe przedziały czasowe, aby sprawdzić czy wpłynie to na stabilność modeli. Może być również warto badać nie następujące po sobie okresy, ale płynniej przesuwać okno badania po dostępnych danych. Rozpoczynanie okresów treningowych w kolejnych dniach (a nie co 50) mogłoby pomóc uchwycić płynne przejścia pomiędzy modelami.

#### 4.5. Wnioski

Przeprowadzone badanie pozwoliło ocenić zastosowanie programowania genetycznego jako narzędzia modelowania ekonomicznego oraz zebrać nowe argumenty w dyskusji nad

efektywnością warszawskiej GPW. Modele budowane na drodze ewolucji wykazywały dużo lepsze własności prognostyczne niż referencyjne modele autoregresyjne (średnie MAPE okresu *post-sample* dla modeli PG wyniosło 1,045, a dla AR – 1,353). Modele ewolucyjne cechowała również dużo większa stabilność prognoz (w myśl wariancji MAPE). Podejście ewolucyjne wykazało również swoje wady. Najważniejszą jest to, że budowane przez programowanie genetyczne modele są na ogół bardzo skomplikowane i trudne w interpretacji ekonomicznej. Nie jest to jednak cechą immanentną podejścia ewolucyjnego, o czym przekonują badania Kozy nad rynkiem pieniądza, opisane w rozdziale 2.4.1. Skomplikowane formy modeli wiążą się prawdopodobnie ze specyfiką modelowanego rynku. Należy zatem stwierdzić, sugerując się przede wszystkim relatywnie dobrymi wynikami modeli, że programowanie genetyczne jest nie tylko ciekawą i innowacyjną formą modelowania, ale również skuteczną i wartą stosowania.

Poczynione w badaniu obserwacje nie pozwoliły niestety w sposób jednoznaczny wypowiedzieć się na temat efektywności informacyjnej w formie słabej warszawskiej GPW. Najlepsze modele dawały prognozy porównywalne z procesem błędzenia losowego. Uzyskanie modeli o wyraźnie lepszych własnościach prognostycznych pozwoliłoby stwierdzić brak efektywności informacyjnej giełdy. W obliczu rzeczywistych wyników modeli na tym rynku można jedynie stwierdzić, iż zebrane wyniki nie podważyły słabej efektywności informacyjnej warszawskiej GPW.

## PODSUMOWANIE

W pracy przedstawiono szczegółową historię powstania koncepcji efektywności informacyjnej rynków kapitałowych, której historia sięga lat sześćdziesiątych XX wieku. Omówiono jej ogólną definicję oraz trzy formy: słabą, półsilną i silną. Dokonano również przeglądu dotychczasowych badań, ze szczególnym uwzględnieniem badań nad rynkiem polskim. Praca przedstawia również dziedzinę obliczeń ewolucyjnych – młodego działa informatyki, który wywodzi się od darwinowskiej koncepcji pochodzenia gatunków. Opisano również jak programowanie genetyczne, jedna z metod obliczeń ewolucyjnych, może zostać wykorzystana do modelowania ekonomicznego oraz zaprezentowano przykładowe badania w tym nurcie. Praca przedstawia również szczegółowo opis stworzonego na jej potrzeby narzędzia do ewolucyjnego modelowania ekonomicznego. Zaimplementowano go w języku obliczeń statystycznych R.

Przygotowane narzędzie posłużyło do realizacji dwóch zasadniczych celów pracy: zwyfikowania efektywności informacyjnej warszawskiej GPW oraz sprawdzenia, czy programowanie genetyczne może przydawać się w pracy ekonomisty. Przeprowadzone zostało badanie giełdy za lata 1994-2007, w podziale na 63 szeregi długości 50 obserwacji. Dla każdego z nich wykonano modelowanie ewolucyjne, uzyskując kilka modeli. Te zostały następnie ocenione według jakości prognozy na okresie testowym. Za referencyjne przyjęto modele autoregresyjne.

Przeprowadzone badanie nie pozwoliło na udzielenie jednoznacznej odpowiedzi na pytanie o słabą efektywność informacyjną warszawskiej GPW. Uzyskane wyniki nie dają podstaw do podważenia tej efektywności, nie pozwalają jednak również stwierdzić, że rynek ten jest efektywny informacyjnie. Doświadczenia zebrane w pracy pozwalają natomiast skonstatować, że programowanie genetyczne jest ciekawą i skuteczną alternatywą dla klasycznej ekonometrii.

