

## 0.1. Wstęp

Celem pracy jest stworzenie systemu układającego strategię inwestycyjną za pomocą programowania genetycznego na podstawie istniejącej literatury i przedstawienie metod poprawy efektywności jego działania.



## Chapter 1

# Wprowadzenie

### 1.1. Heurystyczne rozwiązywanie problemów

Przykład, prezentujący istotę heurystyki jest podany w książce L. Rutkowskiego “Metody i techniki sztucznej inteligencji”. Zaproponowana jest sytuacja, w której ktoś zgubił szkło kontaktowe. Istnieją następujące sposoby jego szukania:

1. Szukanie ślepe - schylanie się i szukanie po omacku, które nie gwarantuje sukcesu.
2. Szukanie systematyczne - rozszerzanie przestrzeni rozwiązań w sposób metodyczny i zorganizowany. Gwarantuje sukces, ale jest bardzo czasochłonne.
3. Szukanie analityczne - wymaga rozwiązania równania matematycznego opisującego upadek szkła, z uwzględnieniem wszystkich czynników takich jak ruch powietrza czy siła ciężenia. Gwarantuje sukces, choć jest w praktyce niemożliwe.
4. Szukanie leniwe - polega na znalezieniu najbliższego optyka i kupieniu nowego szkła.
5. Szukanie heurystyczne - określamy przybliżony kierunek upadku i domyślamy się na jaką odległość może upaść szkło, a następnie przeszukujemy wybrany obszar. Jest to zachowanie najbardziej naturalne i najczęściej nieświadomie wybieramy właśnie taki sposób postępowania.

Warto zauważyć, że szukanie heurystyczne wykorzystuje wiedzę o przestrzeni rozwiązań. Podejmowane są świadome decyzje o kierunku poszukiwań, możemy także ocenić aktualne postępy. Heurystyka rozwiązuje dany problem w oparciu o jakąś kreatywną metodę taką jak metoda prób i błędów lub poprzez analogię. Wyznacza ona sposób przeszukiwania przestrzeni rozwiązań, eliminując z niej te, które nie doprowadzą do rozwiązania. Heurystyka nie zawsze zapewnia dojście do rozwiązania optymalnego, jednak często nie jest to konieczne. Jeśli przegląd zupełny, czyli w podanym przykładzie szukanie systematyczne nie zwraca wyników w zadowalającym czasie, oraz niemożliwe jest wyznaczenie matematycznego opisu problemu jak we wspomnianym szukaniu analitycznym, wtedy najlepszym rozwiązaniem jest heurystyka. Nie jest to jednak naturalny sposób rozwiązywania problemów dla komputera i nie istnieją formalne dowody na skuteczność jego działania. Mimo to, w praktyce jest to najlepszy sposób na rozwiązanie niektórych problemów

Mimo, że komputery mają znacznie większe możliwości przetwarzania danych niż człowiek, niektórych problemów nie da się po prostu przeanalizować w całości. W niektórych przypadkach przestrzeń rozwiązań jest tak duża, że sprawdzenie wszystkich możliwości rozwiązań może zająć zbyt dużo czasu. Takie podejście, zwane siłowym, (lepiej znane jest określenie “brute force”) jest proste w implementacji i wydaje się być naturalne dla komputerów.

Pierwsze naukowe próby heurystycznego rozwiązywania problemów niektórzy naukowcy [28] przypisują George’owi Polya. W 1945 Wydana została jego książka “How to solve It” [22] w której opisane były heurystyczne metody radzenia sobie z matematycznymi problemami. Polya twierdził, że problemy mogą być rozwiązywane przez ograniczony zbiór strategii, z których większość upraszcza problem. Mimo, że książka była skoncentrowana na matematycznych problemach, wiele propozycji sprawdzało się w problemach logicznych

W pracy z 1958 roku Herbert Simmon i Allen Newell [27], późniejsi laureaci nagrody Turinga za badania nad sztuczną inteligencją, problemy do których należy wykorzystać podejście heurystyczne nazwali “źle skonstruowanymi”, lub “źle zdefiniowanymi”. Źle zdefiniowane problemy to takie, które w przeciwieństwie do dobrze skonstruowane nie mogą być wyraźnie opisane (za pomocą konkretnych liczbowych zmiennych), ich cele nie mogą być jednoznacznie zdefiniowane przez funkcję celu i do tego nie można ich rozwiązać znanymi algorytmami.

W tym okresie pojawiły się pierwsze heurystyki rozwiązujące takie problemy, bazujące między innymi na podejściu zachłannym. Opierały się na iteratywnym wyborze najlepszej decyzji w każdym kroku działania algorytmu. Przykładami takich algorytmów jest algorytm Dijkstry do znajdowania najkrótszej ścieżki, lub algorytm Prima do znajdowania minimalnego drzewa rozpinającego [7].

W 1970 roku Lofi Zadeh, sławny z zaproponowania matematyki rozmytej, porównywał działanie ludzkiego mózgu do działania komputerów [2]. Mówił, że ludzki mózg operuje na niekonkretnych danych właśnie “rozmytych pojęciach” i potrafi wykorzystać tolerancję niepewności danych jak i rozwiązania problemu. Wtedy jednak nie potrafił tego dokonać żaden komputer. Mówił jednak, że mimo że budowa komputerów nie jest przystosowana do przetwarzania niepewnych, rozmytych danych, to jest możliwe takie zaprogramowanie go, aby takie działanie były możliwe. Wspomniął również o różnicy między tradycyjnymi obliczeniami - twardymi, które operują na konkretnych, precyzyjnych danych i obliczeniami miękkimi, które w trakcie obliczeń powinny, kiedykolwiek to możliwe, wykorzystywać tolerancję na niepewność [24].

Główne metody obliczeń miękkich rozwijały się w pewnym stopniu niezależnie od siebie. Pierwsza była wspomniana już teoria zbiorów rozmytych Lofi Zadeha. Kolejnym filarem są sieci neuronowe, które jednak były opóźniane przez małe możliwości obliczeniowe ówczesnych komputerów i dopiero ostatnio ta dziedzina nabrała prawdziwego rozpędu. Trzecim elementem są w końcu metaheurystyki.

Na początku metaheurystyki były reprezentowane jedynie przez metody ewolucyjne. Pierwszą, która zyskała uznanie była strategia ewolucyjna zaprezentowana przez Ingo Rachenberga i Hansa-Paula Schwafela [23]. Później L.J.Fogel opracował programowanie ewolucyjne [11], jednak oba podejścia były dość dalekie od dzisiejszych rozwiązań. J. Holland bazując na poprzednich pracach opracował algorytmy genetyczne [33]. Wprowadził pojęcie populacji i krzyżowania osobników. W tym okresie pojawiło się też samo pojęcie

metaheurystyka. Heurystyka oznacza znajdowanie lub odkrywanie przez metody prób i błędów [33]. Dzisiaj funkcjonuje wiele definicji metaheurystyk, nie jest to pojęcie łatwe w zdefiniowaniu. Na podstawie tego, czym jest w rzeczywistości metaheurystyka powstają całe prace naukowe. Jedna z nowszych definicji została stworzona przez Freda Glovera, twórcy oryginalnej definicji oraz Kennetha Sorensena [28]:

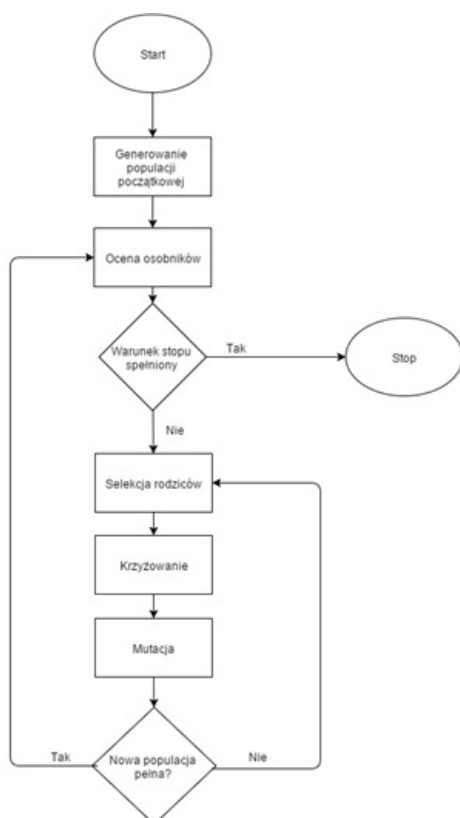
“Metaheurystyka to generyczny model algorytmiczny wysokiego poziomu który udostępnia zestaw wskazówek lub strategii do rozwoju heurystycznych algorytmów optymalizacji. Pojęcie jest również używane aby nawiązać do konkretnej implementacji heurystycznego algorytmu optymalizacji konkretnego problemu na podstawie wskazówek wyrażonych w danym modelu.”

Algorytm genetyczny znacznie zwiększył zainteresowanie naukowców metaheurystykami i w latach osiemdziesiątych i dziewięćdziesiątych powstało wiele nowych algorytmów, takich jak Tabu Search, czy Ant Colony Optimization.

## 1.2. Algorytmy ewolucyjne

Algorytm genetyczny został opisany przez Johna Hollanda w książce “Adaptation in Natural and Artificial Systems”. Jego ogólny schemat jest przedstawiony na rysunku 1.1. Ewolucja naturalna jest głównym sposobem rozwiązywania problemów przez naturę. Analogicznie działa algorytm genetyczny, oraz inne bazujące na niej metaheurystyki. Podstawowe założenie jest proste - Osobniki w naturze, wraz z kolejnymi generacjami przystosowują się do środowiska w którym żyją. Przełożenie tego na algorytm może być intuicyjne.

Figure 1.1: Ogólny schemat algorytmu genetycznego



Działanie rozpoczyna się od wylosowanie populacji o pewnym rozmiarze. Populacja składa się z osobników, reprezentowanych przez zestaw cech, zwanych fenotypem. Sposób kodowanie fenotypu w pamięci nazywa się natomiast genotypem. Każdy osobnik jest oceniany, poprzez specjalną funkcję przystosowania. Następnie losowane są osobniki, które będą krzyżowały się między sobą. Tak jak w naturze, większą szansę na wybór mają osobniki lepiej przystosowane, czyli takie o lepszej ocenie. Para osobników krzyżuje się, wymieniając cechami, niektóre cechy podlegają mutacji, zmieniając się losowo na inne. Proces powtarza się, póki nie zapełni się populacja nowych osobników. Każda kolejna populacja jest trochę lepsza, jednak losowa mutacja sprawia, że rozwiązanie nie utyka w lokalnym minimum.

Osobnik może na przykład reprezentować jedno rozwiązanie problemu kolorowania grafu. Jest to często analizowany problem NP-trudny, mający także praktyczne zastosowanie. Polega na przydzieleniu wierzchołkom grafu jak naj-

niejszej liczby kolorów tak, aby żaden wierzchołek nie był połączony z żadnym innym o takim samym kolorze. Osobnik w tym przypadku jest reprezentowany przez wektor, którego każdy element jest reprezentuje pokolorowanie innego wierzchołka, a funkcja oceny zlicza liczbę użytych kolorów.

Inną metaheurystyką opierającą się na analogii do ewolucji naturalnej jest programowanie genetyczne. Główną różnicą pomiędzy algorytmem ewolucyjnym jest reprezentacja osobnika. W tym przypadku osobnik jest reprezentowany przez drzewo, na którym przeprowadzane są takie same operacje, jak w klasycznym algorytmie ewolucyjnym. Ze względu na to, że osobnik nie ma określonej wielkości, reprezentacja ta jest bardziej elastyczna. Ten zysk jest jednak równoważony przez wzrost skomplikowania.

### 1.3. Inwestycje giełdowe

Giełda papierów wartościowych umożliwia handel akcjami spółek, określającymi prawo do części zysków i aktywów spółki [13]. Celem sprzedaży akcji jest pozyskanie kapitału, a inwestorzy w zamian mogą zarabiać poprzez dywidendy lub odsprzedaż akcji po wyższej cenie. Na giełdzie pracują maklerzy giełdowi, analitycy czy bankierzy inwestycyjni. Praca ich wszystkich jest związana z oceną i czasem też handlem papierami wartościowymi. Ze względu na ogromną liczbę czynników wpływających na cenę akcji, dokładne przewidzenie ruchów cen jest niemożliwe, jednak stosuje się do tego metody, które będą opisane w następnych rozdziałach.

Historię rynków finansowych można rozpocząć w siedemnastowiecznej Holandii. W tamtych czasach zamorski handel był okraszony bardzo dużym ryzykiem, przyćmiewanym jednak przez potencjalne zyski. Aby zminimalizować ryzyko, właściciele statków mieli zwyczaj szukania inwestorów, którzy mogliby wyłożyć pieniądze na wyprawę - wyposażenie statku i załogi. W zamian otrzymywali pewien procent zysków. W siedemnastym wieku zaczęły powstawać Kompanie wschodnioindyjskie, zajmujące się handlem z daleką Azją. Pierwsza była Holenderska Kompania Wschodnioindyjska, która po raz pierwszy w historii zaoferowała kupno swoich akcji. W ten sposób sprzedawany był udział nie tylko w jednej, a we wszystkich wyprawach prowadzonych przez Kompanię. Kompanie nie dość, że proponowały znacznie większą skalę przedsięwzięcia niż pojedyncze wyprawy, to dodatkowo otrzymały monopol od państwa, oferując ogromne zyski dla inwestorów.

Początkowo akcjami handlowano w kawiarniach, jednak po krótkim czasie powstały pierwsze giełdy papierów wartościowych i swoje akcje zaczęły sprzedawać Kompanie innych krajów. Korzystając ze swoich doświadczeń z giełdy holenderskiej, Joseph de la Vega opublikował w 1688 roku pierwszą książkę na temat giełdy "Confusion of Confusions" [9]. Książka nie traktowała jedynie o giełdzie, a autor zamieścił w niej wiele filozoficznych przemyśleń, biblijnych, mitologicznych i historycznych aluzji. Czytelnik jest jednak zaznajamiany z historią i instrumentami giełdowymi i niektóre z zasad inwestycji przedstawionych w książce stanowi bazę dla dzisiejszych rozwiązań.

Sposób wyboru akcji, których kupno będzie najbardziej opłacalne, nie jest prostym zadaniem. Podjęcie tej decyzji zawsze jest obarczone ryzykiem, ponieważ dokładne przewidzenie cen nie jest możliwe. Liczba czynników, które na nią wpływają jest bardzo

duża, nie sposób nawet przewidzieć ich wszystkich. Dodatkowo wiele informacji trudno ocenić. Policzalne aspekty spółki, takie jak przychody, są łatwe do znalezienia. Jednak jak je ocenić w kontekście kadry zarządzającej spółką lub jej reputacji? Kombinacja trudnych w ocenie czynników czyni wybór akcji subiektywnym a nawet intuicyjnym procesem [29].

Istnieją oczywiście pewne techniki, którymi kierują się inwestorzy, a mianowicie analiza fundamentalna i analiza techniczna. Analiza fundamentalna opiera się na badaniu kondycji ekonomicznej spółki. W ocenie brane jest wiele czynników, takich jak przychody, potencjalne zyski. Analizowane jest też otoczenie spółki, jak i sektor w którym działa. Analiza fundamentalna wymaga jednak sporo wiedzy żeby wprowadzić ją w życie, szczególnie w porównaniu z analizą techniczną.

