creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

**저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.**

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학석사 학위논문

# 유전 프로그래밍을 활용한 동적 롱숏 포트폴리오 최적화 및 주식시장 패턴 연구

2023년 2월

서울대학교 대학원

컴퓨터공학부

김 창 겸

# 유전 프로그래밍을 활용한 동적 롱숏 포트폴리오 최적화 및 주식시장 패턴 연구

지도 교수 문 병 로

이 논문을 공학석사 학위논문으로 제출함
2022년 12월

서울대학교 대학원
컴퓨터공학부
김 창 겸

김창겸의 공학석사 학위논문을 인준함
2022년 12월

위 원 장 _____ 이 재 욱 _____ (인)

부위원장 _____ 문 병 로 _____ (인)

위 원 _____ 김 건 희 _____ (인)

# 초 록

동적 포트폴리오 거래 시스템은 포트폴리오의 수익을 극대화하고 포트폴리오의 위험을 최소화하는 방식으로 시간에 따라 동적으로 여러 주식 종목들에 자본을 할당하는데 사용된다. 인공지능 기술인 유전프로그래밍은 금융 분야 특히 금융시장에 대한 단기적인 예측을 하는 작업에 성공적으로 사용되었다. 이 논문에서는 유전프로그래밍을 시간 경과에 따라 주식 시장의 가격의 흐름을 예측하여 최적의 포트폴리오를 구성하는데 활용한다. 본 연구에서 제안하는 접근 방식은 포트폴리오의 비중을 구성할 때 포트폴리오 내의 종목들의 상대적인 상황을 고려하여 포트폴리오의 비중을 결정할 수 있도록 고안하였다. 종목들의 상대적인 관계를 유전프로그래밍의 트리 구조 내에서도 고려하기 위해 일반적인 수치연산자 뿐만 아니라 종목들 사이의 상대적인 순위(rank)와 일정 기간내의 표준 점수(Z score)를 계산할 수 있는 연산자들을 추가하여 공간을 탐색하였다. 또한 동적 포트폴리오의 시스템의 수익성에 큰 영향을 미치는 거래비용을 고려하여 시스템을 개발하였다. 거래 비용을 고려한 포트폴리오를 비중조절 없이 매수 후 보유한 포트폴리오와 비교하여 비중 조절이 실제로 얼마나 유의미한 수익을 창출했는지 확인해보았다. 개발한 동적 롱숏 포트폴리오 시스템의 유의성을 확인하기 위해 롤링방식을 활용하여 다양한 기간에서도 안정된 포트폴리오를 구축할 수 있는지 실험했다. 실험의 결과를 통해 다양한 기간에서 비중조절을 통해 꾸준한 수익을 창출해냈다는 것을 확인할 수 있었다. 추가적으로 일반적인 데이터 기반의 모델에 비해 모델의 작동원리를 보다 자세히 알 수 있다는 유전프로그래밍의 특징을 활용하여 시장에서 반복되는 현상을 분석하는 실험을 진행했다. 이를 위해 해석에 용이한 Boolean type의 트리를 구성하여 실험을 진행하였으며 패턴의 지속성을 판단하기 위해 test기간의 성능과 train기간의 성능의 비율을 확인했다. 실험 결과를 통해 시장에서 찾은 패턴의 빈도수가 그 패턴의 지속성에 강한 영향을 미친다는 것을 확인할 수 있었다.

**주요어** : 유전 프로그래밍, 동적 포트폴리오 최적화, 주식 패턴 연구
**학 번** : 2019-24908

# 목 차

제 1 장 서론 ..... 4
제 2 장 유전 프로그래밍 ..... 6
제 1 절 유전알고리즘과 유전프로그래밍 ..... 6
제 2 절 유전 프로그래밍의 기본 과정 ..... 7
제 3 장 유전 프로그래밍 구조와 설정 ..... 10
제 1 절 해의 표현 ..... 10
제 2 절 초기해 형성 ..... 13
제 3 절 해의 평가 방식 ..... 13
제 4 절 유전 연산자 ..... 15
제 4 장 실험 결과 ..... 15
제 1 절 실험 데이터셋 ..... 15
제 2 절 실험 변수 설정 ..... 17
제 3 절 실험 결과 분석 ..... 18
제 5 장 결론 및 제언 ..... 26

참고문헌 ..... 27
Abstract ..... 29

# 표 목차

[표 1] 트리 노드를 구성하는 연산자 목록 ..... 11
[표 2] 데이터 정보 ..... 16
[표 3] Numeric type 실험 설정 ..... 17
[표 4] Boolean type 실험 설정 ..... 18
[표 5] 동적 롱숏 포트폴리오 P&L결과 ..... 19
[표 6] P&L과 sharp ratio의 test/train 결과의 비율 ..... 20

# 그림 목차

[그림 1] 유전프로그래밍 기본 과정 ..... 8
[그림 2] Crossover 진행방식 ..... 9
[그림 3] 트리 예시 ..... 12
[그림 4] n개의 해집단 예시 ..... 13
[그림 5] Numerical pattern graph ..... 21
[그림 6] 동적 롱숏 포트폴리오 P&L 그래프 ..... 22

[그림 7] Boolean pattern 2 graph ..... 23
[그림 8] 두번째 패턴의 P&L 그래프 ..... 23
[그림 9] 역망치형 패턴들의 P&L 그래프 비교..... 24
[그림10] Boolean pattern 3 graph ..... 25
[그림11] 세번째 패턴의 P&L 그래프..... 25

# 제 1 장 서론

동적 포트폴리오 최적화는 주어진 종목의 비중을 특정 금액으로 투자하는 최적의 방법을 찾는 문제이다. 이 문제를 해결하기 위해서는 종목에 대한 선정이 이루어져야 하고, 매수 매도 여부가 정해져야 하며 마지막으로 각 종목에 대한 비중 조절 방식이 결정돼야 한다. 포트폴리오 최적화 문제의 또 다른 접근방식으로 정적 포트폴리오 최적화 방식이 있는데 이는 각 종목의 비중을 정한 후 특정한 기간 동안 이를 유지하는 방식이라는 점에서 변화하는 상황에 맞추어 동적으로 비중을 계속해서 조절해 나가는 동적 포트폴리오 방식과 차이가 있다.

먼저, 정적 포트폴리오는 1952년 Markov witz에 의해 마련되었다. 그가 제안한 방식은 평균 분산 최적화 방식으로 위험을 최소화하고 기대수익을 극대화하는 아이디어를 기반으로 최적의 포트폴리오를 설계하는 모델이다. 일반적으로 포트폴리오 기대 수익률은 포트폴리오에 선정된 종목들의 기대 수익률의 선형 조합으로 표현하였고, 수익률의 모멘텀들을 통해 포트폴리오의 위험을 표현하고자 하였다. 종목 선정에 따라 이러한 기대 수익과 위험이 변화하는 문제 상황 속에서 최적의 포트폴리오를 설계하기 위해 많은 연구들이 인공신경망이나[1] 유전알고리즘이나 시뮬레이티드 어닐링 같은 메타 휴리스틱 방식의[2] AI 기술을 도입했다.

동적 포트폴리오 방식은 주식들에 대해 저평가됐는지 고평가 됐는지에 대해 지속적으로 평가하여 보다 높은 수익률을 얻을 수 있도록 비중을 조절하고자 한다. 비중을 올바르게 조정하기 위해서 주식들의 미래 경향성에 대한 올바른 판단이 매우 중요한 요소인데, 이를 위한 한가지 방법으로 기술적 분석을 활용하여 주식의 가격 변화를 예측하는 방법이 사용되었다. 가격의 변화를 예측하기 위해 선형적인 모델이 처음으로 적용되었다. 하지만 주식 데이터의 높은 변동성 때문에 예측에 어려움을 겪으며 이를 극복하기 위해 다양한 AI 기술이 적용되기 시작했다. 특히 지속적으로 변화하는 시계열 데이터에서 기술적인 가격의 경향성을 예측하기 위해 artificial neural network(ANN) 분야와 genetic algorithm(GA) 분야에서 많은 연구가 이루어졌다[3].

AI 분야의 주요 접근 방식을 살펴보면, ANN은 가격의 경향성을 정교하게 예측하는 능력 덕분에 금융데이터를 활용하여 예측하고, 트레이딩 시스템을 개발하는 것에 대해 방대한 연구가 진행되어 왔다. 다양한 연구 가운데 단기적인 가격 경향성 예측을 기반으로 한 포트폴리오 최적화 모델에 대한 연구도 이루어졌다[4]. 하지만 금융데이터에 대한 광범위한 인공신경망의 활용에도 불구하고 해결해야할 중요한 문제로 남아있는 것이 있다. 그것은 ANN이 데이터를 기반한 모델로 어떤 규칙으로 작동하는지 명확하게 확인하기

어려운 블랙박스 모델이기 때문이다. 따라서 이러한 측면에서 투자자들은 모델의 발견을 통해 시장의 현상을 보다 구체적으로 이해하는데 도움을 받기 어려울 수 있다.

GA는 가장 인기있는 휴리스틱 최적화 기술 중 하나로 1975년 Holland에 의해 개발되었고 단순하고 제약이 거의 없다는 특징 때문에 다양한 최적화 문제에 널리 사용되어 왔다[5]. 정적 포트폴리오 최적화 분야에서 인덱스 펀드 관리를 위한 포트폴리오 최적화를 위해 사용되기도 하고 새로운 종목 구성으로 최적의 포트폴리오를 구성하는 것에 사용되기도 하였다[6]. 동적 포트폴리오 최적화 분야에서도 다양한 방식으로 적용되었는데, 미국 주식 시장에서 기술적인 거래 규칙을 찾는데 활용되고[7], S&P500 지수의 가격 변화의 경향성을 찾는 데에도 활용되었다[8].