## BIBLIOGRAFIA

- Deborah J. Armstrong. The quarks of object-oriented development. *Commun. ACM*, 49(2): 123–128, 2006.
- Meredith Beechey, David Gruen, James Vickery. The efficient market hypothesis: A survey. Technical report, Reserve Bank of Australia, styczeń 2000.
- Sebastian Buczek. *Efektywność informacyjna rynków akcji. Teoria a rzeczywistość*. Szkoła Główna Handlowa w Warszawie, 2005.
- E. K. Burke, J. P. Newall, R. F. Weare. A memetic algorithm for university exam timetabling. W: *1st International Conference on the Practice and Theory of Automated Timetabling (ICPTAT'95, Napier University, Edinburgh, UK, 30th Aug - 1st Sept 1995)*, str. 496–503, 1995.
- Shu-Heng Chen, Chia-Hsuan Yeh. Toward a computable approach to the efficient market hypothesis: An application of genetic programming. *Journal of Economic Dynamics and Control*, 21:1043–1063, 1996.
- Richard Dawkins. *Samolubny gen*. Prószyński i S-ka, 1996.
- Korneel Duyvesteyn, Uzay Kaymak. Genetic programming in economic modelling. W: *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, str. 1025–1031, Edinburgh, Scotland, wrzesień 2005.
- Agoston Eiben, Jim Smith. *Introduction to evolutionary computing*. Springer, 2003.
- John Fox, Robert Andersen. Using the R statistical computing environment to teach social statistics courses, 2005.
- Peter Grünwald. A tutorial introduction to the Minimum Description Length Principle. W: red. P. Grünwald, I.J. Myung, M. Pitt, *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2005.
- Sanford Jay Grossman, Joseph Stiglitz. On the impossibility of informationally efficient markets. *American Economic Review*, 70(3):393–408, 1980.
- Michał Grotowski, Krzysztof Wyroba. Efektywność informacyjna polskiego rynku walutowego - analiza wstępna. *Bank i kredyt*, 1:65 – 79, 2004.
- Neil Gunton. Open source pros and cons, 2005.
- Simon Jackman. R for the political methodologist. *The Political Methodologist*, 11:20–22, 2003.
- Jacek Jakubowski, Rafał Sztencel. *Wstęp do teorii prawdopodobieństwa*. SCRIPT, 2001.
- A. Keane. The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques. *Acoustical Society of America Journal*, 99:2599–2603, kwiecień 1996.



- John R. Koza. A genetic approach to econometric modeling. W: red. Paul Bourguine, Bernard Walliser, *Economics and Cognitive Science*, str. 57–75. Pergamon Press, Oxford, UK, 1991.
- John R. Koza. The genetic programming paradigm: Genetically breeding populations of computer programs to solve problems. W: red. Branko Soucek, the IRIS Group, *Dynamic, Genetic, and Chaotic Programming*, str. 203–321. John Wiley, New York, 1992.
- John R. Koza. Genetic programming: On the programming of computers by means of natural selection. *Statistics and Computing*, 4(2), 1994.
- Halina Kwaśnicka. Obliczenia ewolucyjne. *Zeszyt Naukowy Sztuczna Inteligencja*, 1:9–22, 2002.
- Witold Kwaśnicki. Ekonomia ewolucyjna - alternatywne spojrzenie na proces rozwoju gospodarczego. *Gospodarka Narodowa*, 10:1–13, 1996.
- Witold Kwaśnicki. Ekonomia ewolucyjna - w poszukiwaniu alternatywnego wyjaśnienia rzeczywistości społeczno-gospodarczej. W: *VII Kongres Ekonomistów Polskich*, 2001.
- Andrew W Lo, A Craig MacKinlay. Data-snooping biases in tests of financial asset pricing models. *Review of Financial Studies*, 3(3):431–67, 1990.
- Spyros Makridakis. Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, 9:527–529, 1993.
- Eric S. Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary (O'Reilly Linux)*. O'Reilly, October 1999.
- Armin Shmilovici, Yael Alon-Brimer, Shmuel Hauser. Using a stochastic complexity measure to check the efficient market hypothesis. *Computational Economics*, 22(2-3):273–284, 2003.

## ZESTAWIENIE SPISÓW

### Wykaz skrótów

- AR – ang. *Auto Regression* – autoregresja lub modele autoregresyjne
- GPW – Giełda Papierów Wartościowych
- MAPE – ang. *Mean Absolute Percentage Error* – średni błąd procentowy w wartościach bezwzględnych
- PG – Programowanie Genetyczne lub modele zbudowane przy użyciu jego metod

### Spis tabel

Tabela 1.	Zestawienie wartości miar $\pi$ dla indeksów giełdy tajwańskiej i amerykańskiej wg badań Chena i Yeha .....	20
Tabela 2.	Powiązanie między teorią ewolucji a programowaniem ewolucyjnym .....	25
Tabela 3.	Zestawienie parametrów programowania genetycznego .....	39
Tabela 4.	Zestawienie wartości miar dopasowania w modelu równowagi na rynku pieniądza wg badań Kozy .....	43
Tabela 5.	Przykłady wywołania funkcji <code>quote</code> w języku R .....	55
Tabela 6.	Zestawienie parametrów konfiguracyjnych procedury <code>em</code> wraz z wartościami domyślnymi i opisami .....	58
Tabela 7.	Charakterystyki miary MAPE dla prognoz wykonanych przy użyciu modeli programowania genetycznego i autoregresyjnych .....	62
Tabela 8.	Zestawienie wartości MAPE prognoz z modeli programowania genetycznego oraz autoregresyjnych dla poszczególnych okresów dla warszawskiej giełdy .....	63

### Spis rysunków

Rys. 1.	Schemat przykładowej realizacji przebiegu operacji <i>crossing-over</i> dla dwóch ścieżek pomiędzy Warszawą a Amsterdamem .....	26
Rys. 2.	Schemat problemów optymalizacyjnych .....	28
Rys. 3.	Schemat problemów symulacyjnych .....	29

Rys. 4.	Obraz efektów pracy algorytmu ewolucyjnego nad projektowaniem ramienia satelity na tle wyjściowego projektu .....	30
Rys. 5.	Schemat problemów modelowania .....	30
Rys. 6.	Schemat drzewiastej reprezentacji funkcji $\sin(\log(1/y) * ((2 * x) + \cos(3)))$ .	34
Rys. 7.	Schematyczna prezentacja przykładowej realizacji procesu mutacji.....	36
Rys. 8.	Schematyczna prezentacja przykładowej realizacji procesu krzyżowania .....	37
Rys. 9.	Obraz najlepszego rozwiązania ewolucyjnego modelowania równowagi na rynku pieniądza.....	42
Rys. 10.	Zmienność wartości MAPE w zależności od czasu dla modeli ewolucyjnych i autoregresyjnych dla warszawskiej giełdy .....	64