Można powiedzieć że analiza techniczna jest przeciwieństwem analizy fundamentalnej. Podczas gdy ta druga ocenia spółki na podstawie ekonomicznych danych, które obiektywnie mogą wpływać na przyszłe ceny akcji, ta pierwsza ogranicza się do analizy przeszłych cen i wolumenu akcji. Za ojca analizy technicznej uważa się Johna J. Murphiego. W swojej książce "Technical Analysis of the Futures Markets", rozszerzonej następnie w "Technical Analysis of the Financial Markets" [17] opisał jej działanie i wykorzystywane narzędzia. Głównym założeniem jest stwierdzenie, że wszystkie ważne informacje szybko są odzwierciedlane w cenie akcji. Analiza techniczna opiera się na następujących podstawach:

1. Ceny odzwierciedlają istotne informacje. Innymi słowy, rynek jest wydajny.
2. Ceny poruszają się w trendach.
3. Historia się powtarza.

W praktyce wykorzystywane są różne wskaźniki. Stworzone aby pomóc w identyfikacji stanów rynku i generowania odpowiednich sygnałów - kiedy jest najlepszy czas na inwestycje. Przykładem takiego wskaźnika jest średnia krocząca, czyli średnia z pewnej liczby ostatnich cen danego aktywa. W tradycyjnej analizie technicznej wykorzystywana do wygładzenia wykresu cen, często wykorzystywana w parach z różnym okresem, generując sygnały kupna lub sprzedaży podczas ich przecięcia.

Przez wiele lat giełdy papierów wartościowych były fizycznymi miejscami gdzie kupcy i sprzedawcy spotykali się i negocjowali [30]. Spokojne transakcje, oko w oko w kawiarniach wyewoluowały w metodę "open outcry". Uczestnicy giełdy zbierali się na specjalnych arenach i krzykiem oraz specjalnymi gestami oznaczającymi parametry transakcji porozumiewali się z kontrahentami. Postępująca komputeryzacja praktycznie wyeliminowała tę formę handlu. W dzisiejszych czasach praktycznie wszystkie transakcje wykonuje się elektronicznie. Wykorzystanie komputerów dało zupełnie nowe możliwości wyboru inwestycji.

Nie dość, że wszystkie transakcje są wykonywane przy użyciu komputerów, to jeszcze znaczna ich liczba jest wykonywana jedynie przez komputery. Za pomocą handlu algorytmicznego, wykonywana jest większość transakcji w Stanach Zjednoczonych [25]. Szczególnym elementem handlu algorytmicznego jest handel wysokich częstotliwości. Opiera się na wykonywaniu bardzo szybkich i krótkich transakcji, w związku z czym jest zdominowany przez duże banki inwestycyjne i fundusze hedgingowe. Tylko one mogą pozwolić sobie na utrzymanie potrzebnej infrastruktury. Do wyboru transakcji wykorzystywane są skomplikowane algorytmy trzymane w tajemnicy. Opierają się one

między innymi na zjawisku arbitrażu. Polega on na zakupie aktywa, które na jednym z rynków jest niedocenione, celem sprzedaży go na innych rynkach. Instytucje korzystające z tej metody nie konkurują jednak ze zwykłymi użytkownikami. Zwykli inwestorzy, jeśli decydują się na handel automatyczny, wykorzystują zwykle różne wskaźniki analizy technicznej, wybrane własnoręcznie lub przez algorytmy sztucznej inteligencji. Wybór analizy technicznej przy automatycznym układaniu strategii jest podyktowany głównie faktem, że wszystkie potrzebne informacje są dostępne od ręki, a jedynym problemem jest ich przetworzenie.

Automatyczne systemy inwestycyjne mają wiele zalet ponad ręcznym inwestowaniem:

1. Eliminacja emocji - podczas inwestycji ludzie często kierują się emocjami. Ryzykują w końcu własne pieniądze. Zaprogramowana strategia może im pomóc podjąć decyzję, przełamać wahanie. Z drugiej strony ograniczają też inwestora, który jest skory do inwestycji przy pierwszej zauważonej okazji.
2. Możliwość backtestingu - backtesting to testowanie strategii na danych historycznych. Można w ten sposób oszacować skuteczność algorytmu, albo dobrać jego lepsze parametry.
3. Osiągnięcie trwałej skuteczności - Niemożliwe jest skonstruowanie strategii, która będzie za każdym razem dochodowa. Teoretycznie jednak automatyczny system inwestycyjny osiągnie stały procent korzystnych inwestycji, umożliwiając dokładniejsze zarządzanie budżetem.
4. Poprawiony czas wejścia w transakcję - komputer może zareagować w tym samym momencie, jak na rynku pojawi się konkretna sytuacja, co jest niemożliwe dla człowieka

Do wad należy podatność na mechaniczne usterki i często konieczność posiadanie umiejętności programowania w którymś z języków używanych w platformach inwestycyjnych. Co prawda istnieją programy, które umożliwiają ułożenie strategii metodą złap i przeciągnij, jednak w ten sposób działa tylko dla najprostszych strategii.

Innym segmentem rynków finansowych jest rynek walutowy, czyli Forex (Foreign Exchange). Umożliwia on handel walutami i jest największym i najbardziej płynnym rynkiem na świecie [12]. Handluje się na nim wszystkimi walutami świata i dzienny obrót może przekraczać bilion dolarów. Nie dość, że możliwe jest na nim korzystanie z analizy technicznej, to posiada wiele zalet, szczególnie dla handlu automatycznego. Po pierwsze, ze względu na to, że jest to największy rynek na świecie, jest on bardzo płynny, więc możemy praktycznie w każdym momencie kupić lub sprzedać nasze aktywa po aktualnej cenie. Po drugie, jest on otwarty przez prawie cały tydzień. Zamykany jest jedynie na weekend i święta. Ogranicza to ryzyko dużych zmian cen pomiędzy zamknięciem i otwarciem handlu. Dodatkowo program inwestycyjny może działać przez większą część roku - giełdy papierów wartościowych. Dla porównania giełda nowojorska jest otwarta tylko sześć i pół godziny dziennie, od 9.30 do 16.00. Po trzecie, ceny transakcji są bardzo niskie, a stanowiły one duży problem w niektórych pracach w których opisane były systemy pracujące na standardowych rynkach. Kolejnym atutem, nie wykorzystywanym jednak w tej pracy jest dostęp do dużej dźwigni, dzięki czemu można osiągnąć duże zyski przy małym wkładzie własnym.



## 1.4. Pojęcia wykorzystywane w inwestycjach

Waluty na Forexie są grupowane w pary walutowe. Określa ona stosunek ceny pierwszej waluty z pary do drugiej. Pierwsza waluta nazywana jest *bazową*, a druga *kwotowaną*. Para może być ona jednak uważana za pojedynczą jednostkę, która może być kupowana i sprzedawana. Kupno danej pary oznacza kupno waluty bazowej i sprzedaż kwotowanej. Pomiędzy cenami kupna i sprzedaży jest różnica, nazywana *spreadem*. Cena którą trzeba zapłacić za kupno to *bid*, a za sprzedaż to *ask*. Spread jest więc ceną, jaką trzeba zapłacić za przeprowadzenie transakcji.

Na giełdzie papierów wartościowych oprócz kupna akcji można również pożyczyć akcje, celem ich natychmiastowej sprzedaży i odkupienia po lepszej, niższej cenie. Rozpoczęcie takiej operacji nazywane jest *krótką sprzedażą* i oznacza otwarcie *krótkiej pozycji*. Analogicznie, klasyczna transakcja kupna akcji celem ich późniejszej sprzedaży nazywana jest otwarciem *pozycji długiej*. Mechanizmy działania tych transakcji są ukryte przed inwestorem, ważny jest fakt, że na pozycji krótkiej zarabia się gdy cena akcji spada, a na pozycji długiej gdy cena rośnie. Pojęcia nawiązują do przekonania, że w dłuższej perspektywie czasu cena większości akcji rośnie, ze względu na rozwój spółek. Na tej obserwacji opiera się strategia buy-and-hold, czyli kup i trzymaj wykorzystywana często jako punkt odniesienia w automatycznym układaniu strategii. Polega ona na otwarciu długiej pozycji i trzymaniu jej, nie przejmując się chwilowymi spadkami cen.

Mimo że mechanizmy transakcji na Forexie są inne niż na giełdzie papierów wartościowych, wspomniane pojęcia są wykorzystywane w takim samym kontekście. Wiele badań na Forexie przeprowadzana jest na parze EUR/USD. Zwana eurodolarem, jest najbardziej płynną parą na rynku, co oznacza że zawsze można znaleźć kogoś, kto chciałby ją sprzedać po aktualnej cenie. Dodatkową zaletą tej płynności jest najmniejszy spread spośród wszystkich par.

Historyczne dane z wielu rynków są ogólnie dostępne w internecie. Mimo, że ceny walut na rynku Forex zmieniają się ciągle, niepraktyczne byłoby analizowanie każdej transakcji. Rozważa się więc ceny o różnych interwałach - np. godzinnych. Dane są wtedy dzielone na godzinne okresy, w których zawarta jest cena otwarcia (ta, z którą rozpoczął się okres), zamknięcia, minimalna i maksymalna. Często dostępny jest także *wolumen*, czyli liczba aktywów które zmieniły właścicieli w danym okresie. Może ona określać stopień zainteresowania inwestorów.



## Chapter 2

# Sztuczna inteligencja w inwestycjach giełdowych

### 2.1. Różne podejścia w literaturze

Kompletny i aktualny przegląd metod inteligencji obliczeniowej wykorzystywanej w rynkach finansowych jest dostępny w pracy [6] z 2015 roku. Zebranych jest tam wiele metod wykorzystujących głównie analizę techniczną, ale także fundamentalną. Wykorzystanie analizy technicznej sprowadza się do przetwarzania szeregów czasowych cen, czyli sekwencji cen pobranych w określonych, stałych odstępach czasu. Ze względu na to, że takie szeregi są złożone, zaszumione, dynamiczne, nieliniowe, nieparametryczne i chaotyczne [26], tradycyjne metody statystyczne nie sprawiają się zbyt dobrze, ustępując pola sztucznej inteligencji. Wykorzystywane są głównie sieci neuronowe i metaheurystyki.

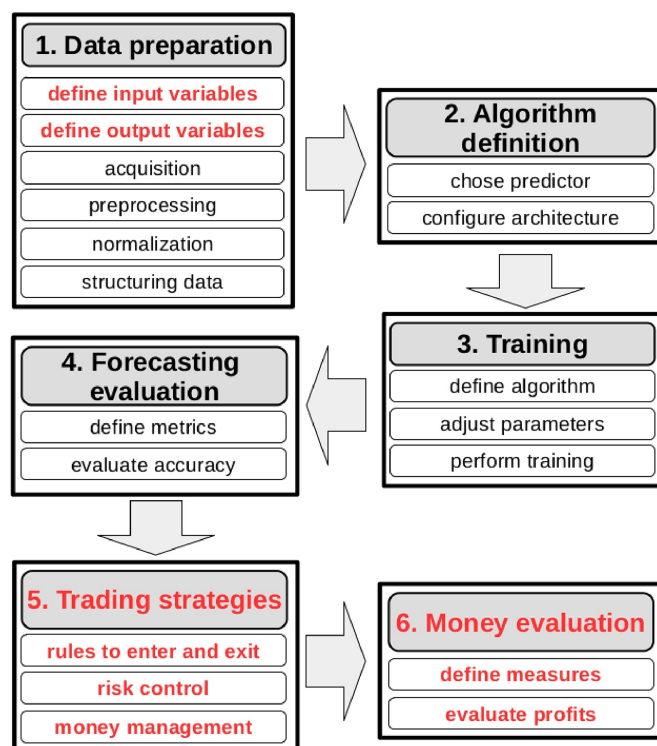
Autorzy wspominają, że nie istnieje ustalona metodologia tworzenia systemów inwestycyjnych. Mimo że prowadzone są nad tym badania, wyniki nie przedostają się do publiczności. Jest to spowodowane faktem, że jeśli komuś uda się stworzyć dobrze działający system, udostępnianie go publiczności oznaczałoby potencjalne ograniczenie własnych zysków. Celem wspomaganie dalszego rozwoju, zostało wyodrębnione sześć kroków, która każda praca, opisująca system inwestycyjny powinna zawierać, opisane na rysunku 2.1.

Czarne elementy można zastosować do konwencjonalnego problemu przewidywania, podczas gdy czerwone odnoszą się szczególnie do wykorzystania w rynkach finansowych. Pierwszy krok obejmuje wybór danych wejściowych i wyjściowych, pozyskanie ich i pre-processing. W przypadku analizy technicznej są to zwykle ceny lub wolumen akcji oraz różne wskaźniki techniczne, opisujące przeszłe zmiany cen. Ten krok ujawnia problem wielu metod wykorzystujących sieci neuronowe. W wielu pracach, daną wyjściową jest szansa na wzrost lub spadek ceny w kolejnym dniu. Jednak to w większości przypadków nie wystarczy do podjęcia decyzji o inwestycji.

W kolejnym drugim, trzecim i czwartym kroku wybierany jest algorytm przewidywania, uczony jest on na wcześniej wybranych danych i oceniana jest jego skuteczność za pomocą klasycznych miar, takich jak średni błąd bezwzględny.

Kolejne, ostatnie dwa kroki odnoszą się do wykorzystania nauczonego już modelu do inwestycji. Konieczne jest takie wykorzystanie danych wyjściowych, aby w dobrym mo-

Figure 2.1: Proponowane kroki rozwoju systemu inwestycyjnego



mencie kupić odpowiednie aktywa. Ważny jest mechanizm kontroli ryzyka, czyli zestaw zasad mających chronić zainwestowane pieniądze przed nagłymi stratami. Określa on, kiedy opuścić pozycję, która nie zachowuje się zgodnie z przewidywaniami. Konieczne jest także system zarządzania pieniędzmi, określający ile pieniędzy zainwestować mając pod uwagę ryzyko. Na samym końcu strategia powinna być oceniana używając backtestingu.

Standardowym podejściem w przypadku korzystania z sieci neuronowych jest zastosowanie prostej sieci MLP, wykorzystując przedtem jakąś innowacyjną metodę przetwarzanie danych wejściowych. Proste podejście zostało opisane w pracy Dhara, Mukherjee i Ghoshala [10]. Sieć MLP z jedną warstwą ukrytą została zastosowana do przewidywania ceny akcji na rynku indyjskim. Na wejściu sieć przyjmuje historyczne ceny otwarcia i zamknięcia oraz zwraca cenę zamknięcia następnego dnia. Sieć uczona została za pomocą algorytmu propagacji wstecznej. Przetestowane zostały różne rozmiary warstwy ukrytej i współczynniki uczenia, osiągając maksymalną skuteczność przewidywania na poziomie około 70%.

W innej pracy, Wang i Wang wykorzystują stochastic time strength neural network (STNN) i analizę głównych składowych. Jako dane wejściowe przyjęto dzienną cenę otwarcia, zamknięcia, minimalną i maksymalną, wolumen i dzienny obrót. Tych sześć parametrów zostało następnie przetworzone przez analizę głównych składowych, co pozwoliło zmniejszyć liczbę danych wejściowych tylko do dwóch. STNN bazuje na zwykłej MLP integrującej dodatkowo stochastyczną funkcję czasu, nadającą mniejsze wagi danym pojawiającym się dalej od teraźniejszości.