GA가 금융분야에서 큰 기여를 하는 가운데 genetic programming(GP)은 구조적인 특징으로 역동적으로 변화하는 상황에 대한 정보를 추출하는데 그 유용성을 보여주었다. GA는 문자열 표현 형식을 사용하는데 반해 GP는 각각의 표현형이 문제를 해결할 수 있는 실행 가능한 프로그램을 의미하며 주로 트리형으로 표현된다. 지금까지 GP는 파산 예측[9], 비선형 시계열 예측[10] 및 개별 주식 트레이딩, 동적 포트폴리오 룰 탐색[11-16]과 같은 광범위한 금융 분야에 성공적으로 적용되었다. 파산 예측을 하기 위해서는 회사들 각각의 재무 지표들에 근거하여 decision tree 구조를 활용한 GP를 활용했고, 비선형 시계열 예측을 위해서 GP를 autoregressive integrated moving average(ARIMA)와 결합해서 활용했다. 주식 트레이딩 룰 탐색을 위한 연구들에서는 이진법(매수, 매도) 혹은 삼진법(매수, 보유, 매도)의 결과값을 내는 트리를 활용하여 트레이딩 룰을 탐색하는 방식을 사용했다. 앞의 연구들을 통해 GP를 활용하여 개별적인 주식들 혹은 시장의 지수 상황에 대해 동적인 트레이딩 시스템을 탐색하는데 효과적임이 확인되었다. 하지만 GP를 활용하여 동적으로 주식시장을 예측하는 과정에서 오직 해당 종목의 상황만으로 평가가 진행되는 구조를 보여준다. 이러한 구조는 GP를 활용한 공간 탐색 과정에서 종목들 사이의 관계성이 반영된 공간에 대한 탐색이 제한된다는 한계점을 보인다.

본 연구는 앞에서 언급한 한계점을 보완하기 위해 동적 롱숏 포트폴리오를 개발하고자 시도하였고, 시장에서 자주 반복되는 현상을 탐색하고 이를 관찰하기 위해 추가적인 실험을 진행하였다. 동적 포트폴리오의 비중을 조절하는데 있어 종목들의 절대적인 상황만으로 비중을 결정하는 것이 아닌 종목들 사이의 관계성에 의해 비중이 결정될 수 있도록 동적 롱숏 포트폴리오 트레이딩 시스템을 개발하고자 했다. 이 트레이딩 시스템은 시간 경과에 따라 포트폴리오 가중치를 업데이트하여 주가 변화에 적응할 수 있도록 한다. 이전의 연구에서 확인된 패턴 인식과 주식 시장에 대한 동적 예측에 대한 GP의 활용성을 고려하여, 최적의 포트폴리오를 탐색하는데 이 기법을 사용한다. 이를

위해 하나의 트리 구조는 각 종목에 대해 이진 형식이 아닌 실수 값을 출력하도록 구성했으며, 출력된 실수 값은 그 자체로 비중으로 활용되는 것이 아니라 종목들 사이의 상대적인 관계를 비중에 반영하기 위해 각 종목들의 출력 값을 중립화하는 과정을 거친다. 평가 지표로는 기간 동안의 최종적인 수익률과 위험을 고려한 수익률로 가장 많이 사용되는 샤프지수를 사용한다. GP를 통해 생성한 포트폴리오의 비중조절 방식을 사용한 전략과 비중 조절 없이 종목들을 매수 후 보유하는 전략을 취했을 때의 평가 지표를 비교하여 개발한 트레이딩 시스템을 평가한다. 동적 포트폴리오를 구성하는 것은 매수 후 보유하는 전략에 비해 많은 거래비용을 야기하기 때문에 이 거래비용을 능가하는 수익을 창출해내는 것은 포트폴리오의 평가에 있어서 매우 중요한 요소이다. 또한 ANN 모델에 비해 시장의 현상을 분석하기에 용이하다는 GP의 특성을 고려하여 시장에서 자주 반복되는 현상을 발견하기 위해 추가적인 실험을 진행하여 발견한 특성에 대해 개발한 동적 롱숏 포트폴리오의 특징과 함께 분석해보고자 했다.

본 연구는 포트폴리오 최적화 공간을 탐색하는데 세가지 측면에서 참신한 관점을 제공한다. 첫번째는 종목들의 비중을 결정하는 방식이다. 종목들의 절대적인 상황을 통해 비중을 책정하는 것에서 나아가 다른 종목들의 상황이 비중을 결정하는데 연관되도록 고안했다. 두번째는 트리를 구성하는 연산자에서 기존 연구에서 활용되지 않는 연산자를 새롭게 도입했다는 것이다. 먼저 다른 주식과의 관계성을 반영할 수 있는 연산자를 도입하였다. GP 트리 구조에서 실수 값을 연산할 때 종목들 사이의 상대성을 고려할 수 있도록 다른 종목들의 값들과 비교를 통해 scaling하는 연산자를 추가하였다. 다음으로 값 자체보다 변화하는 경향성이 더 큰 정보를 줄 수 있는 시계열 데이터의 특성을 고려하여 과거 기간동안 상대적인 위치를 알 수 있는 연산자를 고안하여 추가하였다. 세번째는 표현형이 자유롭고 해석에 용이하다는 GP의 특성을 활용하여 주식시장에서 반복되는 유의미한 패턴을 탐색하고 그 의미를 분석하였다.

# 제 2 장 유전 프로그래밍

## 제 1 절 유전알고리즘과 유전프로그래밍

유전 알고리즘(Genetic Algorithm)은 자연세계의 진화과정에 기초한 계산 모델로서 존 홀랜드(John Holland)에 의해서 1975년에 개발된 전역 최적화 기법으로, 최적화 문제를 해결하는 기법의 하나이다. 유전알고리즘은 풀고자 하는 문제에 대한 가능한 해들을 정해진 형태의 자료구조로 표현하고 이들을 변형시키는 과정을 통해 점점 더 좋은 해들을 생성해 나간다. 각각의 가능한 해를 하나의 유기체(organism) 또는 개체(individual)로 보며 이들의 집합을

개체군(population)이라 한다. 하나의 개체는 보통 한 개 또는 여러 개의 염색체(chromosomes)로 구성되며 염색체를 변형하는 연산자들을 유전 연산자(genetic operators)라 한다. 염색체를 표현하는 방법과 사용되는 유전 연산자의 종류 및 특성에 따라 여러가지 진화 알고리즘이 있는데 유전 알고리즘은 고정된 길이의 이진 스트링을 염색체로 사용한다[17].

유전알고리즘의 기본 과정은 문제에 적합한 해의 표현 방식을 설정하는 것과 해가 문제의 솔루션으로 얼마나 적합한지 평가하는 방식을 결정하는 것에서 시작한다. 유전알고리즘은 문제의 형태에 따라 해의 표현 방식과 평가 방식의 설정이 바뀔 수 있으므로 다양한 문제에 적용이 가능하다는 특징이 있고, 주어진 문제에 대한 좋은 솔루션을 찾기 위해서 이에 대한 설정은 매우 중요하다. 해의 표현 방식이 결정된 후에는 해집단을 이루는 해들을 무작위로 생성한다. 이때 해집단의 크기는 클수록 더 큰 다양성을 보장하지만 더 많은 계산 시간을 야기한다[7]. 해집단의 형성 이후에는 더 높은 fitness의 해들이 살아남는 환경에서 selection, crossover 그리고 mutation을 통해 세대를 거듭하여 보다 높은 fitness의 해를 탐색해 나간다.

유전프로그래밍은 유전알고리즘을 computer program 분야에 적용한 분야이다[5]. 유전프로그래밍은 사용자가 정의한 문제의 솔루션을 computer program의 형태로, 진화적인 과정의 연산을 통해 자동적으로 탐색하고 생성한다. 유전프로그래밍의 진화하는 해들 각각은 computer programs을 의미하고 그 형태는 트리를 기반으로 한 구조를 갖는다. 이러한 트리를 기반으로 한 해의 형태는 고정된 해의 형태에서 공간을 탐색하는 유전알고리즘과 달리 해의 형태의 크기가 가변적이라는 특징이 있다. 이러한 특징을 갖고 있는 GP는 증폭기 설계, 필터 설계, 로봇 제어회로 설계 등 다양한 문제들에 적용되어 문제의 형태에 따라 가장 잘 맞는 모델 형태를 결정하고 목적함수를 설정하여 해당 공간을 탐색하는 과정을 통해 인간의 경험과 지식을 뛰어넘는 프로그램을 찾는데 기여하였다.

## 제 2 절 유전프로그래밍의 기본 과정

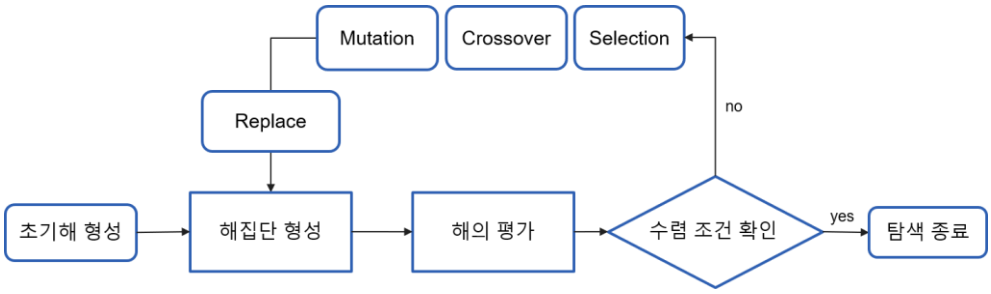
유전프로그래밍의 첫번째 기본 과정은 개별적인 초기 해집단을 생성하는 과정이다. 해의 형태는 문제를 풀기에 적합한 트리 형태를 이루어 각각의 해가 문제 상황을 해결하는 목적함수 역할을 한다. 일반적으로 초기해들은 미리 정해진 해집합의 크기만큼 랜덤하게 다양한 크기와 모양으로 형성된다.