Niektóre podejścia wykorzystują także założenia analizy fundamentalnej. Nassirtoussi, Aghabozorgi, Wah i Ngo w 2015 roku [20] wykorzystują eksplorację tekstu

w nagłówkach wiadomości finansowych do przewidywania zmian cen walut na rynku Forex. Przeprowadzona jest semantyczna analiza tekstu, oraz analiza sentymentu, aby ocenić nastawienie inwestorów do wiadomości. Ważna również okazała się redukcja wymiarowości otrzymanych danych.

Autorzy przeglądu [6] twierdzą, że duży potencjał, choć jeszcze niedokładnie zbadany, ma wykorzystanie głębokich sieci neuronowych. Pierwsze prace wykorzystują DBL, uzyskując obiecujące wyniki. Mało czasu jest natomiast poświęcone metaheurystykom, które wykorzystywane są w trochę inny sposób.

## 2.2. Metaheurystyki w budowaniu strategii inwestycyjnej

Do układania strategii inwestycyjnej może być wykorzystany klasyczny algorytm ewolucyjny. L. Mendes, P. Godingo i J. Dias wykorzystują go w swojej pracy [16]. Mając wykształcenie ekonomiczne, skonstruowali wiele reguł, pomagając w podjęciu decyzji o inwestycji. Zmienne tych reguł, osobnych dla sygnałów kupna i sprzedaży, zostały umieszczone w wektorze, który reprezentował osobnika. Jako funkcje oceny wykorzystano *współczynnik Sterlinga*. Jest to miara zwrotu z inwestycji skorygowana o ryzyko. Istnieją różne definicje tego wskaźnika ale wykorzystany został wzór:

$$\text{współczynnik Sterlinga} = \frac{\text{zysk}}{MDD}$$

Gdzie MDD (Maximum Drawdown) to maksymalna osiągnięta strata w danym czasie. Liczona jako:

$$MDD = \frac{\text{maksymalna osiągnięta wartość portfela} - \text{minimalna osiągnięta wartość portfela}}{\text{maksymalna osiągnięta wartość portfela}}$$

Autorzy testowali swoje strategie na Forexie. Ograniczenie się do otwierania tylko długich pozycji na tym rynku byłoby zbyt dużym uproszczeniem. Forex wprowadza dodatkowo kolejny problem - brak jest strategii, która stanowiła by punkt odniesienia, tak jak buy-and hold w przypadku handlu akcjami. Na Forexie nie można liczyć na stały trend w którymś kierunku. Autorzy porównywali więc swoje strategie do sytuacji, gdy zysk jest zerowy. Testowano dane o różnym interwale: minutowe, pięciominutowe, piętnastominutowe oraz godzinne. Każdy interwał miał przydzielony inny okres treningowy, od miesiąca przy interwale minutowym do całego roku przy interwale godzinnym. Do każdego interwału zostały też wybrane okresy testowe o odpowiednich okresach. Brano pod uwagę parę EUR/USD i GBP/USD, jako największe, najbardziej płynne i o najmniejszym spreadzie. Osiągnięte wyniki na zbiorach treningowych były zdecydowanie pozytywne, jednak na zbiorze testowym jedynie 45% było zyskowych na EUR/USD

Russo Vincenzo w swojej pracy magisterskiej [32] wykorzystał algorytm świetlikowy. Jest to metaheurystyka rojowa, która symuluje zachowanie robaczek świętojańskich. Optymalne wartości są znajdowane biorąc pod uwagę położenie innych świetlików, które świecą intensywniej będąc w pobliżu lepszego rozwiązania. W jego podejściu metaheurystyka miała mniejszy wpływ na strategię niż we wcześniej opisanych podejściach. Wybierała jedynie wagi pomiędzy kilkoma, wcześniej ustalonymi wskaźnikami analizy technicznej. Miała za zadanie zoptymalizować stworzony system inwestycyjny,

a nie tworzyć go od podstaw. Największą zaletą takiego podejścia okazała się szybkość działania, która umożliwiała optymalizację strategii w ciągu kilku sekund, otrzymując zysku na zbiorze testowym przewyższające buy-and-hold. Sporym problemem okazały się koszty transakcji, które znacznie ograniczały potencjalne zyski.

## 2.3. Programowanie genetyczne w budowaniu strategii inwestycyjnej

Wiele prac do układania strategii inwestycyjnej na różnych rynkach wykorzystuje programowanie genetyczne. Ma ono tę zaletę, że nie wymaga wcześniejszej wiedzy ekonomicznej czy starannego wyboru wskaźników ekonomicznych. Te zależności są wydobywane przez sam algorytm.

Jedną z pierwszych na ten temat jest praca F. Allena i R. Karjalainena z 1999 roku [3]. We wszystkich przeszłych pracach strategię czy reguły inwestycyjne były tworzone ad hoc, co stwarzało ryzyko nadużywania danych - kierowania się nadmierną motywacją do uzyskania pozytywnego wyniku. Strategie były tworzone poprzez dopasowanie ich do posiadanych danych. Aby uniknąć tego problemu, autorzy zaproponowali użycie programowania genetycznego do układania strategii, tak aby reguły były tworzone przed użyciem danych testowych. Badania prowadzone były na indeksie S&P 500, do którego należą 500 największych amerykańskich spółek.

Każdy osobnik reprezentował funkcję zawartą w drzewie, z których każda reprezentowała osobną strategię. Drzewo jest budowane z zestawu bazowych funkcji rzeczywistych i logicznych. Każdy liść zawiera funkcję rzeczywistą, podczas gdy jego korzeniem jest funkcja logiczna. Funkcje logiczne obejmują operatory logiczne i porównania dwóch funkcji rzeczywistych. Funkcje rzeczywiste zawierają średnią kroczącą, której okres jest parametrem, oraz funkcje zwracające minimum i maksimum lokalne cen w zadanym okresie. Oprócz tego występują operatory arytmetyczne, oraz funkcja zwracająca aktualną cenę. Dodatkowo występują wartości stałe (true i false) oraz stałe wartości liczbowe. Stałe wartości logiczne są losowane na początku trenowania, tak sam jak wartości liczbowe. Podczas kolejnych generacji te wartości nie są losowane ponownie.

Ten zestaw funkcji bazowych może reprezentować strategię powszechnie wykorzystywaną w inwestycjach. Przykład takich strategii jest przedstawiony na rysunku ???. Przy wyborze takich funkcji bazowych autorzy dodatkowo kierowali się na innych pracach, z których wynikało, że wiele skomplikowanych reguł inwestycyjnych można przedstawić korzystając z lokalnych ekstremów przeszłych cen. Średnie kroczące są wykorzystane w modelowaniu istniejących trendów cenowych. Unikano bardziej skomplikowanych reguł, przedstawianych w poprzednich badaniach, aby uniknąć ich zbytniego dopasowania do danych.

Zastosowano ciekawy sposób normalizacji danych - cena w każdym analizowanym dniu została podzielona przez 250-dniową średnią kroczącą, otrzymując ceny w okolicach jedynki.

Jako funkcję przystosowania użyto zysku strategii ponad strategią buy-and-hold. Ze względu na to, że nie rozważano otwierania pozycji krótkich, taki wybór jest oczywisty.

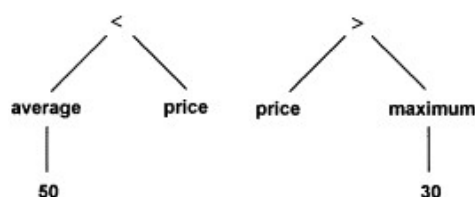


Figure 2.2: Osobniki programowania genetycznego reprezentujące strategie. Lewa generuje sygnał kupna, gdy średnia krocząca z 50 ostatnich cen jest mniejsza niż aktualna cena, sygnał sprzedaży w przeciwnym wypadku.

Wspomniano jednak o możliwości dodatkowego karania dużych spadków posiadanego kapitału, w ten sposób biorąc pod uwagę ryzyko inwestycji.

Aby właściwie przetestować zyskowność strategii, wprowadzono zbiór danych podzielono na treningowy i walidacyjny. Wszystkie operacje genetyczne było przeprowadzane na treningowym, a co generację sprawdzano, czy najlepsza otrzymana strategia jest lepsza od poprzedniej na zbiorze walidacyjnym. Tylko wtedy była ona zapamiętywana zastępując poprzednią najlepszą i ostatecznie wybierana jako najlepsza ze wszystkich generacji. Do krzyżowania wybierane były dwa osobniki, które tworzyły nowego ze swoich poddrzew. Mutacja została zaimplementowana poprzez wybór losowego drzewa jako kandydata do krzyżowania, z małym prawdopodobieństwem.

Otrzymane reguły inwestycyjne nie przynosiły stałych zysków w porównaniu do strategii buy-and-hold po odliczeniu kosztów transakcji. Mają jednak pewne zdolności przewidywania, bo opóźnienie transakcji o jeden dzień zdecydowanie pogarsza zyski.

Niedługo potem C. Nelly [21] przetestował techniki kontroli ryzyka zaproponowane w poprzedniej pracy. W podejściu zmieniony został tylko sposób oceny osobników. Zastosowano między innymi wskaźnik Sharpe'a. Opracowany przez laureata Nobla Williama Sharpe'a, jest powszechnie wykorzystywany jako miara potencjalnego zysku w stosunku do poniesionego ryzyka. Im współczynnik jest wyższy, tym inwestycja jest bardziej atrakcyjna.

$$S = \frac{r_p - r_f}{\sigma_p}$$

Gdzie:

$S$ : wskaźnik Sharpe'a

$r_p$ : średnia stopa zwrotu z inwestycji

$r_f$ : średnia stopa zwrotu wolna od ryzyka

$\sigma_p$ : odchylenie standardowe inwestycji

Inwestycją wolną od ryzyka są bony skarbowe, krótkoterminowe papiery dłużne emitowane przez rząd.

Wygenerowano strategie wykorzystując model Allena i Karjalainena z różnymi funkcjami oceny. Po zmianie funkcji oceny na wskaźnik Sharpe'a, osiągnięto trochę lepsze wyniki niż przy wcześniej zaproponowanej ocenie, jednak wciąż gorze niż strategia buy-and-hold. Powstałe strategie wykonywało mało inwestycji, upodabniając i zbliżając się do buy-and-hold, jednak nie udało się jej prześcignąć. Podobne wyniki otrzy-

mano wykorzystując bardziej skomplikowane funkcje oceny zaczerpnięte z literatury ekonomicznej.

W pracy L. Beckera i M. Seshardi [5] udało się w końcu prześcignąć buy-and-hold. Po pierwsze, obliczenia przeprowadzano na danych miesięcznych zamiast dziennych. Celem zmiany było zredukowanie liczby inwestycji i co za tym idzie potencjalnych kosztów transakcji. Następnie zdecydowanie uproszczono reprezentację strategii. W węzłach nieterminalnych drzewa mogły pojawiać się tylko operatory logiczne (and, or, not) oraz porównania ( $>$  i  $<$ ). W liściach mogło znaleźć się kilka różnych wskaźników technicznych:

- Ceny: otwarcia, zamknięcia, maksymalna i minimalna w miesiącu
- Średnie kroczące: 2, 3, 4 i 10 miesięcy
- Rate of Change: 3 i 12 miesięcy
- Wskaźniki oporu ceny: 2 ostatnie minima i maksima 3-miesięcznej średniej kroczącej
- Wskaźniki trendu: dolna linia wyznaczona na podstawie 2 ostatnich minimów 3-miesięcznej średniej kroczącej oraz górna linia wyznaczona na podstawie 2 ostatnich maksimów 3-miesięcznej średniej kroczącej

Pierwszy eksperyment przeprowadzono z modyfikując funkcję oceny, aby oprócz zysku brała pod uwagę skomplikowanie drzewa. Większe drzewa były obarczone karą. To podejście okazało się znacznie polepszać generalizację wyników. Podczas gdy wielkość drzewa nie była ograniczana, powstałe strategie miały po kilkadziesiąt elementów. Były one bardzo trudne do zrozumienia przez człowieka. Drzewa ułożone wykorzystując zmienioną funkcję oceny, miały maksymalnie kilkanaście elementów i dodatkowo miały zdecydowanie lepsze wyniki na zbiorze testowym. Zyski na zbiorze treningowym były mniejsze, co oznaczało, że większe drzewa miały tendencję do nadmiernego dopasowania do danych uczących.

W kolejnym eksperymencie znowu zmodyfikowano funkcję oceny. Tym razem zamiast zysku brana pod uwagę była liczba dochodowych okresów, modyfikowana również przez karę za wielkość drzewa. Średnie wyniki okazały się jeszcze lepsze niż w poprzednim eksperymencie, prawie w każdym przypadku lepsze od strategii buy-and-hold.

Kolejne polepszenie wyników uzyskano poprzez zastosowanie ewolucji kooperatywnej. W populacji istniały dwa drzewa - jedno oznaczające sygnał kupna, drugie sygnał sprzedaży. Podczas gdy oba ewoluowały w tym samym czasie w jednej populacji, każde z nich było uważane za osobny gatunek i krzyżowało się tylko ze swoim odpowiednikiem. Wykorzystując to podejście, 99,5% otrzymanych strategii osiągało zyski większe niż buy-and-hold.

Zostały przeprowadzone dodatkowe badania nad ewolucją kooperatywną [4]. Najlepsze wyniki otrzymano, gdy zamiast dwóch gatunków osobników w populacji, każdy osobnik zawierał w sobie dwa drzewa każde z jedną regułą - jedną zakupu drugą sprzedaży. W tym przypadku ewaluacja osobnika jest reprezentowana przez tabelę na rysunku 2.3. Puste pozycje oznaczają, że dana reguła nie jest brana pod uwagę w danym przypadku.

Ważny jest fakt, że zwiększenie kary za rozmiar drzewa gdy zwiększa się ich liczbę w osobnikach poprawia wyniki na zbiorze testowym. Oznacza to, że konieczne jest stałe kontrolowanie wielkości drzew, które łatwo podlegają nadmiernemu dopasowaniu do danych treningowych.



Figure 2.3: Podejmowanie decyzji przy osobniku reprezentowanym przez dwie reguły

2-Rule Case	Buy Rule	Sell Rule	Action
In Market		T	Sell
In Market		F	<i>Do Nothing</i>
Out of market	T		Buy
Out of market	F		<i>Do Nothing</i>

Wspomniane prace Beckera i Seshardieg nie były jednak napisane na tyle dokładnie, aby na ich podstawie odtworzyć zaprezentowane modele. Na tym problemie skupili się D. Lohpetch i D. Corne [14]. Oprócz dokładnego opisanie wszystkich operatorów programowania genetycznego, zaproponowano nowy sposób wyboru najlepszej strategii z populacji. Podstawowe podejście polegała po prostu na wybraniu tej, która po określonej liczbie generacji miała najlepsze wyniki i testowaniu jej na zbiorze testowym. Zaproponowany sposób, mimo że autorzy tak tego nie nazwali, wprowadzał kolejny zbiór - walidacyjny, rozłączny z pozostałym treningowym i testowym. Po skończeniu ewolucji na zbiorze treningowym, najpierw ewaluowane były wszystkie strategie z finalnej populacji na zbiorze walidacyjnym i z niej wybierana była najlepsza, gotowa do następującego po nim zbioru testowego. To podejście dawało lepsze i bardziej stałe wyniki, poprzez ograniczenie dopasowania do danych treningowych.