두번째는 개별 해들이 얼마나 좋은 성능을 보이는지에 대해 문제 상황에 맞게 미리 정의된 fitness 함수를 활용하여 해들의 각 성능을 측정한다.

세번째는 다음 세대의 해집합을 생성하는 단계이다. 이 단계에서는 첫번째로 해집단을 구성하는 해들의 fitness를 고려한 해의 선택(selection)이 이루어진다. 이후 선택된 해들에 대해서 crossover, mutation 등의 유전자

연산을 실시한다. 이후 새롭게 생성된 해를 기존의 해집단의 해와 교체하는 과정을 통해 새로운 해집단을 생성한다.

네번째는 수렴조건을 확인하며 세대를 진행시키는 과정이다. 매 세대마다 미리 정해진 수렴 조건을 확인하며 이 수렴조건이 만족될 때까지 두번째와 세번째 단계를 통해 세대 진화를 지속한다. 수렴조건이 만족되면 일반적으로 마지막 세대까지 진행하며 최고의 성능을 보인 해를 최적해로 선정한다. 이러한 4가지 단계를 통해 유전 프로그래밍이 진행되며 이는 수많은 다양한 최적화 문제에서 좋은 성능을 보여주었다[18-19].



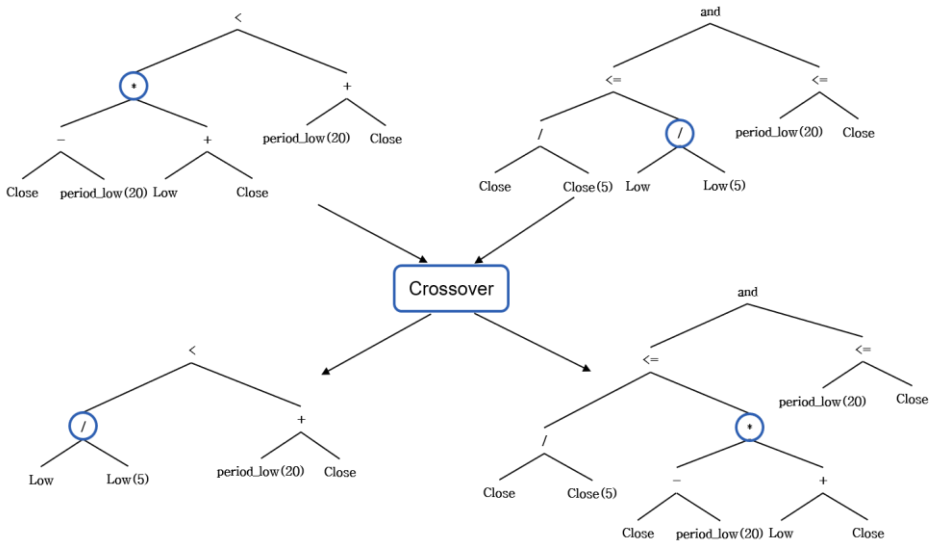

<그림 1> 유전 프로그래밍의 기본 과정

2.2.1 Selection

Selection은 이후 crossover에 활용될 두 개의 부모해를 고르기 위한 연산자이다. 진화 알고리즘에서는 룰렛 휠 선택과 토너먼트 선택 방식을 포함한 다양한 선택 방식들이 있다[20]. 이러한 다양한 연산자들의 공통적인 원칙은 해집단 내에서 우수한 해가 선택될 확률이 높아야 한다는 것이다. 우수한 해가 선택될 확률이 높다는 가정하에 그 정도를 조절하는 것을 선택압이라고 한다. 선택압이 높을수록 우수한 해를 뽑을 확률이 높은 것을 의미하며 이로 인해 수렴 속도는 빠르나 설익은 수렴의 가능성이 높아진다. 반대로 선택압이 너무 낮을 경우 충분한 시간이 지나도 해집단의 평균 품질이 개선되지 않을 가능성이 높아진다. 본 연구에서는 선택 방식에서 보편적으로 많이 사용되는 방식 중 하나인 토너먼트 선택 방식을 활용하였다. 토너먼트 선택 방식은 해집단에 대해서 k개(tournament size)의 해를 랜덤으로 선정한 후, 선정된 해들 중에 가장 좋은 해를 p의 확률로 두번째로 좋은 해를 $p*(1-p)$의 확률로 세번째로 좋은 해를 $p*(1-p)^2$ 의 확률로, 선정된 모든 해들의 성능에 따라 각각에 대해 $p*(1-p)^n$의 확률을 부여하여 selection을 진행하는 방식이다. 이 방식에서는 변수 k, p가 선택압에 영향을 미치는 요소로 작용한다. 본 연구에서는 p=1인 deterministic tournament 방식으로 진행하였다.

2.2.2 Subtree crossover

Crossover 연산은 유전 알고리즘의 대표 연산자이다. Crossover 연산은 가지고 있는 해집합 안의 유전형을 조합하여 더 좋은 해를 얻기 위한 연산으로 해의 형태에 따라 그 방식을 달리한다. 본 연구에서는 트리 구조를 활용하였기 때문에 트리 구조의 유전형의 특성을 결합하기 위해 subtree crossover 방식을 활용하였다. 일반적인 subtree crossover 방식에서는 교차점이 첫 번째 부모에 대해 무작위로 선택된다. 하지만 본 연구에서 첫번째 교차점이 선정된 이후 두 번째 부모에서 다른 교차 지점을 선정할 수 있도록 구성하였다. 이때 트리의 구조적 특징을 고려하여 교차 이후 트리의 결과값이 타당한 형태를 띄도록 제약을 주어 진행하였다. <그림 2>는 트리 연산의 결과값이 True, False로 나오는 Boolean type의 트리에 대해 교차 연산을 진행한 예를 보여준다. 이 예시에서는 두 함수 * 와 /가 각 트리에서 교차점으로 선택되었다. 그런 다음 교차 지점에 있는 하위 트리를 교환하여 두 가지 새로운 트리를 생성한다.




<그림 2> Crossover 진행방식

2.2.3 Mutation

Crossover 방식이 해집단 내의 해들의 조합으로 더욱 유용한 해를 탐색하는 과정이었다면 변이 연산은 해집단에 없는 속성을 도입하여 탐색 공간을 넓히려는 목적을 가진 연산이다. 전형적인 형태의 변이 연산자는 각각의 유전자에 대하여 [0, 1) 범위의 난수를 생성하여 미리 정한 임의의 임계값 미만의 수가 나오면 해당 유전자를 임의로 변형시키는 방식이다. 변이 연산은

문제 상황에 대한 조건 없이 임의로 일어나기 때문에 높은 확률로 해의 품질을 저하시킨다. 변이의 결과로 잘못 만들어진 유전자는 시간이 흐르면서 사라지고 가끔 발생하는 성공적 변이는 집단의 품질을 향상시키는 데 기여를 하게 된다. 변이 연산의 이러한 특성을 고려하여 해집단을 진화시키는 과정에서 변이를 적용하는 방식에 따라 균등 변이와 비균등 변이로 나눌 수 있다. 균등 변이는 미리 정한 임계값을 고정한 상태로 진화를 진행하는 것이다. 비균등 변이는 품질이 좋지 못한 초기 해집단에서는 개선의 가능성이 크지만 수렴이 어느정도 진행된 이후는 강한 변이로 해의 품질 향상이 이루어지기 어렵다는 특징을 고려하여 시간에 따라 변이 강도를 점차 줄여가는 방식이다.

#### 2.2.4 Replace

유전 연산자를 통해 새롭게 형성된 유전형들이 기존의 해집단을 개선하기 위해서는 대치(Replace)의 과정이 필요하다. Replace는 좋은 성능의 개체들이 사라지지 않으면서 해의 집단을 진화시키도록 이루어져야 한다. 대치 방식에는 가장 단순한 방식으로 해집단에서 품질이 가장 좋지 못한 해와 대치하는 방식이 있다. 이는 새로 만들어진 해가 즉각적으로 해집단의 진화에 영향을 주기 때문에 빠른 개선을 가져올 수 있으나 설익은 수렴을 조심해야한다. 다른 방식으로 crossover에 활용된 부모해 중 품질이 좋지 못한 해와 대치를 하는 preselection 방식이 있다. 이는 새로운 해와 가장 유사성이 높은 부모해들 중 하나와 대치함으로써 해집단의 다양성을 유지하는데 큰 도움을 주지만 수렴성이 많이 떨어지므로 최악의 품질의 해와 대치하는 방식과 혼합하여 사용되기도 한다. 이에 대한 추가적인 변형으로 전체 해집단 중에 후보해들을 선정한 후 그중 유사도가 가장 높은 것과 대치를 진행하는 crowding 방식이 제안되기도 하였다. 본 연구에서는 다양한 패턴을 생성하는데 초점을 맞추어 최대한 많은 수의 초기해를 생성하도록 시스템을 구성하였다. 이에 수렴 속도를 빠르게 하기 위해 기존의 해집단의 개체들 중 최악의 성능을 지닌 개체를 새롭게 생성한 개체와 교환하는 방식으로 전체 해집단의 크기를 유지하며 해집단을 개선해 나갈 수 있도록 구성했다.