Badania nad walidacją wyników były też przeprowadzone wcześniej przez J. Thomasa i K. Sycarę [31] na Forexie. W badanym podejściu dane również były podzielone na 3 zbiory: treningowy, walidacyjny i testowy. Przetestowane były dwa podejścia: jedno na którym finalna populacja była testowana na zbiorze walidacyjnym i wybierana była jedna najlepsza strategia, co odpowiadało podejściu z poprzedniej pracy [14]. W drugim na zbiór testowy kwalifikowały się wszystkie strategie, przynoszące dochód ze zbioru walidacyjnego. Mimo podobnego podejścia, autorzy otrzymali zupełnie inne wyniki. Dodatkowa walidacja okazała się pogarszać działanie powstałych strategii w każdym badanym przypadku. Wyjaśnione było to brakiem korelacji pomiędzy zbiorem walidacyjnym i testowym - wybrana strategia ze zbioru walidacyjnego nie musiała najlepiej sprawować się na testowym. Innymi słowy, nie ma korelacji pomiędzy odległością od zbioru testowego a efektywnością strategii.

Natomiast różnice pomiędzy wynikami z obu prac trudno wyjaśnić po prostu różnicami w modelu. Prawdopodobna przyczyna leży jednak w doborze danych. Być może dane na Forexie aż tak różnią się od danych z indeksu S&P. Innym wytłumaczeniem jest dopasowanie do akurat wybranych danych. D. Lohpetch i D. Corn [14] analizowali jedynie jeden zestaw danych treningowych i następujące po nim trzy różne zestawy zbiorów walidacyjnych i testowych różniły się tylko okresem. Podobne podejście zastosowane było w J. Thomasa i K. Sycary [31], w którym zbiór treningowy był tylko jeden, z następującym bezpośrednio walidacyjnym i testowym.

Ciekawe podejście do wyboru danych treningowych i testowych zostało zaprezentowane w [19]. Autor przeprowadzając badania na warszawskiej Giełdzie Papierów Wartościowych, wybrał trzy stosunkowo krótkie zbiory treningowe. Każdy przejawiał inny trend: spadkowy, neutralny lub rosnący. W taki sam sposób wybrane zostały ciągi testowe, po trzy dla każdego trendu. Podejście to korzysta z wniosków otrzymanych w

poprzednio opisanej pracy, że zbiory testowe nie muszą występować po treningowych. Otrzymane wyniki sugerowały, że strategie uczone na zbiorach spadkowych, osiągały mniejsze straty w podobnych zbiorach testowych, jednocześnie osiągając mniejsze przychody w trendach wzrostowych. Dużym problemem było duże odchylenie standardowe strategii otrzymanych po uczeniu. Oznaczało to, że algorytm miał tendencję do utykania w lokalnych minimach.

Podobny sposób wyboru zbiorów danych wykorzystane było przez P. Myszkowskiego i A. Bicza w pracy wykorzystującej programowanie genetyczne do znajdowania strategii inwestycyjnej na Forexie [18]. Zbadane zostały różne kombinacje zbiorów treningowych i testowych o różnych trendach. Znowu ewidentna okazała się duża niestalość otrzymywanych strategii. Zaproponowane zostało podejście, które wykorzystywało tę różnorodność, tworząc jedną strategię z 200 innych. Transakcja była podejmowana tylko wtedy, gdy 60% wszystkich strategii ze zbioru zgadzało się co do tego. Takie rozwiązanie było bardzo skuteczne, w każdym przypadku poprawiając średni wynik z otrzymanych strategii na danych zbiorach.

## 2.4. Podsumowanie

Wśród prac wykorzystujących programowanie genetyczne można rozróżnić dwa podejścia. Pierwsze jest skupione na ewolucji podejścia przedstawionego przez F. Allena i R. Karjalainena [3]. Kolejne prace skupiają się głównie na uproszczeniu reprezentacji strategii, a także testowaniu potencjalnych ulepszeń. Dostępne są prace wyjaśniające niedokładnie opisane elementy algorytmu, stanowiące pewnego rodzaju poradnik do implementacji [14]. Jednak nawet we wspomnianej pracy brakuje niektórych detali niezbędnych w odtworzeniu modelu.

Kolejne podejście, reprezentowane przez prace P. Myszkowskiego i współautorów [19][18], różni się głównie reprezentacją liści drzewa. Podczas gdy w pierwszym rozwiązaniu, w liściu znajduje się wskaźnik techniczny porównany do innego wskaźnika technicznego (z których oba wcześniej ustalone), to w drugim podejściu wskaźnik techniczny jest porównany do wylosowanej liczby rzeczywistej. To rozwiązanie daje więcej wolności algorytmowi genetycznemu, poszerzając przestrzeń rozwiązań. Większa przestrzeń rozwiązań oznacza jednak potencjalnie trudniejsze strojenie algorytmu, a że oba podejścia umożliwiają otrzymanie pozytywnych wyników (dokładne porównanie nie jest możliwe, ze względu na to że różne prace korzystają z różnych zbiorów danych, mają też inne miary sukcesu) w zastosowanym podejściu skupiono się bardziej na podejściu pierwszym. Dodatkowo, pierwsze podejście jest bardziej intuicyjne, jako że lepiej odzwierciedla sposób inwestowania ludzi. Podstawowym sygnałem wykorzystywanym przez inwestorów jest przecięcie średniej kroczącej czy to przez średnią kroczącą o innym okresie czy przez aktualną cenę. Podczas gdy jest to proste do przedstawienia w pierwszym podejściu, porównanie wartości średniej kroczącej do pewnej losowej wartości liczbowej nie daje takich możliwości.

## Chapter 3

# Zastosowane podejście

Ze względu na niewystarczająco dokładne opisanie implementacji we wspomnianych pracach, nie było możliwe dokładne odtworzenie istniejących podejść. Zastosowany algorytm jest jednak silnie inspirowany istniejącymi rozwiązaniami. Główną inspiracją była praca D. Lohpetcha i D. Corna [14], które jest najnowszą i najdokładniejszą poświęconą reprezentacji zapoczątkowanej przez F. Allena i R. Karjalainena[3]. Zaczepnięte z niej zostały szczegóły implementacji algorytmu. Podczas testowania wzorowano się jednak bardziej na pracy A. Bicza i P. Myszkowskiego [18]. W dalszym opisie znajduje się wyjaśnienie opis każdego elementu implementacji z argumentacją użycia. Poszczególne podejścia będą referowane jako “AiK” (podejście F. Allena i R. Karjalainena[3]) oraz “BiM” (A. Bicz i P. Myszkowski [18]).

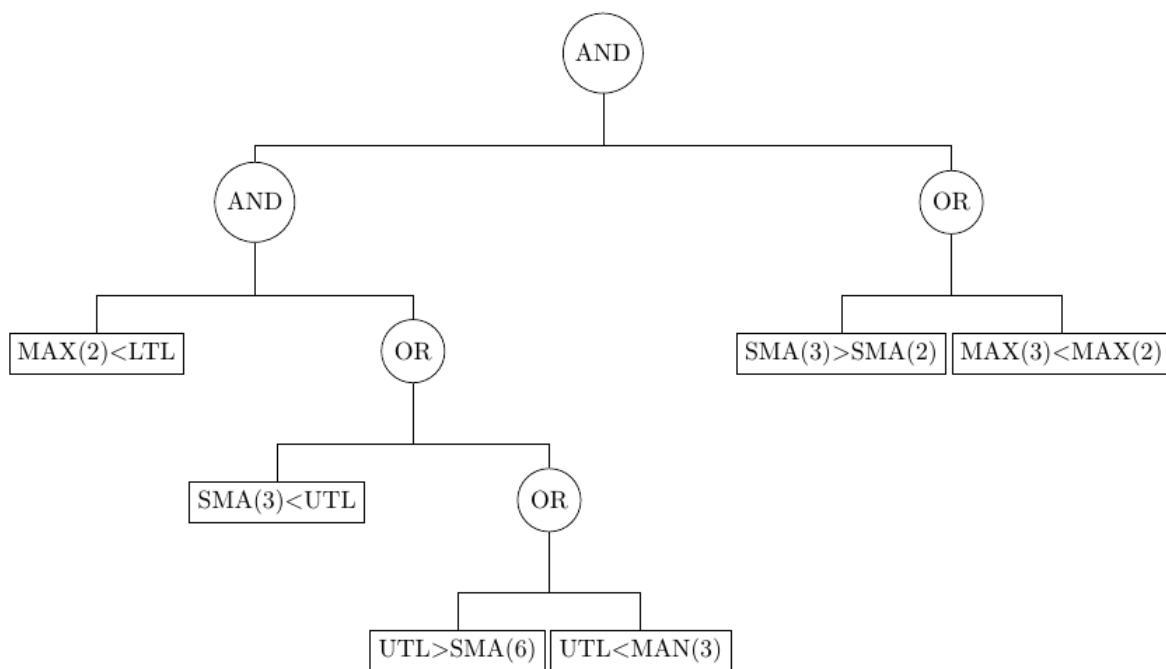
Przy konstrukcji algorytmu brano pod uwagę wyniki i problemy i wyniki opisanych podejść. Istotna też była spójność implementacji oraz logiczne i proste przedstawienie strategii.

### 3.1. Reprezentacja osobnika

Osobnik algorytmu genetycznego reprezentuje drzewo strategii. Przykład takiej strategii jest widoczny na rysunku 3.1. Zdecydowano się na ciągłe pozostawanie na rynku i strategię określa kiedy powinna być otwarta pozycja długa a kiedy krótka, w zależności od tego, czy drzewo zwraca wartość true czy false. Po początkowej inwestycji sprawdzane otwarta pozycja jest jedynie odwracana - z długiej na krótką i odwrotnie.

Węzły drzewa dzielą się na terminalne i nieterminalne. W węzłach nieterminalnych mogą znajdować się operatory logiczne OR oraz AND. Zrezygnowano z operatora NOT dlatego, że nie niesie ze sobą praktycznie żadnej informacji. Ze względu na to, że w węzłach terminalnych (liściach drzewa) znajdują się porównania, operator NOT występujący przed porównaniem jedynie odwracał operator porównania, co może być przedstawione w łatwiejszy sposób. Jak już zostało wspomniane, w liściach występuje porównanie wskaźników technicznych. Ich zestaw został zaczerpnięty z pracy AiK, jako że był wykorzystywany już w poprzednich pracach traktujących o tym podejściu i dawał pozytywne wyniki. Dodatkowo, wybór odpowiednich wskaźników technicznych wymaga wiedzy ekonomicznej, posiadanej przez autorów wybranego zestawu:

Figure 3.1: Przykładowa strategia wygenerowana przez algorytm



- Ceny: otwarcia, zamknięcia, maksymalna i minimalna w okresie
- Średnie kroczące (SMA): 2, 3, 4 i 10 ostatnich okresów
- Rate of Change (ROC): 3 i 12 ostatnich okresów
- Wskaźniki oporu ceny: 2 ostatnie minima (MAN) i maksima (MAX) średniej kroczącej z 3 ostatnich okresów
- Wskaźniki trendu: dolna linia wyznaczona na podstawie 2 ostatnich minimów średniej kroczącej z 3 ostatnich okresów (LTL) oraz górna linia wyznaczona na podstawie 2 ostatnich maksimów średniej kroczącej z 3 ostatnich okresów (UTL)

Adaptując zestaw z literatury została zmieniona jedynie długość okresu - ze względu na fakt, że transakcje na Forexie są obciążone mniejszym kosztem, można potencjalnie przeprowadzać ich więcej nie bojąc się o to, że cały zysk będzie przeznaczony na spłatę kosztów. Wybrano więc okresy 10-minutowe, chcąc wyuczyć system do podejmowania częstszych inwestycji.

Większość wykorzystywanych wskaźników jest prosta w implementacji. Średnia krocząca z  $x$  okresów jest obliczona jako zwykła średnia arytmetyczna z  $x$  ostatnich cen zamknięcia. Bardziej skomplikowany jest wskaźnik ROC:

$$ROC = \frac{P_a - P_b}{P_a}$$

Gdzie:

- $P_a$ : cena zamknięcia w okresie a
- $P_b$ : cena zamknięcia w okresie b

ROC opisuje zmianę ceny aktywa w czasie. Jest oscylatorem, co oznacza, że jest w praktyce wykorzystywany w inny sposób niż pozostałe wskaźniki. Podczas gdy wszystkie pozostałe mogą być nałożone na wykres ceny i generować sygnały przy porównaniu ze sobą nawzajem, to oscylator przyjmuje konkretne wartości - w przypadku ROC można je ograniczyć od -1 do 1, chociaż w praktyce przedział jest zwykle mniejszy. Teoria mówi, że gdy jego wartość jest duża, może to wskazywać że cena w krótkim okresie jest zbyt duża i sugerować zmianę trendu. W wykorzystywanej literaturze nie zostało wspomniane, czy ten wskaźnik jest traktowany szczególnie. W zaimplementowanym podejściu jednak jako jedyny nie jest porównywany do pozostałych - zamiast tego, jeśli wylosowany został wskaźnik ROC jako pierwszy z porównywanej pary z liścia, to zamiast kolejnego wskaźnika losowana jest liczba rzeczywista z zakresu -0.6 do 0.6.

### 3.2. Wykorzystywane dane i ocena osobnika

Figure 3.2: Wykorzystywane dane

Local time	Open	High	Low	Close	Volume
01.01.2007 00:00:00.000	1.31896	1.31921	1.31884	1.31906	7.28E+09
01.01.2007 00:10:00.000	1.31896	1.31921	1.31887	1.31904	8.03E+09
01.01.2007 00:20:00.000	1.31908	1.31924	1.31889	1.31907	7.32E+09
01.01.2007 00:30:00.000	1.31908	1.31924	1.3189	1.31901	8.56E+09
01.01.2007 00:40:00.000	1.31905	1.31922	1.31893	1.31902	7.88E+09
01.01.2007 00:50:00.000	1.31903	1.31924	1.31878	1.31913	7.9E+09
01.01.2007 01:00:00.000	1.31908	1.31963	1.31896	1.31945	9.29E+09
01.01.2007 01:10:00.000	1.31959	1.31964	1.31918	1.31929	7.34E+09
01.01.2007 01:20:00.000	1.31919	1.3194	1.31902	1.31913	8.21E+09

Wszystkie badanie przeprowadzono na Forexie, ze względu na wspomniane już zalety. Dodatkową motywacją było przetestowanie wariantu sprawdzonego już modelu na zupełnie innych danych. Dane w formacie .csv zostały pobrane z portalu Dukascopy [1]. Ich początek jest widoczny na rysunku 3.2. Można zauważyć, że w tabeli znajdują się ceny otwarcia, zamknięcia a także maksymalna i minimalna w danym okresie. W preprocessingu usunięto jedynie okresy, w których giełda była zamknięta. Normalizacja danych nie była potrzebna, bo wykorzystywane wskaźniki opierają się głównie na wzajemnym porównaniu.

Bardzo ważnym aspektem praktycznie nie poruszonym w literaturze jest sposób, w jaki otrzymywany jest zysk na historycznych danych. Ta wartość jest podstawą do właściwej oceny osobnika i jest niezbędna do zbudowania działającego algorytmu. W zastosowanej implementacji wzorowano się na sposobie testowania z programu MetaTrader 5 opisanym w artykule [8]. Metatrader jest uniwersalną platformą do handlu na różnych rynkach i udostępnia zaawansowane opcje konstruowania jak i testowania robotów handlowych - czyli programów podejmujących autonomiczne decyzje o transakcjach.

Podstawowym podejściem jest analiza każdej zmiany ceny, czyli tak zwanego *ticku*. Takie rozwiązanie wymaga jednak bardzo dużo obliczeń, ze względu na to, że cena

aktywa na forexie potrafi się zmieniać bardzo często. Metatrader wykorzystuje więc symulację ticków. Punktem wejścia jest zestaw cen o pewnym okresie, na przykład 10 minutowym. Następnie w kolejności analizowane są ceny otwarcia, najmniejsza, największa i ostatecznie cena zamknięcia. W każdym etapie przeprowadzana jest symulacja, sprawdzane jest, czy robot powinien otworzyć jakąś pozycję. W naszym przypadku sprawdzane jest czy strategia generuje sygnał odwrócenia pozycji. Takie podejście oferuje równowagę pomiędzy dokładnością, a wymaganą mocą obliczeniową.

Konieczna była jednak jedna modyfikacja. Jeśli cena największa była za każdym razem po najmniejszej, podczas uczenia strategia była dopasowywana do tego wzorca. Dlatego kolejność ceny najmniejszej i największej była za każdym razem losowana.

Zysk liczony jest w punktach zwanych pipsami częściowymi. Pips to określenie wykorzystywane na rynku Forex, określające czwartą cyfrę po przecinku. Historycznie była to najmniejsza jednostka zmiany notowań, po prostu mniejsze zmiany nie były zapisywane. Aktualnie w popularnych parach walutowych, także w EUR/USD stosuje się pipsy częściowe, odpowiadające 1/10 zwykłego pipsa. Przyjęty sposób liczenia zysku można więc łatwo zamienić na punkty procentowe. Mimo, że ta wartość nie jest od razu przekładalna na procentowy zysk zainwestowanego kapitału, który interesuje wszystkich inwestorów, to takie podejście zostało wykorzystane we wspomnianej pracy BiM. Umożliwia to więc porównanie wyników i jednocześnie jest proste w implementacji.

Ważne w symulacji jest także wzięcie pod uwagę kosztów transakcji. W wykorzystywanych danych nie jest zapisany aktualny spread, który zależy od wybranego pośrednika, dlatego przyjęto stałą wartość równą 40 pipsom częściowym. Również wykorzystywana w w pracy BiM, przewyższa jego aktualną wartość wynoszącą około 12 pipsów częściowych. W literaturze wspomniane zostało jednak, że wyższe koszty transakcji podczas uczenia mogą powodować tworzenie bardziej ostrożnych strategii.

W ocenie osobnika uwzględnia się także aktualną wysokość drzewa. Poprzednie badania twierdziły, że nagradzanie prostszych strategii, czyli takich reprezentowanych przez mniejsze drzewa poprawia ostateczne wyniki. Oczywiście jest wiele sposobów na branie pod uwagę tego czynnika. Zastosowano rozwiązanie z literatury:

$$P = \frac{H * Gain}{\max(H, H_r)}$$

Gdzie:

$P$ : ocena osobnika

$H$ : wysokość drzewa ustalona na początku działania algorytmu

$Gain$ : zysk w pipsach częściowych na zbiorze treningowym

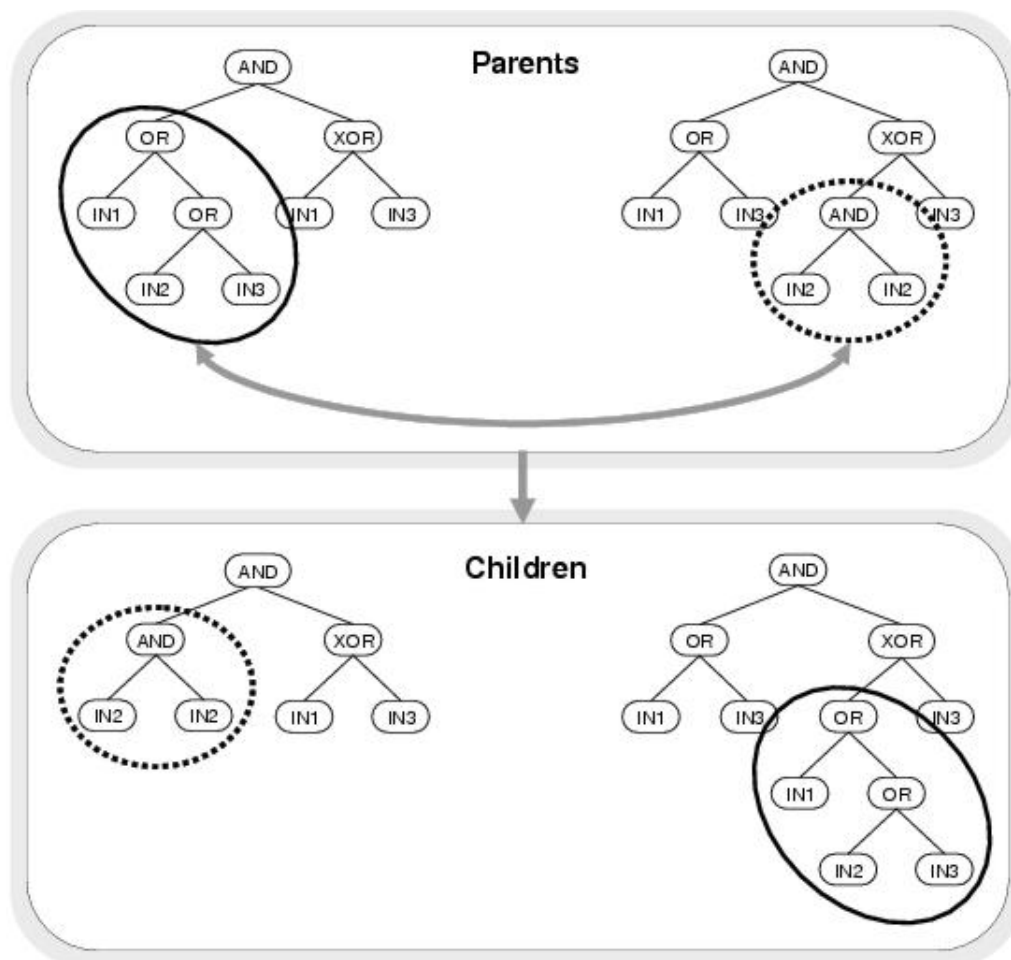
$H_r$ : aktualna wysokość drzewa strategii

### 3.3. Krzyżowanie

Do krzyżowania drzew wykorzystano metodę zaproponowaną przez J. Kożę, użytą także we wspomnianej już literaturze. Jej przykład można zaobserwować na rysunku 3.3. Krzyżowane są dwa osobniki, wybierane na zasadzie turnieju. Po każdym krzyżowaniu powstają dwa osobniki, będące rekombinacją swoich rodziców. Takie podejście jest podstawowe w programowaniu genetycznym. Punkt przecięcia wybierany jest w dwóch

krokach: w pierwszym wybierana jest losowa ścieżka w drzewie, a w drugim losowane jest miejsce przecięcia na tej ścieżce.

Figure 3.3: Przykład krzyżowania drzewa poprzez wymianę poddrzew [15]



### 3.4. Mutacja

Zaimplementowano 4 operatory mutacji:

- Mutacja liści: wybrany liść jest zamieniany na inny losowy liść
- Mutacja węzła wewnętrznego: wybrany węzeł nieterminalny jest zamieniany na odwrotny, tzn. AND na OR i odwrotnie
- Obcięcie drzewa: wybrany węzeł nieterminalny jest zamieniany na losowy liść
- Wzrost drzewa: wybrany liść jest zamieniany na losowe poddrzewo

Wykorzystanie krzyżowania i mutacji jest następujące: W każdej generacji, póki nie jest zapełniona populacja dzieci, wybierana są na zasadzie turnieju 2 osobniki, które następnie są krzyżowane z pewnym prawdopodobieństwem. Każde powstałe dziecko jest poddawane mutacji. Każdy operator mutacji działa niezależnie, z własną szansą na działanie na każdym węźle dziecka. Alternatywą użytą w AiK jest użycie jednego,

losowego sposobu mutacji na całym drzewie. Zastosowane podejście umożliwia jednak większą kontrolę nad ewolucją drzewa, będąc jednak trudniejszą w dostrojeniu.

### 3.5. Podsumowanie algorytmu

Początkowa populacja jest wypełniana całkowicie losowymi osobnikami. Określona jest maksymalna wysokość drzewa które reprezentują, jednak tylko w początkowej populacji. W kolejnych generacjach limit ten może być przekraczany w wyniku krzyżowania i mutacji. Nagradzane jest jednak nieprzekraczanie podanego limitu, o co dba funkcja oceny osobnika. Jest to podejście zastosowane przez AiK. Następnie wszystkie osobniki są oceniane wykorzystując backtesting - zwrócony zysk w pipsach częściowych jest wykorzystywany w funkcji oceny. Kolejna populacja jest wypełniana, poprzez wybieranie kandydatów za pomocą turnieju. Niektóre wybrane osobniki są krzyżowane, każdy wchodzący do nowej populacji ma też szansę na mutację. Gdy nowa populacja się zapełni, cały proces rozpoczyna się od nowa aż do osiągnięcia warunku stopu wyznaczonego przez z góry określoną liczbę generacji.

Całe uczenie przebiega na jednym zbiorze treningowym, najlepszy osobnik z ostatniej generacji jest wybierany do testowania na zbiorze testowym.

Istnieje więc 9 hiperparametrów wpływających na działanie algorytmu

- Rozmiar populacji
- Liczba generacji
- Maksymalna wysokość drzewa
- Wielkość turnieju
- Szansa na każdy z 4 operatorów mutacji
- Szansa na krzyżowanie



## Chapter 4

# Wpływ operatorów na działanie algorytmu

Aby zaproponować ewentualne metody poprawy działania algorytmu, konieczne jest przeprowadzenie badań ujawniających wpływ różnych operatorów na jego działanie. Częścią tego działania jest dobór hiperparametrów. Hiperparametry wpływają na efektywność algorytmu, dlatego konieczne jest ustalenie ich wartości tak, aby zapewnić działanie algorytmu bliskie optymalnemu. Parametry mogą przyjmować szeroki zakres wartości, a dodatkowo mogą wpływać na siebie nawzajem. W standardowym problemie, takim jak kolorowanie grafu, mamy jednak tę zaletę, że możemy w każdym momencie łatwo zweryfikować działanie algorytmu, uruchamiając go na różnych danych treningowych. W przypadku układania strategii inwestycyjnej, nawet idealne działanie na danych treningowych nie wystarczy. Jak udowadniają przeprowadzone testy, często średni zysk na danych testowych jest odwrotnie zależny od zysku na danych treningowych. Oznacza to, że gorsze dopasowanie do danych treningowych jest później nagradzane.

Testowanie działania na samych danych treningowych jest jednak również przydatne. Dzięki temu można ocenić działanie badanego operatora w trakcie ewolucji i jego wpływ na proces optymalizacji strategii.

Strategia tworzona z wykorzystaniem danych treningowych za każdym razem będzie trochę inna. Trudno jest więc ocenić, która będzie miała największe możliwości generalizacji. Konieczne jest zatem wielokrotne uruchamianie algorytmu i uśrednianie wyników, żeby na ich podstawie podjąć decyzję.

Przetestowano wpływ hiperparametrów na wyniki na zbiorach treningowych jak i testowych, w celu odkrycia wzorców, mogących posłużyć do ewentualnej poprawy działania algorytmu.

## 4.1. Wykorzystywane dane

Testując działanie algorytmu wykorzystano sześć szeregów czasowych. Każdy z nich obejmował około dwóch tygodni 4.1. Procedura testowania polegała na wyuczeniu strategii na jednym z szeregów (pierwszym, drugim lub trzecim) a następnie przetestowanie jej na 3 kolejnych szeregach. Oprócz uruchamiania algorytmu na różnych zbiorach, każde badanie powtarzano dwustukrotnie. Dopiero taka liczba pozwalała na otrzy-

manie wystarczająco stałych wyników. Otrzymane wyniki były uśredniane, aby podjąć decyzję o działaniu algorytmu.

## 4.2. Początkowe wyniki

Początkowe wyniki, z parametrami wybranymi intuicyjnie, są porównywalne do tych z literatury. Utworzone strategie dobrze dopasowują się do zbioru treningowego, otrzymując stały, dodatni wynik. Na zbiorze testowym ujawnia się jednak bardzo duże zróżnicowanie strategii. Wiele z nich jest stratnych, jednak średni zwrot jest pozytywny.

Do otrzymania początkowych wyników wykorzystano następujące wartości parametrów:

- Rozmiar populacji = 200
- Liczba generacji = 100
- Maksymalna wysokość drzewa = 5
- Wielkość turnieju = 10%
- Szansa na każdy z 4 operatorów mutacji = 5%
- Szansa na krzyżowanie = 20%

Table 4.1: Początkowe wyniki na szeregach treningowych

Zysk w pipsach	Odchylenie standardowe zysku
4258.9	705.8

Table 4.2: Początkowe wyniki na szeregach testowych

Zysk w pipsach	Odchylenie standardowe zysku
225.4	2067

## 4.3. Wysokość drzewa

Ustalona wysokość drzewa ma bardzo istotny wpływ na działanie algorytmu. Jest kluczowa w ograniczaniu dopasowania do danych treningowych. Badania potwierdzają, że ta cecha w największym stopniu wpływa na osiągnięte wyniki na zbiorze testowym.

### 4.3.1. Wysokość drzewa na danych treningowych

Wysokość drzewa ma ewidentny wpływ na wyniki na zbiorze treningowym. Na wykresach 4.2 można zauważyć, że średnia wysokość drzewa wszystkich osobników w populacji jest odwrotnie zależna od ich średniej oceny. Podobnej zależności nie ma w stosunku do osobnika najlepszego, którego ocena stopniowo pnie się do góry mimo zmian w wysokości drzewa.

Na wykresach 4.2 można też zauważyć, że podczas gdy zastosowane ograniczenie wysokości drzewa dobrze ogranicza jego wartość maksymalną, to średnia wysokość może mieć tendencję do spadku. Oczywiście takie było założenie, zgodnie z literaturą

mniejsze drzewa mają większe możliwości generalizacji. Spadek wysokości drzewa jest więc teoretycznie pozytywnym zjawiskiem, jednak nie występuje on zawsze - w niektórych przebiegach algorytmu wysokość drzewa utrzymuje się blisko wartości maksymalnej 4.3.

Ostatecznie, wykres 4.4 pokazuje, że im większy parametr wysokości drzewa, tym lepszy jest wynik na zbiorze treningowym. Oznacza to, że tworzone są bardziej skomplikowane strategie, lepiej dopasowujące się do danych. Wynik ten kontrastuje z poprzednimi wykresami prezentującymi ocenę osobników w trakcie działania algorytmu.

Gorsza ocena średniego osobnika może więc wynikać z faktu, że strategie reprezentowana przez większe drzewo są bardziej skomplikowane, więc trudniej jest je dopasować do danych. Większe możliwości pozwalają jednak najlepszym osobnikom osiągnąć poziomy przystosowania niedostępne dla tych mniej skomplikowanych.

#### 4.3.2. Wysokość drzewa na danych testowych

Na danych testowych widoczna jest zupełnie odwrotna zależność niż na danych treningowych 4.5. Najlepsze wyniki osiągają proste strategie, zbudowane z drzewa o wysokości jedynie 2. Mimo prostoty, takie strategie mają większe możliwości generalizacji. To stwierdzenie pojawiło się już w literaturze. Zastanawiające są dobre wyniki przy wysokości drzewa równej 1. Jest to sytuacja, w której strategia składa się jedynie z dwóch wskaźników połączonych operatorem logicznym.