# 제 3 장 유전프로그래밍의 구조와 설정

## 제 1 절 해의 표현

#### 3.1.1 트리의 구성

GP의 각각의 트리는 함수 집합(function set)과 말단 집합(terminal set)으로 이루어져 있고, 각각은 내부 노드와 말단 노드들을 구성한다. 본

연구에서 활용한 함수 집합은 논리 연산자, 비교 연산자, 수치 연산자, Scaler 연산자 그리고 Time series 연산자이다. 말단 집합에는 개별 주식들의 가격, 재무데이터를 활용하였는데 이에 대해서는 실험데이터에서 더 자세히 살펴볼 것이다. 함수 집합에서 논리 연산자, 비교 연산자, 수치 연산자는 GP를 활용하여 공간을 탐색하는데 가장 빈번하게 사용되는 연산자들이다. 본 연구에서 공간 탐색을 위해 별도로 추가한 Scaler 연산자와 Time series 연산자의 역할에 대해 살펴보자.

먼저 Scaler 연산자로 활용한 rank라는 함수는 rank의 입력값에 해당하는 각 종목들의 수치를 크기 순으로 나열하고, 이를 0에서 1사이의 값으로 대응시킨다. 이 과정을 통해 트리에서 연산하는 값이 그 자체의 크기로만 활용되는 것이 아니라 다른 종목과의 상대적인 값으로 활용될 수 있도록 하였다.

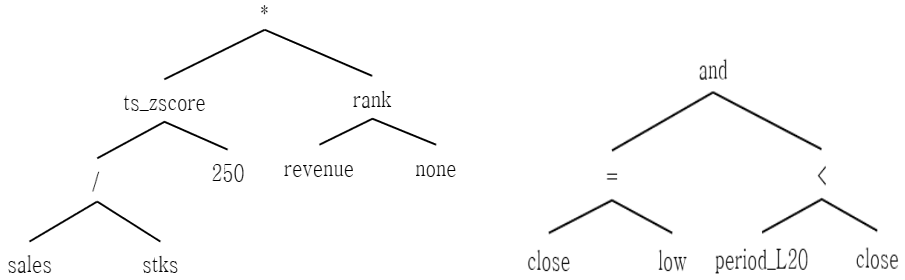
다음으로 Time series 연산자로 ts_mean과 ts_zscore를 활용하였다. ts_mean의 경우 입력값에 대한 주어진 기간 동안의 평균을 의미하며 이는 다른 실험에서도 자주 사용되는 연산자이다. 하지만 시계열 데이터는 그 특성상 특정 시점의 데이터의 값 자체보다 과거에 비해 상대적인 위치가 중요할 경우가 있다. 특히 재무데이터에서 그러한 특성을 관찰하기 쉬운데, 개별종목들의 현재의 매출 혹은 순이익 자체보다 과거 기간과 비교했을 때 어떤 상태에 있는지를 확인하는 것이 중요할 수 있다. 이러한 특징을 공간 탐색에서 반영하기 위해 ts_zscore 연산자를 추가했다. ts_zscore(x, window)는 각 주식에 대해 window 기간 동안에 해당하는 출력값들에서 x의 zscore 값을 구하는 함수이다. zscore는 기간 동안의 해당 값들을 통해 정규분포를 만들고 정규분포상 어떤 위치를 차지하는지를 보여주는 수치이다. 이를 통해 각 종목의 데이터를 절대적인 값 뿐만 아니라 시계열 상 상대적인 값으로 활용할 수 있도록 하였다.

| 연산자 | 종류 |
|---|---|
| **논리 연산자** | and, or |
| **비교 연산자** | >=, >, <=, < |
| **수치 연산자** | +, −, *, / |
| **Scaler 연산자** | rank |
| **Time series 연산자** | ts_mean(,window), ts_zscore(,window)<br>window = 3, 5, 10, 20, 60, 120, 250 |

<표 1> 트리 노드를 구성하는 연산자 목록

3.1.2 트리의 구조

본 연구에서는 주식 시장에서 수익을 낼 수 있는 전략을 개발하고 시장에 대한 이해도를 높이기 위해 두가지 형태의 트리를 활용했다. 동적 롱숏 포트폴리오를 생성하기 위해 Numeric type 트리를 활용했고, 주식 시장의 현상을 보다 자세히 관찰하기 위해서는 Boolean type의 트리를 활용했다. 각 트리의 구성을 살펴보자. 먼저 Numeric type의 트리의 경우 함수 집합은 수치연산자, Scaler연산자와 Time series 연산자로 구성되었고, 말단 집합에는 각 주식들의 가격, 재무데이터 그리고 time series 연산자의 입력 값으로 사용되는 기간을 나타내는 변수들이 활용되었다. 다음으로 Boolean type의 트리의 경우 함수 집합은 논리연산자, 비교연산자, 수치연산자로 구성되었고 말단 집합에는 각 주식들의 가격, 재무데이터들로 구성 되어있다. 루트 노드는 논리 연산자 and, or 중에 하나로 결정된다. 이후 트리를 생성할 때 트리의 올바른 구성을 위해 비교 연산자 이후에는 논리연산자가 출현하지 못하고 수치연산자 이후에는 논리연산자와 비교 연산자의 출현이 제한되도록 설정하였다. 논리 연산자는 Boolean 값에 대해 판단하는 것을 목표로 하며 비교 연산자는 실수 값에 대한 비교를 목표로 하기 때문에 올바른 입력값을 구성하기 위해 트리의 생성 단계에서부터 설정을 제한하였다. 이러한 트리 구성의 제약은 해를 진화시키는 과정에서도 동일하게 시행되었다.




<그림 3> 트리 예시(왼쪽이 Numeric type tree, 오른쪽이 Boolean type tree)

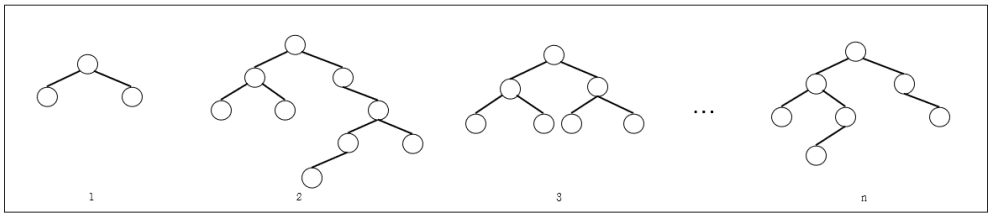
### 3.1.3 Time series 연산 속도의 개선

데이터의 관계성을 고려하기 위해 고안된 연산자들을 실제로 활용해서 공간을 탐색할 때 일반적인 수치 연산자처럼 단순한 로직으로 모든 데이터에 대해 연산을 시행하면 지나치게 많은 연산 시간이 소요된다는 것을 확인했다. 유전프로그래밍을 활용한 공간 탐색에서 제한된 시간동안 최대한 많은 공간을 탐색해야 하기 때문에 공간 탐색의 시간을 줄이는 일은 매우 중요하다. Time series 연산에서 zscore를 구하는 과정에는 특정 기간 동안의 값의 평균과 표준편차를 구하는 과정이 필요하다. 이때 매시점에서 새롭게 평균과 표준편차를 구한다면 데이터가 연속되어 존재하는 시계열 데이터의 특성상 많은 연산이 중복된다는 것을 확인했다. 연속된 날짜의 연산 결과는 이전 날짜의 연산에의 처음과 마지막 그리고 새롭게 추가되는 연산을 제외하고는

똑같이 진행된다. 이러한 현상은 라빈 카프 알고리즘을 통해 문자열 매칭 속도를 개선한 방법과 유사하다. 이러한 개선을 통해 time series 연산의 window 크기와 비례하는 배수만큼의 개선을 이끌어 낼 수 있었다.

# 제 2 절 초기해 형성

트리는 3.1.1에 설명된 트리의 구조에 맞추어 함수 집합과 말단 집합들을 사용하여 무작위로 생성되고, 미리 결정된 해집단의 크기만큼 생성된다. 일반적으로 초기해를 형성하는 방식에는 full, grow 그리고 ramped half and half 방식 이렇게 3가지 방식이 있다[20]. Full 방식에서는 모든 트리에 대해 미리 정의된 최대 깊이가 되도록 생성하여 각각의 트리들이 모두 동일한 모양과 크기로 생성된다. Grow 방식에서는 미리 정의된 최대 깊이 아래에서 더 다양한 모양과 크기를 가질 수 있도록 생성한다. 마지막으로 ramped half and half 방식은 위의 두 방식을 조합한 방식이다. 본 연구에서는 복잡도가 다양한 초기해를 형성하기 위해 full 방식을 선택하여 최소 트리의 높이와 최대 트리 높이를 설정한 후 좌우의 새로운 노드를 생성할 때 50%의 확률을 가지고 생성을 추가할지 멈출지를 판단하도록 하였다.




<그림 4> n개의 해집단 예시

# 제 3 절 해의 평가방식

3.3.1 Profit & Loss와 sharp ratio

Profit & Loss(P&L)와 sharp ratio는 투자 전략의 성능을 확인할 때 가장 대표적으로 활용되는 지표들이다. P&L는 정해진 기간동안 고정된 금액을 가지고 투자를 진행했을 때 실질적으로 얻을 수 있는 수익금 혹은 손실금을 의미한다. Sharp ratio는 투자성과를 평가함에 있어 해당 투자의 위험을 조정해 반영하는 방식이며, William F. Sharpe의 이름을 따 명명되었다.

$$\text{P\&L} = \sum_{d} r^{d}$$

$$\text{Sharp ratio} = \frac{\text{mean}(r^d)}{std(r^d)} * \sqrt{252}$$

그곳에서 $r_s^d = \frac{Close_s^{d+1} - Close_s^d}{Close_s^d} - 1$로 종목별 d일의 수익률을 의미하며, 이며 $r^d = \sum_s weight_s^d * r_s^d$ 으로 d일 전체 수익률의 총합을 의미한다.

본 연구에서는 유전프로그래밍의 성능을 평가하기 위한 fitness로 train 기간 전체의 P&L을 설정하였다. 이후 유전 프로그래밍을 통해 찾아낸 패턴 즉, 전략의 성능을 확인하는 과정에서 test 기간의 P&L 뿐만 아니라 sharp ratio도 함께 확인하며 전략의 성능을 분석하였다.

### 3.3.2 Boolean type과 Numeric type의 해의 평가 방식

본 연구는 유전 알고리즘의 표현과 목적함수를 다양하게 할 수 있는 특징을 활용하여 시장에서 유의미한 패턴을 찾기 위해서 어떤 형식의 패턴이 강한 지속성을 가질 수 있는지를 실험을 통해 확인하고자 하였다. 이를 위해 트리의 출력값이 Boolean type과 Numeric type을 갖는 두가지 형태의 트리를 정의하였다. 각 트리 출력 값이 어떻게 활용되고 각 패턴이 어떤 의미를 갖는지를 설명하고자 한다.

첫번째 Boolean type 트리의 경우 일자별로 각 주식에 해당하는 데이터에 대한 트리 연산의 값이 True(1), False(0)로 출력된다. d일의 각 주식에 대한 출력값에 대하여 전체 True의 개수로 출력값을 나눈 것을 각 주식의 weight로 하여 투자를 진행한다. 이때 fitness는 이 d일의 수익률의 총합이다.

두번째 Numeric type 트리의 경우 Boolean type과 동일하게 일자별로 각 주식에 해당하는 데이터에 대해 트리 연산이 진행되지만 그 값이 실수 값으로 출력된다. 이때 각 주식들의 출력 값들의 평균을 0으로 맞추고 절대값의 합으로 나누어 주는 과정을 통해 각 주식들의 비중을 결정한다. d일의 각 주식에 대한 출력값에 대하여 모든 출력값들의 평균을 빼서 전체의 평균을 0으로 만들고 이 값들의 절대값의 합을 나누어 주어 전체 비중을 1로 맞추어 투자를 진행한다. 각 비중에 대해 수익률을 곱한 값의 총합이 d일의 수익률이 된다. 이때 fitness는 이 d일의 수익률의 총합이다.

## 제 4 절 유전 연산자

### 3.4.1 유전 연산자

문제 공산에서 최적의 솔루션을 찾기 위해서는 구성한 해집단이 유전 연산자를 통해 세대를 거듭하며 진화해나가는 과정이 필요하다. 유전프로그래밍에서 진화 과정에 중요한 영향을 미치는 유전연산자에는

crossover, mutation, reproduction 과정이 있다. 본 실험에서는 crossover 방식으로 subtree crossover 방식을 선택했다. Subtree crossover는 selection 과정을 통해 선정된 각 트리에서 랜덤으로 노드를 선택하여 선택된 노드를 루트로 하는 서브 트리끼리 교환하는 방식이다. 본 연구에서는 연산자의 종류에 비교연산자와 수치연산자가 함께 사용되었으므로 교배하는 과정에서 트리의 구조가 무너지지 않도록 subtree의 연산 결과가 같은 형태인 노드들끼리 crossover가 가능하도록 진행했다. Mutation은 또 다른 유전 연산자로 미리 결정된 mutation rate를 바탕으로 개별적인 유전체에 적용된다. 본 연구에서는 selection과정으로 선정된 트리들이 crossover 과정을 마치고 생성된 트리에 대해 GP의 전통적인 mutation 방식 중 하나인 node mutation 방식을 적용했다. 연구에서 활용된 트리 구조의 특징을 고려하여 말단 노드들을 탐색하며 지정된 mutation rate로 같은 속성의 terminal set에 있는 데이터들 중에 무작위로 선택하는 방식을 활용했다.

### 3.4.2 지역 최적화

데이터들 사이의 관계성을 패턴 탐색과정에서 더 알아볼 수 있도록 time series 연산자를 추가적으로 고안해서 활용했지만, 탐색 과정에서 time series 연산자의 window에 따른 공간을 탐색하기에는 기본적인 유전연산자만으로는 부족함을 느꼈다. 이에 time series 연산자의 경우 다른 트리 구조를 유지하며 최적의 window를 탐색하는 지역 최적화 과정을 추가하여 공간 탐색을 진행했다.

# 제 4 장 실험 결과

## 제 1 절 실험 데이터셋

한국 주식데이터 2007년에서 2020년의 가격데이터와 재무데이터를 사용하여 실험을 진행하였다. 변동폭이 컸던 시기와 장기간 박스권 장세의 지속으로 수익을 내기 힘들었던 기간을 실험기간으로 두어 개발한 포트폴리오 및 패턴의 유의미성을 확인하고자 하였다. 금융시장은 2007년에 발생한 서브프라임 모기지 사태(subprime mortgage crisis)로 인해 세계 금융 위기를 맞이하고 그 어떤 시기보다 변동성이 컸던 시기를 맞이했다. 강한 폭락과 반등 이후 한국 주식시장은 이전의 고점을 넘는 강한 상승장을 맞이하지 못하고 2019년까지 박스권에서 횡보하는 흐름을 보였다. 이후 코로나 팬데믹으로 2020년 강한 변동성을 보였다. 이러한 기간적 특징을 고려하여 동적 롱숏 포트폴리오 모델이 다양한 시기에서 수익을 낼 수 있는지 확인하기 위해 3년의

train기간과 1년의 test기간을 두어 롤링하여 실험을 진행했다. 그리고 주식시장의 패턴을 연구하기 위한 실험에서는 폭락과 반등 그리고 횡보의 기간을 모두 포함한 2007년에서 2009년까지를 train기간으로 고정하여 다양한 장세에서 지속되는 패턴을 찾고자 하였고, test기간은 train 기간만큼 긴 2010년부터 2012년까지 3년을 설정하여 장기간 패턴의 유의성을 확인하고자 했다.

가격 데이터로는 단기적인 상황의 변화에 대한 예측을 필요로 하기 때문에 최근 20일 동안의 시가, 고가, 저가, 종가의 데이터와 거래량 데이터를 활용하였다. 중장기적인 상황에 대한 정보는 가격의 이동평균선 정보를 통해 제공하였다. 재무 데이터로는 투자자들이 많이 인지하고 있는 데이터로 자산, 부채, 매출, 영업이익, 순이익, 판매비용 등의 데이터를 활용하였다. 활용한 데이터의 자세한 내용은 표4에서 확인할 수 있다.

| 구분 | 설명 |
|---|---|
| Price | Open, High, Low, Close |
| Price(k) | K 일전의 가격, k=1,2, … , 20 |
| ma(k) | mean(price, k), k = 5, 10, 20, 60, |
| maV(k) | mean(volume, k), k = 5, 10, 20, 60 |
| period_high(k) | Max(price, k), k = 5, 10, 20 |
| period_low(k) | Min(price, k), k = 5, 10, 20 |
| 재무정보데이터 | 자산(Assets), 부채(Liabilities), 순자산(equity), 순매출(Net sales), 영업이익(operating profit), 순이익(Net profit), 매출원가(Cost of sales), 매출총이익(Gross margin), 영업비용(operating cost) |
| Stks | 상장주식수 |
| Mcap | 시가총액 |

<표 2> 데이터 정보

(mean(, k), min(, k), max(, k) 함수는 각각 k일간의 평균, 최소값, 최대값을 의미한다. 재무정보데이터의 단위는 만원이다.)

# 제 2 절 실험 변수 설정

### 3.3.1 Numeric type 트리 탐색의 실험 설정

동적 롱숏 포트폴리오 모델을 찾기 위한 최적의 변수는 여러 변수에 대한 실험을 통해 결정되었다. 다양한 패턴의 조합을 나타낼 수 있는 초기해집단의 크기를 설정하기 위해 1000-7000개까지 범위에서 초기해 집단에 대해 사전 실험을 진행하였으며 train기간에 안정적인 3000개 이상의 초기해를 형성했을 때 랜덤하게 생성된 초기해들의 성능에 지배되지 않고 보다 안정적으로 포트폴리오 탐색에 성공하는 것을 확인했다. 해집단이 커질수록 다양한 패턴이 보장되지만 이에 대한 평가의 시간이 늘어나고 수렴 속도가 느려지기 때문에 이를 고려하여 5000개의 해집단을 설정하였다. 이후 다양한 트리 높이를 통해 실험했을 때 최대 트리의 높이가 8개가 넘어갈 경우 train기간의 성능이 개선 속도가 확연히 줄어들었고 해의 구성을 살펴보았을 때 겹치는 패턴이 많은 상태에서 일부의 수정만 일어난다는 것을 확인할 수 있었다. 그 외 사전 실험을 통해 설정한 자세한 실험 설정은 표 3에서 확인할 수 있다.

| 실험 설정 | |
|---|---|
| 초기해집단 크기 | 5000 |
| 초기해집단 최소 트리 높이 | 1 |
| 초기해집단 최대 트리 높이 | 4 |
| 초기해집단 트리 생성 비율 | 0.5 |
| Selection 방식 | Tournament selection(k = 200) |
| Crossover 방식 | Subtree crossover |
| Mutation rate | 0.01 |
| 학습기간 최대 트리 높이 | 8 |