Wyniki sugerują, że konieczne jest unikanie dopasowania do zbioru treningowego.

### 4.4. Krzyżowanie i mutacja

Operatory krzyżowania i mutacji są ze sobą silnie powiązane. W teorii spełniają tę samą funkcję, edytują osobniki w celu eksploracji i eksploatacji przestrzeni rozwiązań. Krzyżowanie odgrywa główną rolę w eksploatacji, ponieważ nie wprowadza żadnych nowych informacji do modelu - modyfikuje jedynie te strategie, które wcześniej powstały. Powinno więc poprzez łączenie już dosyć dobrze przystosowanych osobników, generować ich poprawione wersje.

Oba operatory mają jednak możliwości szybkiej i zdecydowanej zmiany osobnika tak, że jego przystosowanie zmienia się diametralnie. Już edycja jednego tylko liścia w drzewie strategii, może spowodować zmianę przystosowania o tysiące pipsów. Wynika to z faktu, że wiele węzłów drzewa jest ze sobą powiązanych. Podczas gdy niektóre mogą być redundantne (na przykład połączone operatorem OR), to inne mają kluczowe znaczenie dla działania strategii. Na szczęście, nawet pogorszenie osobnika może pomóc wyjściu z lokalnego minimum poprzez wprowadzenie nowych genów.

Table 4.3: Wyniki na zbiorze treningowym bez mutacji

Zysk w pipsach	Odchylenie standardowe zysku
2222.5	726.207

Z krzyżowania i mutacji, mutacja jest ważniejsza. Podczas gdy sama mutacja bez krzyżowania osiąga podobne wyniki co przy zastosowaniu obu operatorów, to samo

Table 4.4: Wyniki na zbiorze testowym bez mutacji

Zysk w pipsach	Odchylenie standardowe zysku
367	2124.4

krzyżowanie nie ma takich samych możliwości dopasowania się do danych 4.3. Wynika to z ograniczonych możliwości wprowadzania nowych informacji - mutacja jest jedynym operatorem, które może wprowadzić nowe dane w liściach, czyli takie, które wcześniej się jeszcze nie pojawiły w żadnej generacji.

Co ciekawe, gorszy wynik na zbiorze treningowym zdaje się przekładać na lepszy wynik na zbiorze testowym. Być może występuje zjawisko zbytniego dopasowania się do danych treningowych, a może oddzielne szeregi czasowe nie mają ze sobą tyle wspólnego, aby strategia wyuczona na jednym była skuteczna na drugim. Fakt, że na zbiorze testowym średnia strategia jest jednak dochodowa, sugeruje jednak że podobieństwa są, jednak niewyraźne. Co więcej, dobry wynik na jednym zbiorze testowym nie gwarantują dobrego wyniku na innym zbiorze testowym.

Testy jednak dowodzą, że krzyżowanie, mimo że nie tak kluczowe, wpływa na poprawę wyników na zbiorach treningowych. Na wykresie 4.6, można zauważyć tę zależność. Wpływ krzyżowania na danych testowych można ocenić dopiero po ograniczeniu maksymalnej wysokości drzewa wykres `KrzyzowanieTest`. Widoczny jest pozytywny wpływ działania krzyżowania.

#### 4.4.1. Mutacja na danych treningowych

Przebadano wszystkie operatory mutacji osobno. Z wykresów działania z różnymi wartościami mutacji wynika, że każdy ma swoją funkcję. Obecność trzech z nich bezpośrednio poprawia wynik, a oba operatory wpływające na wysokość drzewa są niezbędne do utrzymania pożądanej jego wysokości. Przedstawione wykresy przedstawiają działanie algorytmu z różnymi wartościami kolejnych operatorów mutacji, podczas gdy te aktualnie niebadane są ustawione na 5%.

Na wykresach 4.10 widoczna jest nieregularność. Wynika ona z faktu, że nawet wielokrotne powtórzone badania cechują się dużą zmiennością wyników. Widoczna jest jednak przydatność operatorów w dopasowywaniu osobników do danych treningowych. Na wykresach prezentujących wpływ mutacji liścia czy węzła wewnętrznego otrzymywane wyniki są ewidentnie niższe gdy dany operator nie jest używany. W przypadku kolejnych dwóch operatorów mutacji otrzymywane wyniki są niepewne. Operatory te mają jednak rolę w kontroli wysokości drzewa strategii. Widoczne to jest na wykresach prezentujących średnią wysokość drzewa w kolejnych generacjach 4.9.

#### 4.4.2. Mutacja na danych testowych

Analizując dane testowe trudno ocenić działanie różnych operatorów mutacji. Powtarza się trend małych zmian w zysku i ciągłych dużych odchylen standardowych. Odchylenia nie zostały przedstawione na wykresach, bo uczyniłyby je nieczytelnymi. Wynoszą około 2000 pipsów, podczas gdy średnie wartości jedynie około 200.

## 4.5. Liczba generacji i liczba osobników w populacji

Analizując wykresy przedstawiające działanie algorytmu z różnymi wielkościami populacji czy liczbą generacji 4.11, nasuwa się stwierdzenie, że aby osiągnąć najlepszy wynik na zbiorze testowym, powinno się ograniczać wyniki na zbiorze treningowym. Oba parametry mają podobne działanie - umożliwiają na eksplorację większej przestrzeni rozwiązań, zwiększając potencjalną liczbę osobników w trakcie ewolucji. Ten zysk jest jednak obciążony kosztem - zwiększanie populacji lub liczby generacji liniowo zwiększa złożoność obliczeniową algorytmu. Nie jest to problem w przypadku pojedynczego użycia algorytmu, bo 100 generacji dla populacji 200 osobników przebiega w około 12 sekund. Zwiększający się czas wykonania może być jednak uciążliwy w przypadku wielokrotnych uruchomień w celu testowania.

Wyniki na dla liczby generacji równej 5 prawie odpowiadają przypadkowi, w którym wybierany jest najlepszy osobnik z losowej populacji. Nie świadczy to dobrze o algorytmie, sugerując że całe jego skomplikowanie jest redundantne. Jednak dla wysokości drzewa ustalonej na 2, optymalnej według poprzednich testów, otrzymane wyniki uzasadniają użycie algorytmu genetycznego i lepiej odpowiadają literaturze 4.12. Tym razem, prawdopodobnie ze względu na ograniczenie potencjalnego dopasowanie do danych treningowych, do pewnego stopnia korzystne jest umożliwienie dłuższej pracy algorytmu, jak i dostarczenie większej liczby osobników.

## 4.6. Wyniki po dostosowaniu hiperparametrów

Name	S1(S2)	S1(S3)	S1(S4)	4	S2(S3)	S
Zysk na zbiorze treningowym	1986.51	1986.51	1986.51		3218.22	32
Zysk na zbiorze testowym (bez prowizji)	275.205	4185.37	-1848.4		3608.05	-1
Zysk na zbiorze testowym	-130.195	3743.67	-1923.4		3188.64	-1
Ochylenie standardowe zysku na zbiorze testowym	633.818	2790.3	2454.43		3054.26	18
Liczba zyskownych transakcji	11.895	14.21	1.55		13.655	9
Liczba stratnych transakcji	8.375	7.875	2.2		7.315	
Liczba wszystkich transakcji	20.27	22.085	3.75		20.97	2
Odchylenie standardowe z liczby transakcji	23.1166	22.5228	5.64779		24.0454	22
Minimalny zysk	-2257	-4251	-3959		-6667	-
Pierwszy kwartył zysku	-306	3762	-3436		3182	-
Mediana zysku	-306	3920	-3436		4211	-
Trzeci kwartył zysku	-38	5932	-1004		4690	
Maksymalny zysk	2021	6344	3396		6290	3

Figure 4.1: Szeregi czasowe wykorzystywane przy testach

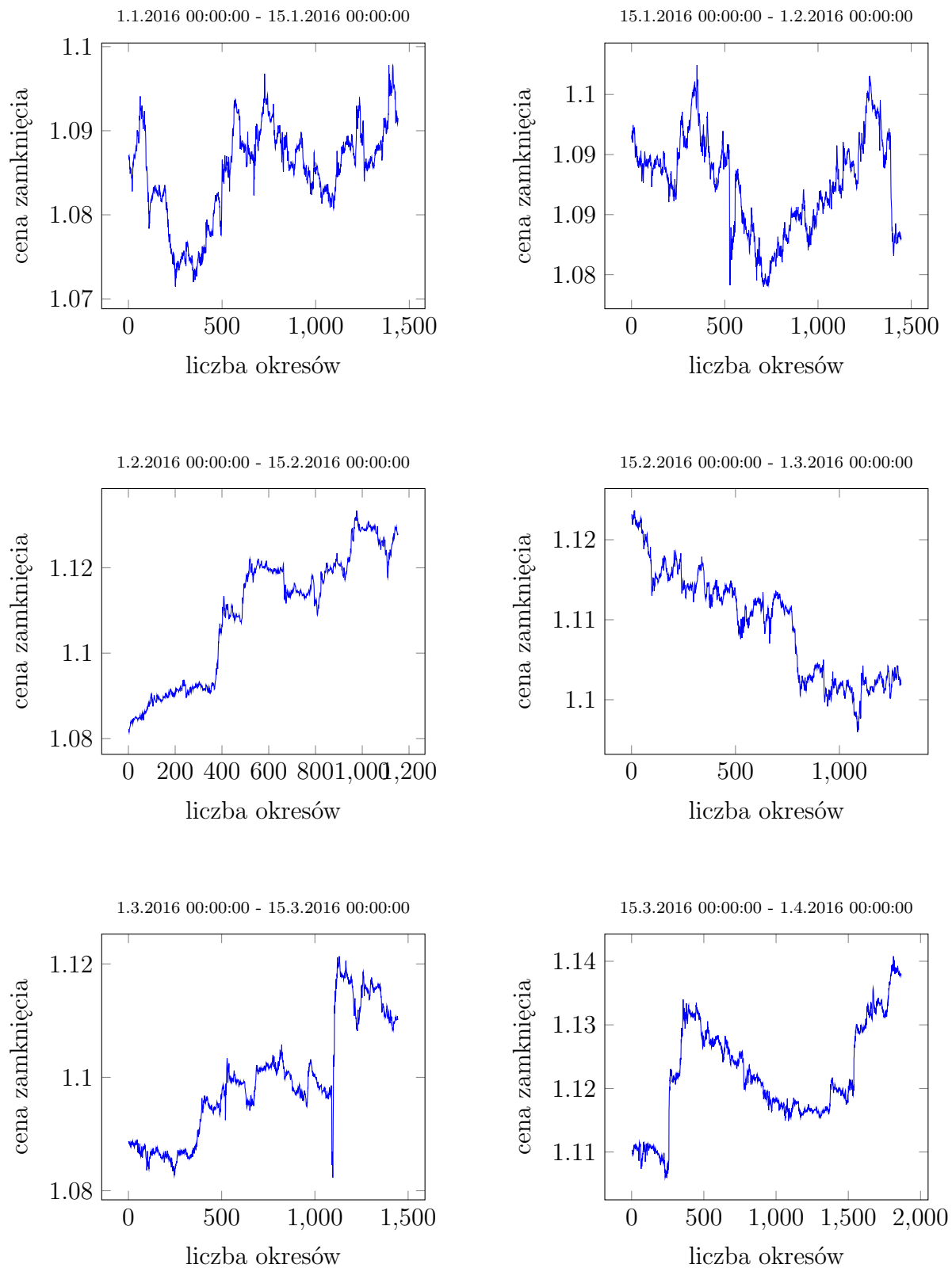


Figure 4.2: Zależność pomiędzy średnią oceną osobników a średnią ich wysokością

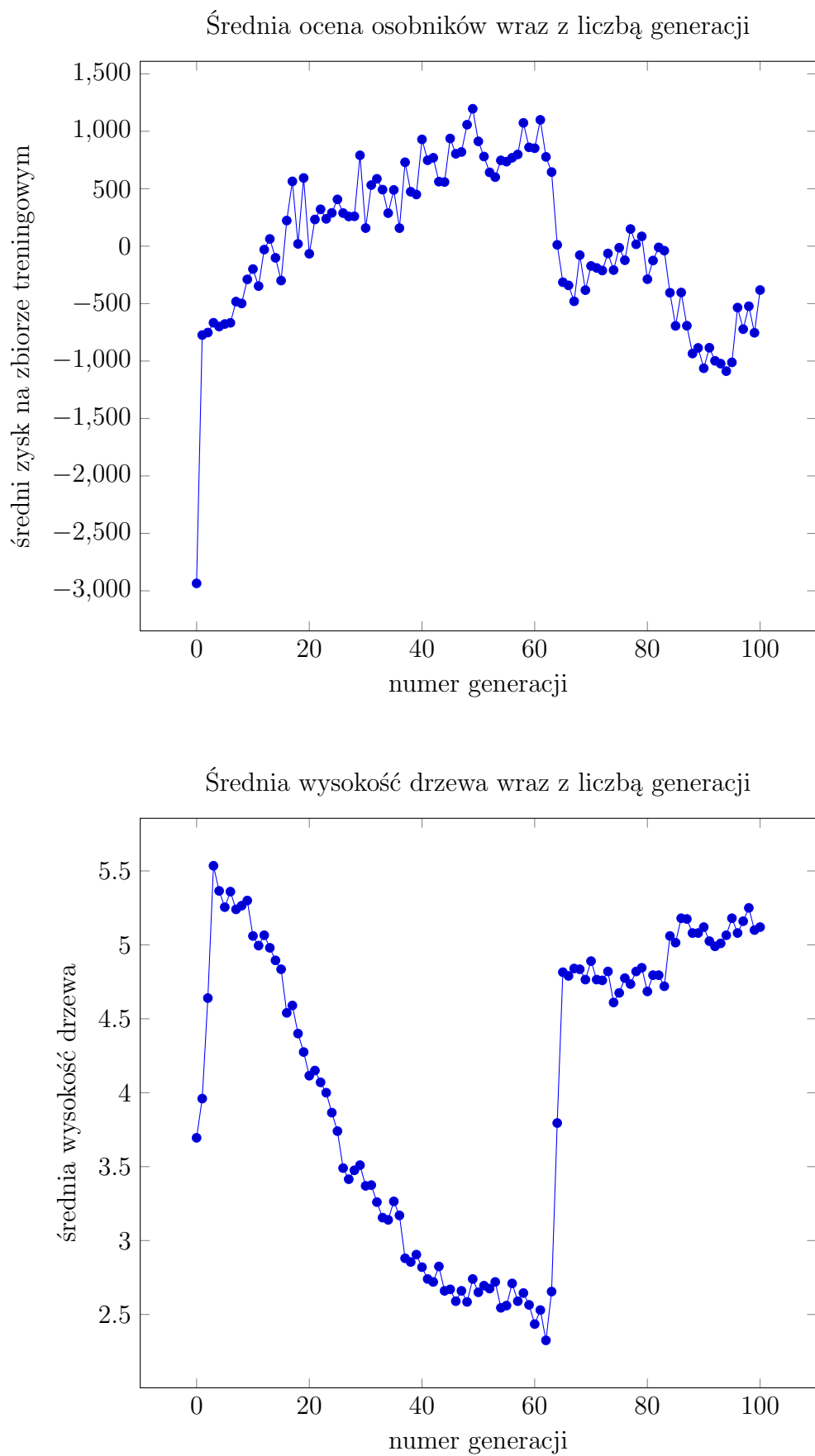


Figure 4.3: Średnia wysokość drzewa utrzymująca się w pobliżu maksymalnej = 5

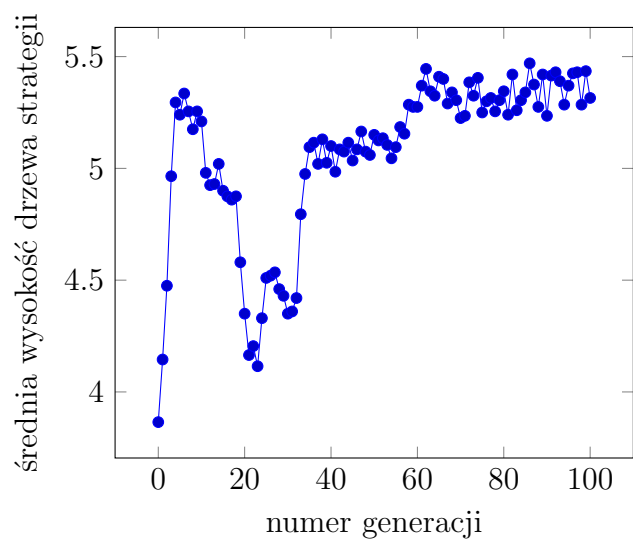


Figure 4.4: Średnia ocena osobników na danych treningowych z różnym parametrem wysokości drzewa