<표 3> Numeric type 실험 설정

### 3.3.2 Boolean type 트리에 대한 실험 설정

시장의 현상을 더 자세히 관찰하기 위해 실험을 진행한 Boolean type 트리에서는 지속성이 강한 패턴의 특성을 관찰하기 위해 트리의 높이와 패턴의 빈도수의 변화가 패턴의 성능에 어떤 영향을 주는지 관찰하고자 하였다. 트리의 높이는 초기해를 만들 때와 세대가 진화할 때 각각의 최대 높이를 init max height와 train max height로 나누어 설정하였다. 패턴의 빈도수에 대한 차이를 관찰하기 위해서 전 기간에 대해 패턴에 해당하는 종목수의 평균의 최소한의 크기를 변수로 설정하였으며 이 변수를 초기해를 만들 때, 세대가 진화할 때 각각에 대해 init minimum threshold of signals(init min signals)와 train

minimum threshold of signals(train min signals)로 나누어 설정하였다. N은 해집단의 크기이고 k는 tournament selection의 tournament size이다.

| 구분 | N | k | init max height | train max height | init min signals | train min signals |
|---|---|---|---|---|---|---|
| B-1 | 5000 | 50 | 8 | 16 | 1 | 1 |
| B-2 | 5000 | 50 | 4 | 8 | 1 | 1 |
| B-3 | 5000 | 50 | 4 | 8 | 5 | 10 |
| B-4 | 5000 | 50 | 2 | 4 | 1 | 1 |
| B-5 | 5000 | 50 | 2 | 4 | 5 | 10 |

<표 4> Boolean type 실험 설정

# 제 3 절 실험 결과 분석

### 4.3.1 동적 롱숏 포트폴리오 성능 평가

GP을 활용하여 얻은 동적 롱숏 포트폴리오의 성능을 평가하기 위해 시간 경과에 따른 모델의 안정성을 평가하는데 자주 사용되는 롤링분석을 활용하였다. 롤링 분석은 일반적으로 안정성과 예측 정확도를 평가하기 위해 과거 데이터에 대한 통계 모델을 백테스트 하는데 사용된다[21]. 전체 사용할 수 있는 데이터에서 train기간 3년 test기간 1년을 설정하여 이를 1년씩 롤링하여 11개의 기간 세트에 대해 실험을 진행했다. 각 기간에 대해 10번씩의 실험을 진행하였고 이를 통해 얻은 최적의 포트폴리오의 성과를 평균을 내어 비교하였다. 포트폴리오의 성과를 비교하기 위해 모든 종목들에 대해 동일 비중으로 매수하여 그 상태를 유지하는 BH 포트폴리오를 설정하여 비중 조절을 통해 얼마 많은 성과를 창출했는지를 확인하였다. 이때 GP를 통해 개발한 동적 롱숏 포트폴리오는 BH 포트폴리오와는 달리 비중을 지속적으로 조절하는 포트폴리오로 비중을 조절하는 과정에서 거래비용이 발생하게 된다. 이에 개발한 동적 롱숏 포트폴리오는 비중을 조절하는 대가로 발생하는 거래비용을 능가하는 수익을 창출해내야 유의미한 성능을 보일 수 있다. 본 실험에서 거래비용은 한국 주식시장에서 거래하는데 필요한 거래수수료와 거래세를 고려하여

0.3%로 설정하였다.

GP를 통해 탐색한 동적 롱숏 포트폴리오의 성능을 평가하기 위해 test 기간의 절대적인 성능을 확인하고 공간 탐색과정에서 overfitting의 정도를 확인하기 위해 train기간의 수익률과 test 수익률의 비율을 비교했다. 거래비용을 고려한 매매에서 모든 test기간에서 BH 포트폴리오의 수익률을 능가하는 것을 확인할 수 있고, 절대적인 수익률을 확인했을 때 10%이상의 수익률을 기록해낸 것을 확인할 수 있다. 하지만 11개의 기간 동안의 test 수익률의 평균/train 수익률의 평균의 평균은 0.66으로 overfitting을 막았다고 보기는 힘든 결과를 확인할 수 있다. test기간에 해당하는 2010년에서 2020년까지의 한국 주식시장은 박스권에서 장기간 횡보하는 흐름을 보였다는 것을 고려하였을 때, 롱숏 포트폴리오를 통해 꾸준한 수익을 내는 포트폴리오를 구성해냈다는 것은 긍정적인 결과로 볼 수 있다.

| Training 기간 | Testing 기간 | Train 수익률 평균 | Test 수익률 평균 | Test기간 BH 수익률 |
|---|---|---|---|---|
| 2007 - 2009 | 2010 | 21.1 | 18.2 | 7.1 |
| 2008 - 2010 | 2011 | 27.6 | 10.1 | −1.0 |
| 2009 - 2011 | 2012 | 31.3 | 15.3 | 9.1 |
| 2010 - 2012 | 2013 | 33.2 | 24.6 | 3.5 |
| 2011 - 2013 | 2014 | 25.4 | 20.2 | 10.7 |
| 2012 - 2014 | 2015 | 20.5 | 19.1 | 29.8 |
| 2013 - 2015 | 2016 | 29.2 | 15.1 | 8.0 |
| 2014 - 2016 | 2017 | 28.3 | 14.3 | 6.9 |
| 2015 - 2017 | 2018 | 32.6 | 15.6 | −4.3 |
| 2016 - 2018 | 2019 | 22.3 | 13.4 | 4.0 |
| 2017 - 2019 | 2020 | 23.1 | 20.4 | 31.3 |

<표 5> 동적 롱숏 포트폴리오 P&L결과

### 4.3.2 패턴의 지속성에 대한 관찰

유전프로그래밍을 통해 공간을 탐색하여 얻은 패턴이 장기적으로 유의미한 패턴인지를 확인하기 위해 train기간과 동일한 기간인 3년의 test기간을 두어 장기간 동안 유사한 성능과 경향성이 지속되는지를 확인하고자 했다. 이를 위해 단순히 test 기간 동안의 P&L을 비교하여 전략의 성능을 평가하는 것이 아닌 train 기간과 test기간의 P&L의

비율과 sharp ratio의 비율을 관찰하여 어떤 요인이 유의미한 패턴을 탐색하는데 도움을 주는지를 살펴보았다. 해당 비율은 train 기간에서 얻은 값을 test 기간에서 얻은 값으로 나누어 계산하였다. 이 비율은 1에 가까울수록 train과 유사한 성능을 냈다는 것을 의미하고 1을 넘는 수치는 train 기간의 성능을 능가하는 결과를 보인 것을 의미한다. 앞에서 제시된 5가지의 실험 세팅을 10번씩 수행하여 얻은 결과를 통해 어떻게 패턴을 탐색하는 것이 장기적인 패턴을 찾는데 유리한지를 확인해보았다.

우선 tree의 높이가 지나치게 커지면 train기간에서 찾은 패턴이 test기간에 거의 지속되지 못하는 것을 확인할 수 있었다. 깊은 트리를 통해 찾은 패턴이 지나치게 train 기간에만 적합한 패턴을 만들어내는 모습을 확인했다(B-1). tree height를 줄여주는 것만으로 보다 간단한 형태의 패턴을 형성하여 장기적으로 지속적인 경향성을 보이는 패턴을 발굴하는데 긍정적인 효과를 볼 수 있었다(B-2). 하지만 지나치게 낮출 경우 다양한 해를 형성할 수 없고, 이로 인해 다양한 공간을 탐색할 수 없을 수 있다는 것 또한 확인했다(B-4).

단순히 트리의 크기를 제한하는 것에서 나아가 매일 패턴에 해당하는 종목의 수(signal)의 발생 빈도에 대한 조건을 추가했을 때 지속성이 강한 패턴을 찾는데 매우 긍정적인 효과를 얻을 수 있다는 것을 확인했다. B-3의 실험 결과에서 B-1과 B-2의 결과에 비해 P&L과 sharp ratio 모두에서 개선이 일어난 것을 확인할 수 있다. 이를 통해 다양한 해를 표현할 수 있는 트리 공간에서 발생빈도에 대한 제약을 가지고 공간을 탐색할 경우 더욱더 유의미한 패턴을 찾을 수 다는 것을 확인할 수 있었다.

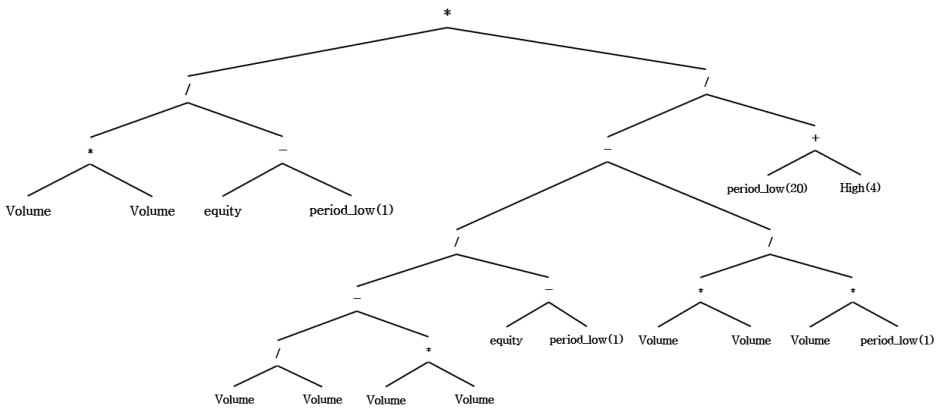
| 구분 | Best pattern test/train P&L ratio | Best pattern test/train sharp ratio | mean test/train P&L ratio | mean test/train sharp ratio |
|---|---|---|---|---|
| B-1 | 0.08 | 0.11 | 0.03 | 0.05 |
| B-2 | 0.58 | 0.85 | 0.26 | 0.35 |
| B-3 | 0.79 | 1.11 | 0.53 | 0.71 |
| B-4 | 0.23 | 0.22 | 0.07 | 0.04 |
| B-5 | 0.83 | 1.09 | 0.68 | 0.90 |

<표 6> P&L과 sharp ratio의 test/train 결과의 비율

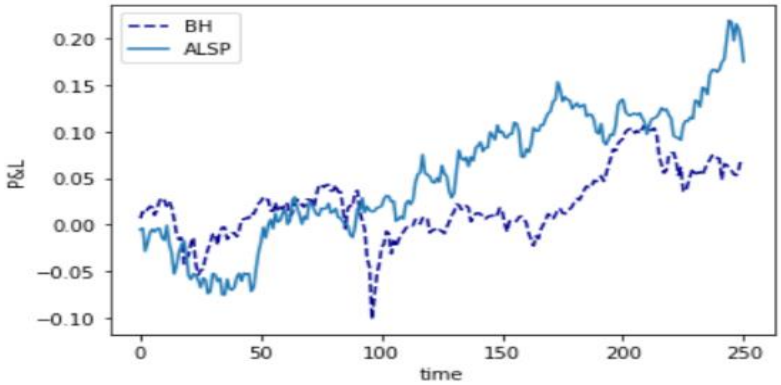
4.3.3 test기간 시뮬레이션을 통한 고찰

발견한 다양한 패턴 중에서 Numeric type 트리 중에 가장 성능이 좋았던 패턴에 대해 소개하고, Boolean type 트리 중에 test/train P&L 성능이 특히 좋았고 해석에 용이했던 패턴, 가장 좋지 못했던 패턴을 통해 주식 시장의 현상을 분석해보고자 한다.

첫번째 소개할 패턴은 Numeric type 트리를 통해 찾아낸 패턴으로 이며 <그림 5>을 통해 확인할 수 있다. 이 패턴은 매우 복잡해 보이지만 equity의 scale이 volume과 일반적인 price데이터에 비해 매우 크다는 것을 고려하여 그래프의 수식을 정리하면 대략적으로 $-period_H5 * volume^3 / equity^2$ 라는 수식을 얻을 수 있다. 이 수식을 통해 각 종목에 비중을 책정한다는 의미는 equity가 낮고 거래 많이 일어난 종목에 대해서는 상대적으로 매도 포지션을 잡고 equity가 높고 거래가 적게 일어난 종목에 대해서는 상대적으로 매수 포지션을 잡았을 때 수익을 얻을 가능성이 크다는 것을 보여준다. 수식의 값이 모두 음수이지만 비중을 책정하는 과정에서 평균을 0으로 맞춰주므로 수식의 결과값의 상대적인 위치를 비교해야 한다. 아래는 해당 패턴을 통해 2010년 test기간 동안 직접 포트폴리오를 운용했을 때의 결과이다. 시장 전반적인 흐름을 따르는 BH의 움직임과 크게 연동되지 않고 꾸준히 우상향하는 그래프를 확인할 수 있다.



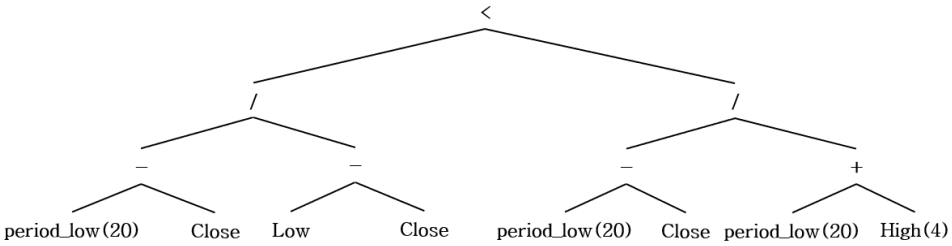

<그림 5> Numeric pattern graph



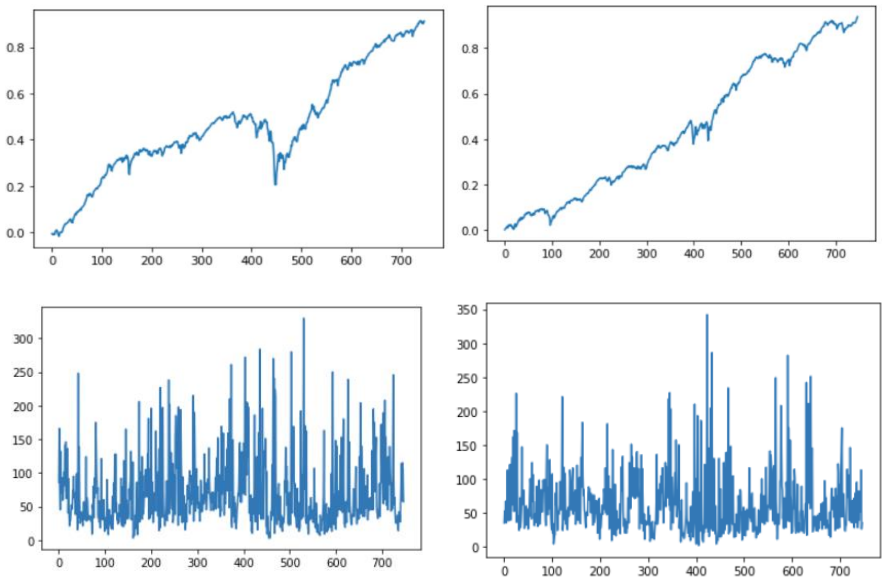
<그림 6> 동적 롱숏 포트폴리오 P&L 그래프

두번째로 소개할 패턴은 Boolean type 트리 패턴으로 <그림 7>를 통해 트리 구조를 확인할 수 있다. 이를 수식으로 간단히 표현하면 $(Close == Low)\ and\ (period\ low(20) < Close)$ 로 작성할 수 있고 이는 오늘 저가가 오늘의 종가이지만 그 종가가 20일 기간 동안의 저가는 이탈하지 않은 경우에 해당하는 종목을 매수한다는 의미이다. P&L을 확인해보면 train기간 28.59%과 test기간 27.82%의 수익률을 보이며 유의미한 수익을 가져다 주는 것을 확인할 수 있다. 시장에서 장 마지막까지 공포 속에서 투매가 발생했지만 그 종가가 20일 기간 동안의 저점을 이탈하지 않았다면 다음날 이를 회복할 가능성이 크다는 것을 이 패턴을 통해 확인할 수 있다. 본 패턴은 주식시장에서 이미 알려진 가격 캔들 패턴의 종류 중에 역망치형의 변형이라고도 볼 수 있다. 역망치형은 봉의 길이가 짧고 아래 꼬리가 없이 위꼬리만 존재하는 형태의 캔들이다. 이에 대한 실험의 일환으로, 국내 주식 패턴에 대한 실험을 진행한 메트릭 스튜디오[22]에서는 역망치형의 캔들을 그대로 사용하는 것보다 전일의 상황을 고려한 패턴을 사용하는 것이 효과적이라는 것을 실험을 통해 밝혔다. 이를 참고하여 메트릭 스튜디오에서 제시한 전일 길이 5% 이상의 음봉을 보여준 후 역망치형을 형성하는 패턴(pattern_m)과 본 연구에서 발견한 패턴(pattern_2)을 비교하고 추가적으로 두 패턴의 특징을 결합하여 구성한 패턴의 성능을 확인해보았다. Combined pattern은 전일 길이가 5% 이상의 음봉이 출현하고 20일 기간의 저점을 이탈하지 않았다는 두 상황을 모두 반영한 패턴으로 제작하였다. <그림 9>의 P&L 그래프를 확인해보았을 때 새롭게 발견한 패턴이 이전에 제시된 pattern_m에 비해서 저조한 성능을 보이는 것을 확인할 수 있다. 하지만 pattern_2의

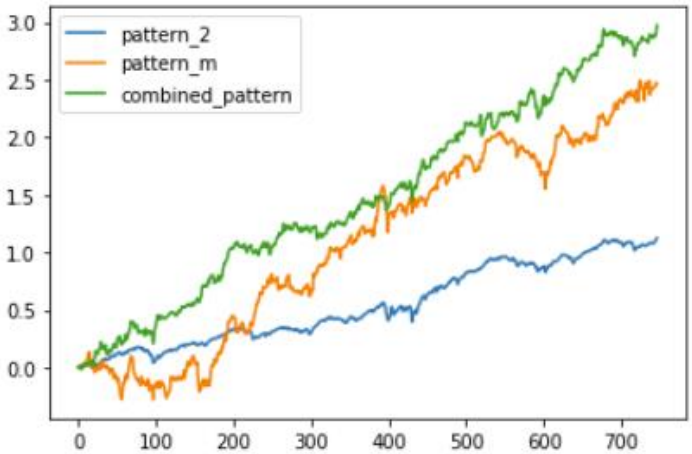
특징을 활용하여 결합한 combinded pattern에서 변동성은 낮아지고 수익률은 높아지는 것을 확인할 수 있다. 이를 통해 유전프로그래밍을 통해 찾은 전략을 직간접적으로 시장에서 지속적으로 반복되는 유의미한 패턴을 찾는데 활용할 수 있음을 알 수 있다.