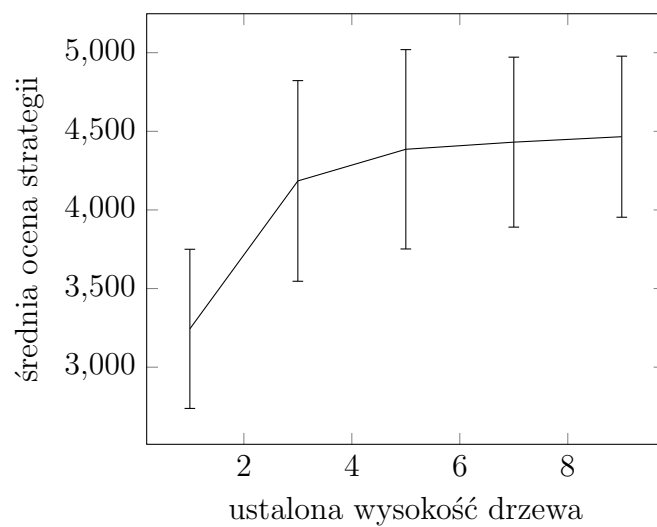




Figure 4.5: Średnia ocena osobników na danych testowych z różnym parametrem wysokości drzewa

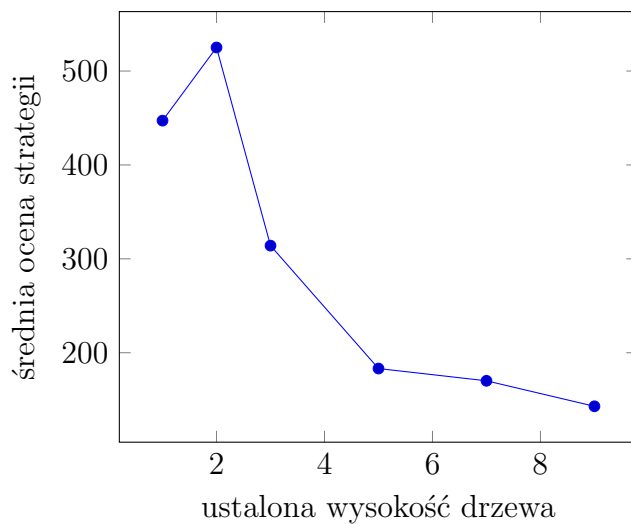


Figure 4.6: Działanie algorytmu z różnymi wartościami krzyżowania na szeregach treningowych

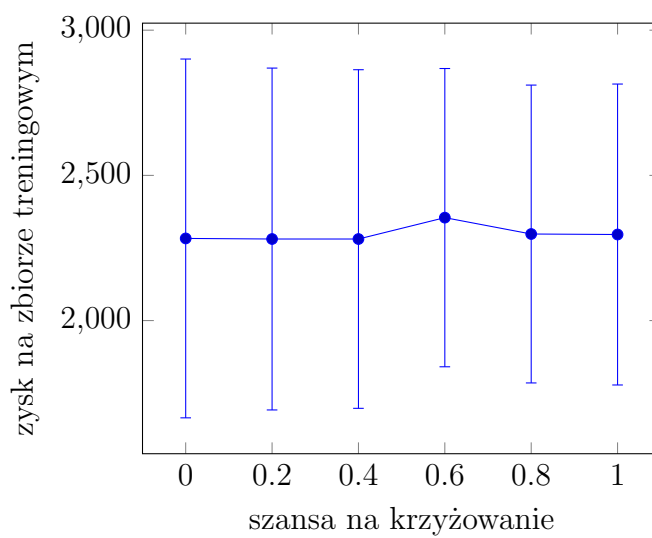


Figure 4.7: Działanie algorytmu z różnymi wartościami krzyżowania na szeregach testowych, wysokość drzewa = 2

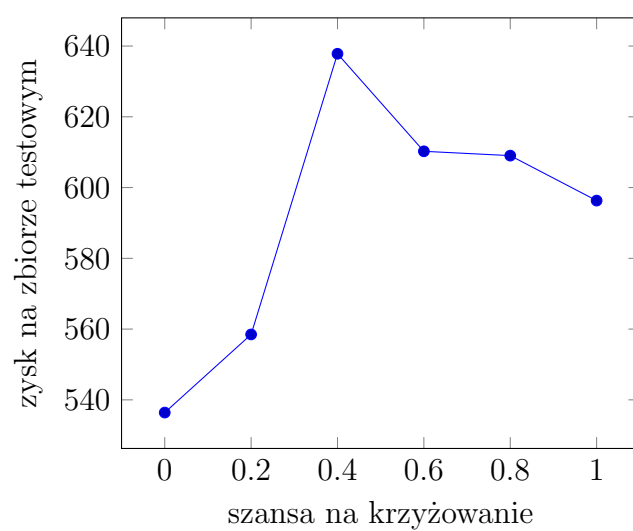


Figure 4.8: Działanie algorytmu z różnymi parametrami mutacji na danych treningowych

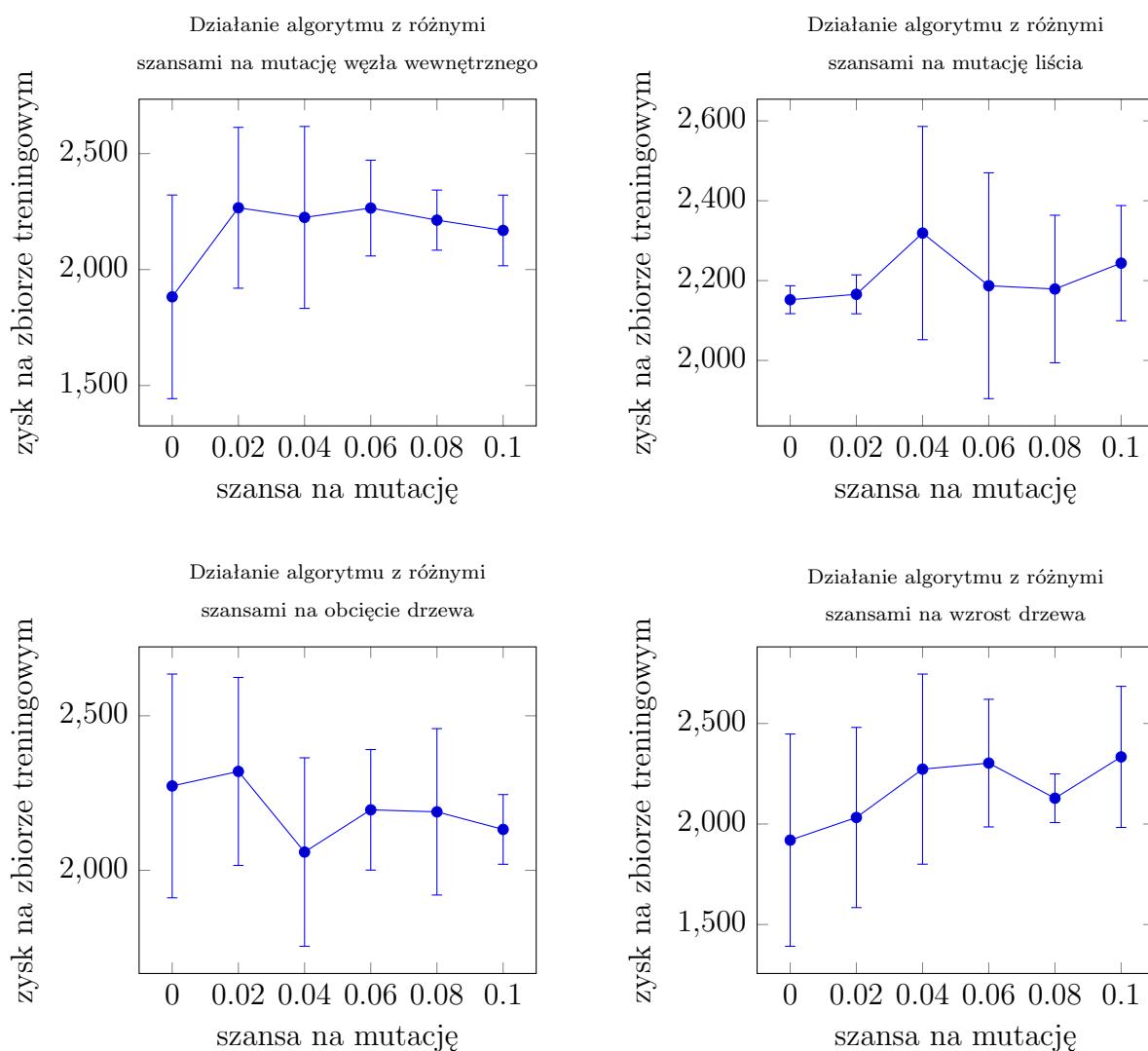
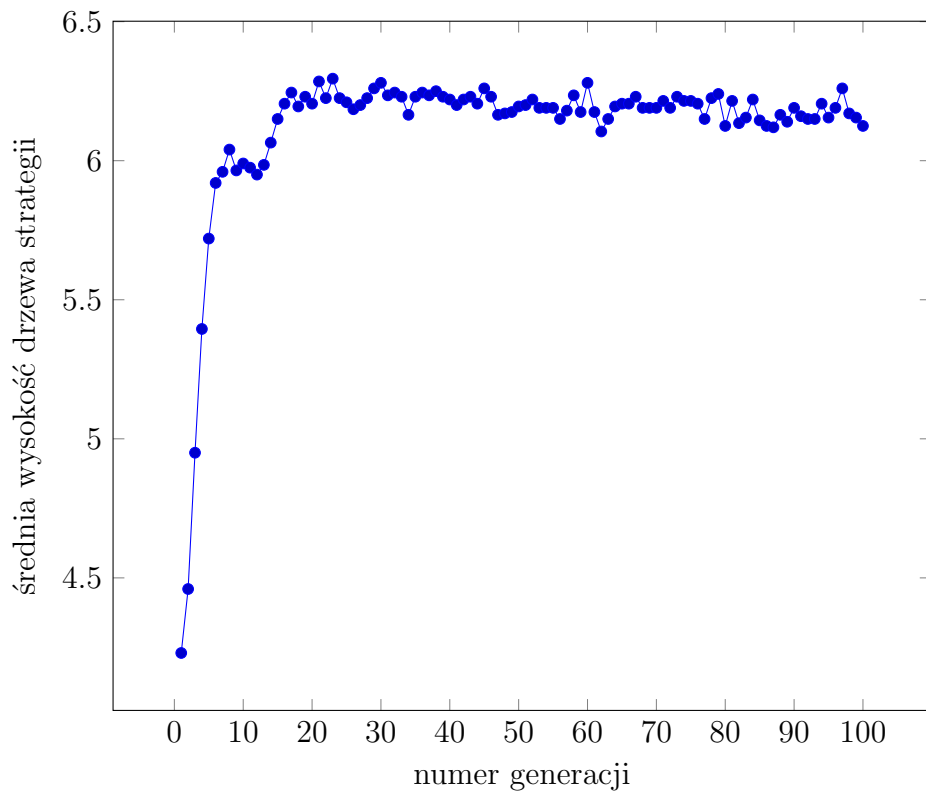


Figure 4.9: Działanie operatorów modyfikujących wysokość drzewa

Działanie algorytmu bez operatora skracania drzewa dla podanej wysokości



Działanie algorytmu bez operatora zwiększania drzewa dla podanej wysokości

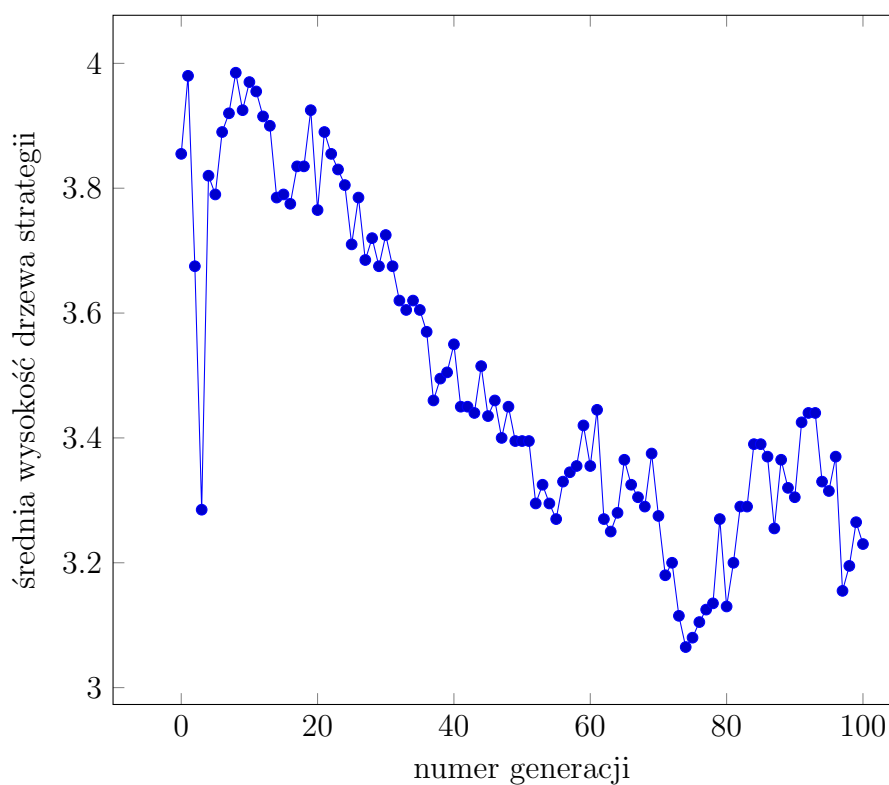


Figure 4.10: Działanie algorytmu z różnymi parametrami mutacji na danych testowych

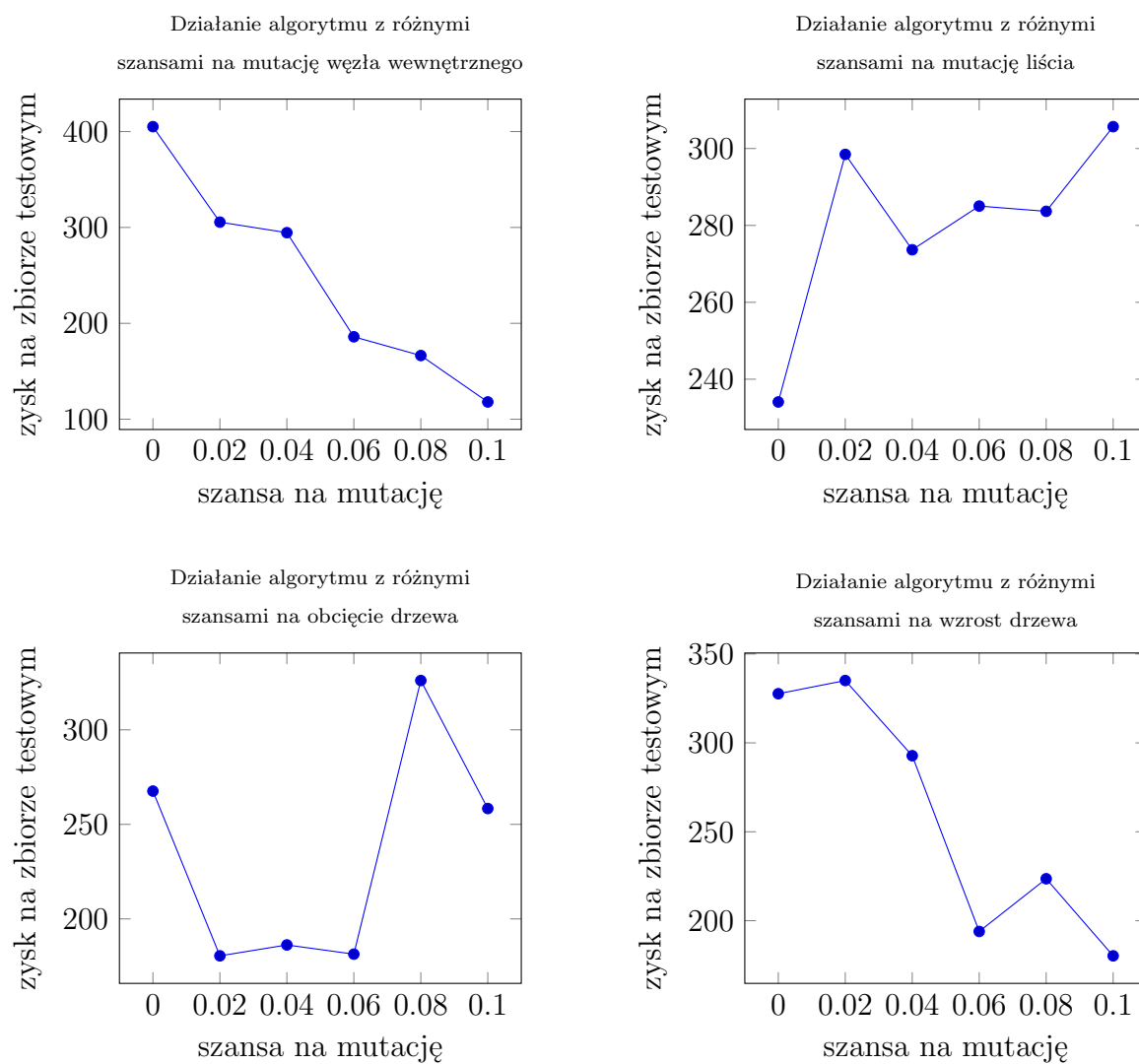


Figure 4.11: Działanie algorytmu z różną liczbą generacji i wielkości populacji dla wielkości drzewa = 5

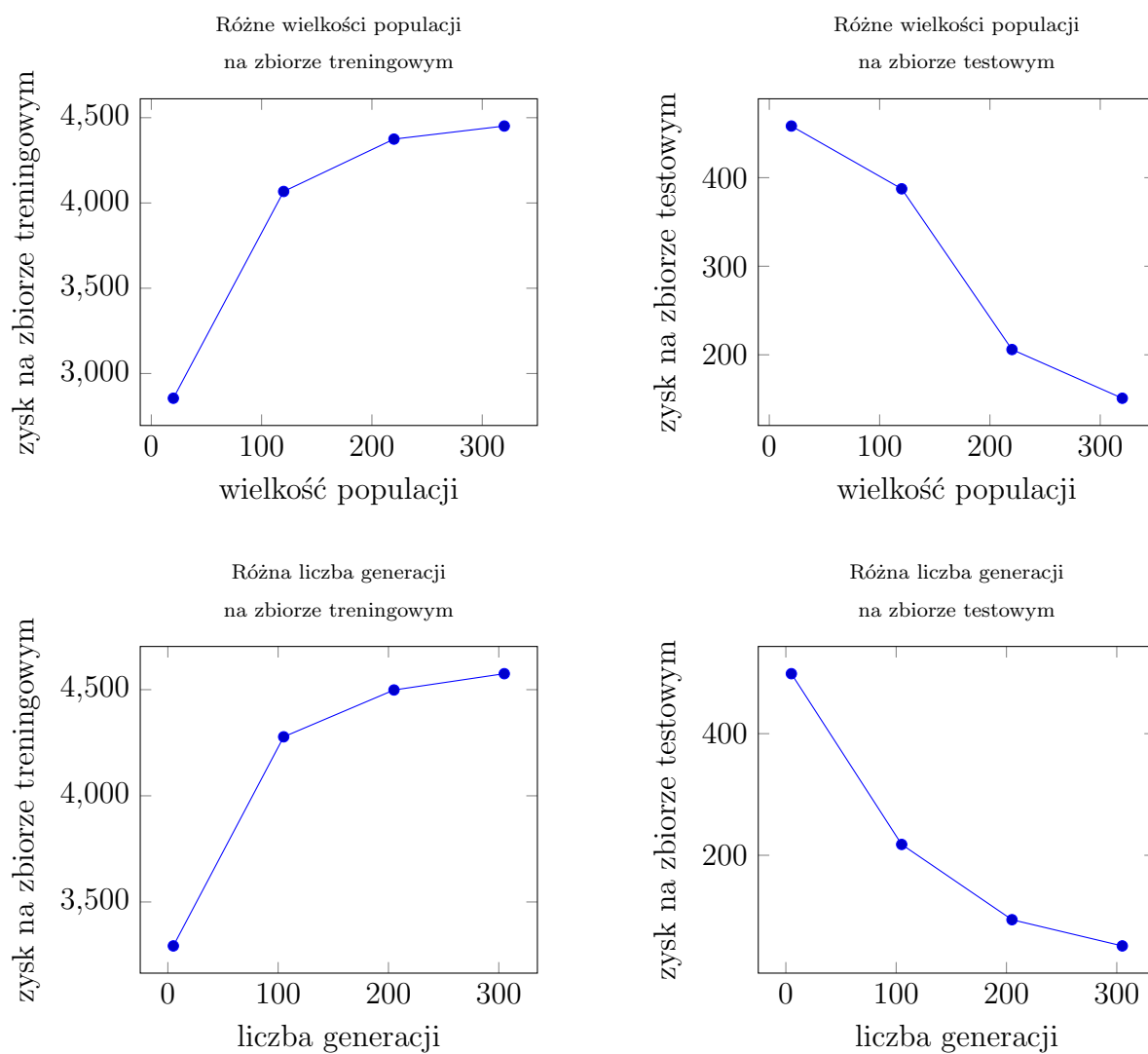
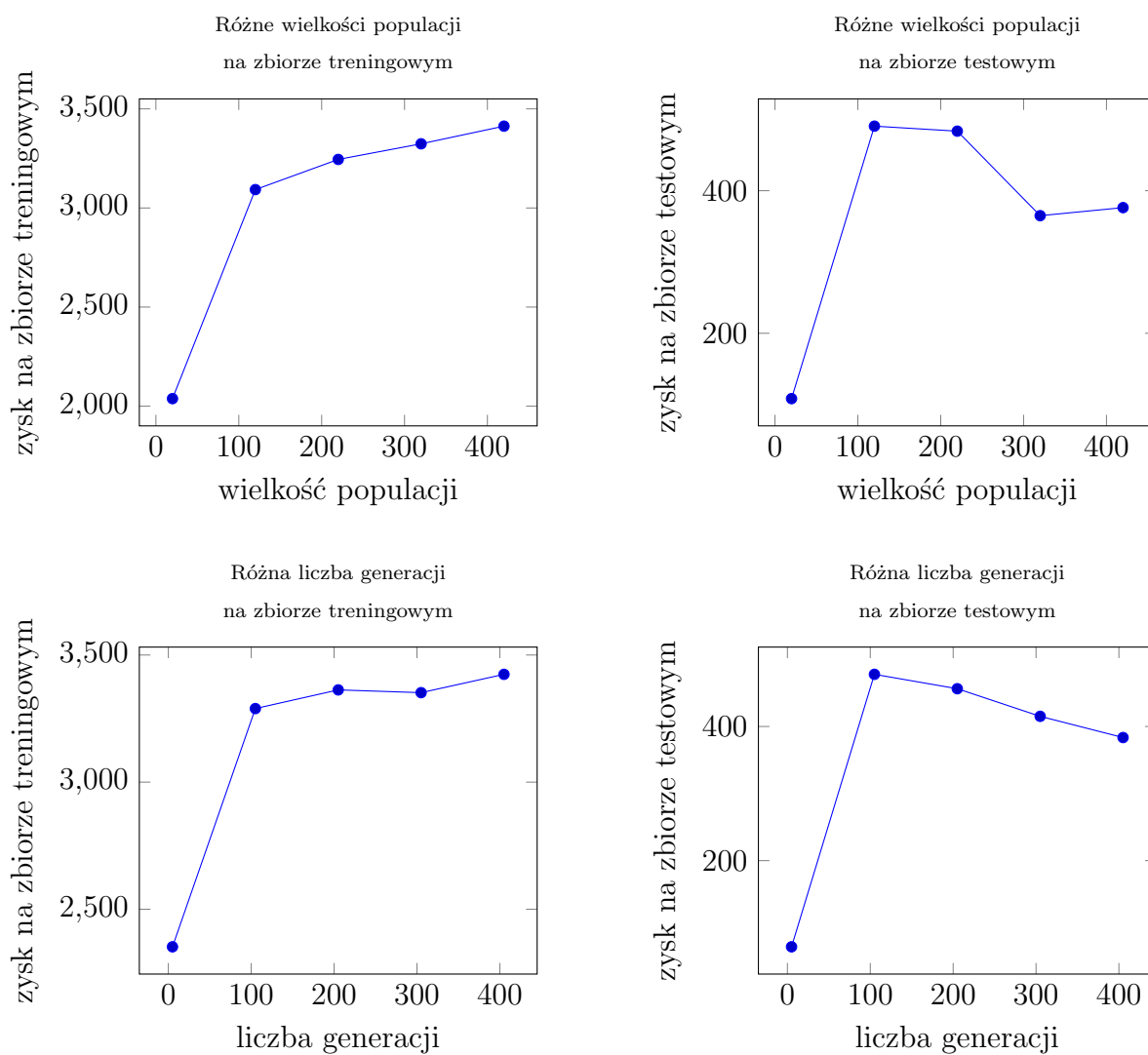


Figure 4.12: Działanie algorytmu z różną liczbą generacji i wielkością populacji dla wielkości drzewa = 2







## Chapter 5

# Analiza działania i możliwości poprawy



# Bibliography

- [1] Dukascopy historical data.
- [2] *Fuzzy Languages and their Relation to Human and Machine Intelligence, Man and Computer.*, 1970.
- [3] Allen F., Karjalainen R. Using genetic algorithms to find technical trading rules1. *Journal of Financial Economics*, 51(2):245 – 271, 1999.
- [4] Becker L., Seshadri M. Cooperative coevolution of technical trading rules, 2003.
- [5] Becker L. A., Seshadri M. Gp-evolved technical trading rules can outperform buy and hold. 2003.
- [6] Cavalcante R. C., Brasileiro R. C., Souza V. L., Nobrega J. P., Oliveira A. L. Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications*, 55:194 – 211, 2016.
- [7] Cormen T. H., Stein C., Rivest R. L., Leiserson C. E. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.
- [8] Corp. M. S. The fundamentals of testing in metatrader 5. 2011.
- [9] de la Vega J. *Confusion des confusions*. 1688.
- [10] Dhar S., Mukherjee T., Ghoshal A. K. Performance evaluation of neural network approach in financial prediction: Evidence from indian market. In *2010 International Conference on Communication and Computational Intelligence (INCOCCI)*, pages 597–602, Dec 2010.
- [11] Fogel G. B., Fogel D., Fogel L. Evolutionary programming. *Scholarpedia*, 6(4):1818, 2011. revision #90772.
- [12] Investopedia. Forex - fx. 2017.
- [13] Investopedia. Stock market. 2017.
- [14] Lohpetch D., Corne D. Discovering effective technical trading rules with genetic programming: towards robustly outperforming buy-and-hold. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 439–444, Dec 2009.

- [15] Lones M. A. Enzyme genetic programming, modelling biological evolvability in genetic programming, 2003.
- [16] Mendes L., Godinho P., Dias J. A forex trading system based on a genetic algorithm. *Journal of Heuristics*, 18(4):627–656, 2012.
- [17] Murphy J. *Technical Analysis of the Financial Markets*. 1999.
- [18] Myszkowski P. B., Bicz A. Evolutionary algorithm in forex trade strategy generation, Oct 2010.
- [19] Myszkowski P. B., Rachwalski . Trading rule discovery on warsaw stock exchange using revolutionary algorithms. In *2009 International Multiconference on Computer Science and Information Technology*, pages 81–88, Oct 2009.
- [20] Nassirtoussi A. K., Aghabozorgi S., Wah T. Y., Ngo D. C. L. Text mining of news-headlines for {FOREX} market prediction: A multi-layer dimension reduction algorithm with semantics and sentiment. *Expert Systems with Applications*, 42(1):306 – 324, 2015.
- [21] Neely C. J. Risk-adjusted, ex ante, optimal technical trading rules in equity markets. *International Review of Economics and Finance*, 12(1):69 – 87, 2003.
- [22] Polya G. *How to Solve It*. 1945.
- [23] Rechenberg I. *Evolution Strategy: Nature’s Way of Optimization*, pages 106–126. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989.
- [24] Seising R., Tabacchi M. E. A very brief history of soft computing: Fuzzy sets, artificial neural networks and evolutionary computation. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pages 739–744, June 2013.
- [25] Seth S. The world of high frequency algorithmic trading. 2015.
- [26] Si Y.-W., Yin J. Obst-based segmentation approach to financial time series. *Engineering Applications of Artificial Intelligence*, 26(10):2581 – 2596, 2013.
- [27] Simon H. A., Newell A. Heuristic problem solving: The next advance in operations research. *Operations research*, 6(1):1–10, 1958.
- [28] Sorensen K., Sevaux M., Glover F. A history of metaheuristics. 2017.
- [29] Staff I. Guide to stock-picking strategies. 2017.
- [30] Stephen D. Simpson C. The death of the trading floor. 2011.
- [31] Thomas J. D., Sycara K. The importance of simplicity and validation in genetic programming for data mining in financial data. 1999.
- [32] Vincenzo R. Does using firefly algorithm to find effective optimized trading rules produce greater results compared to a simple b&h strategy? Master’s thesis, Reims Management School, 2003.
- [33] Yang X. Metaheuristic Optimization. *Scholarpedia*, 6(8):11472, 2011. revision #91488.