<그림 7> Boolean type pattern 2 graph

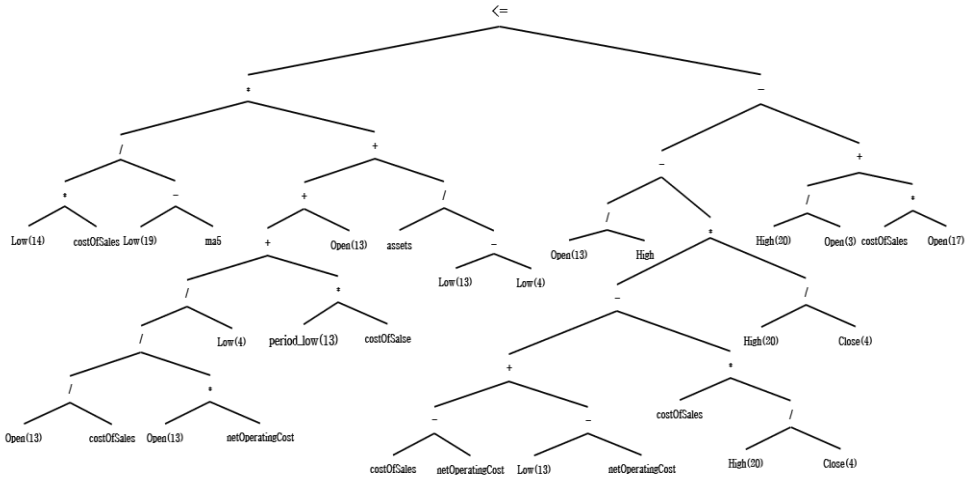


<그림 8> $(Close == Low)\ and\ (period\ low(20) < Close)$ 패턴의 P&L 그래프(위)와 날짜별 signal의 수에 대한 그래프(아래), 왼쪽이 train기간, 오른쪽이 test기간

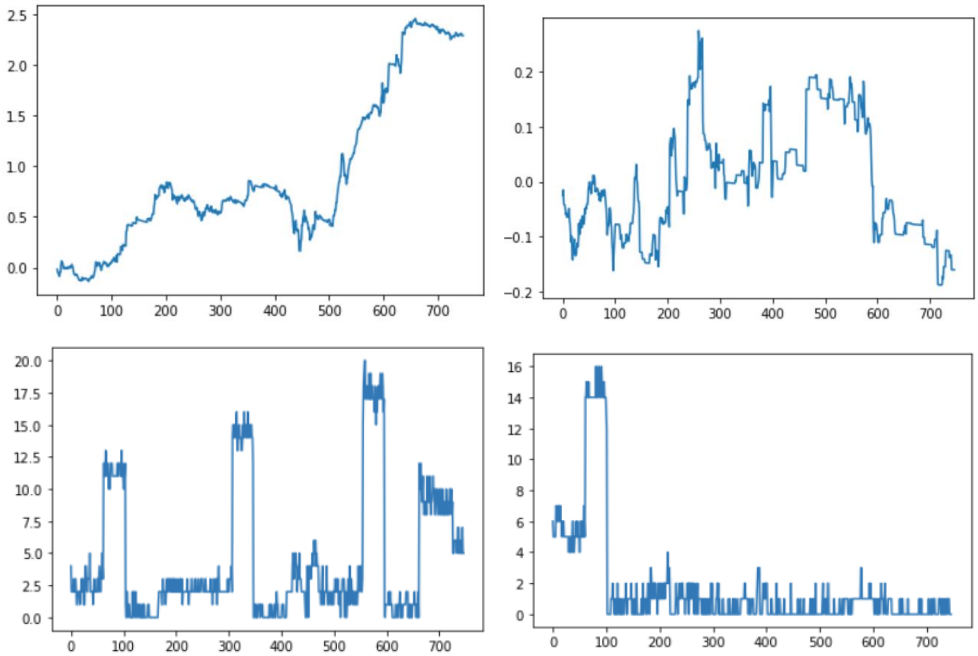



<그림 9> pattern의 특징을 활용한 새로운 pattern의 test기간에 대한 P&L 비교

세번째로 소개할 패턴은 Boolean type 트리 패턴으로 <그림 10>을 통해 확인할 수 있다. 이 패턴은 매우 복잡한 트리 구조로 그 의미는 이해하기 어렵지만 앞에서 소개한 패턴과 P&L그래프와 signal의 수의 분포를 통해 그 차이를 분석해보고자 한다. 두 패턴의 성능은 test 기간의 성능을 비교해 보았을 때 확연히 차이가 나는 것을 쉽게 확인할 수 있다. 이에 대한 분 것을 위해 우선 train기간의 누적 P&L 그래프를 비교해 보았을 때 오히려 P&L값은 77%로 매우 매력적인 패턴을 찾은 것처럼 보일 수 있다. 하지만 이 두 패턴의 큰 차이는 signal의 수의 분포에서 볼 수 있는데 첫번째 패턴의 경우 train기간의 signal 수의 분포를 보았을 때 signal이 많이 발생하는 시기가 한 기간에 모여 있지 않고 잦은 주기로 반복되는 것을 확인할 수 있고, 이러한 현상이 test기간에도 반복되는 것을 확인할 수 있다. 하지만 두번째 패턴의 경우 train기간에 200-300일 간격으로 많은 시그널이 발생되는 것을 확인할 수 있고 이러한 현상이 test기간 초기까지 이어졌지만 그 이후로 발생하지 않는 것을 확인할 수 있다. 이를 통해 패턴으로 인한 signal이 많이 발생하는 발생 빈도가 클수록 반복성이 강한 유의미한 패턴일 가능성이 크며 test 기간에도 지속될 가능성이 크다는 것을 유추해볼 수 있다.




<그림 10> Boolean type pattern 3 graph



<그림 11> 세번째 패턴의 P&L 그래프(위)와 날짜별 signal의 수에 대한 그래프(아래) (왼쪽이 train기간, 오른쪽이 test기간)

# 제 5 장 결론 및 제언

본 연구는 유전프로그래밍을 활용하여 종목들의 관계성을 고려한 동적 롱숏 포트폴리오 거래 시스템을 제안했다. 또한 유전프로그래밍의 유연한 표현형과 패턴을 해석하기 용이하다는 특징을 활용해 시장에서 강한 지속성을 지닌 패턴을 탐색하고 그 특징을 확인했다. 본 연구에서 제시된 동적 롱숏 포트폴리오 시스템은 개별 종목들의 절대적인 상황만을 고려하는 것이 아닌 상대적인 관계성을 고려하도록 설계되었다. 과거 데이터에서 최적의 포트폴리오에 대한 규칙을 추출하기 위해 트리 구조를 활용하였고 공간 탐색에 용이하기 위해 일반적인 수치연산자 외에 데이터 사이의 관계성을 더 깊이 탐색할 수 있는 연산자를 추가하였다. 이렇게 제안된 포트폴리오는 몇 가지 장점을 지닌다. 첫째로 포트폴리오의 구성을 결정하는 비중 조절 방식에서 종목들 사이의 관계에 기반한 정보를 추출해낼 수 있다. 둘째로 제안된 거래 시스템을 활용하여 투자자는 주가의 작은 변동으로부터 이익을 얻을 수 있다. 이 이점은 동적으로 비중을 조절하는 거래 방식에 연관된 것인데 이러한 거래 방식은 큰 거래비용을 야기하게 된다. 이를 고려하기 위해 공간을 탐색할 때 비중 조절에 따른 거래 비용을 평가함수에 반영하였다. 또한 거래 시스템의 안정성을 확인하기 위해 롤링분석을 활용하였고 모든 기간에서 거래비용을 능가하는 수익을 창출하는 포트폴리오를 구성할 수 있다는 것을 확인했다. 유전프로그래밍을 활용해 시장의 현상을 분석한 추가 실험에서는 트리의 크기와 signal 발생 빈도에 대한 제약을 통해 signal 빈도가 잦은 패턴을 찾았을 때 그 지속성이 더욱 강하다는 것을 확인할 수 있었다.

본 연구에서 발견한 패턴들은 그 자체로서 통계적인 우위에 있고 거래비용을 능가하는 수익성을 창출할 수 있는 것은 사실이지만 이것을 투자전략에 활용하기 위해서는 보다 고려해야할 요소가 많다. 특히 매일매일 거래 종목을 선정하거나 리밸런싱을 진행하는 본 연구의 설정은 시장에서 거래가 적게 일어나는 종목의 경우 설정한 기본적인 거래비용에 추가적으로 적은 거래 환경에 의해 원하는 가격보다 높은 가격에 매수하게 되는 슬리피지가 지나치게 커질 수 있다. 이러한 상황을 고려하기 위해 포트폴리오의 비중을 설정할 때 거래량을 고려하도록 평가함수를 고안하거나 포트폴리오 종목군 자체를 거래가 많이 일어나는 함수들로 제한하여 공간을 탐색하는 실험이 추가적으로 필요할 것으로 생각한다. 본 연구에서 제시한 동적 롱숏 포트폴리오 거래 시스템에 이러한 요소들이 추가된다면 투자자들에게 더욱 유용한 포트폴리오 구성방식을 제안할 수 있을 것으로 보인다.

# 참고 문헌


[1] Yu, Pengfei, and Xuesong Yan. "Stock price prediction based on deep neural networks." Neural Computing and Applications 32.6 (2020): 1609-1628.

[2] J. Branke, B. Scheckenbach, M. Stein, K. Deb, H. Schmeck, Portfolio optimization with an envelope-based multi-objective evolutionary algorithm, Eur. J. Oper. Res. 199 (3) (2009) 684–693.

[3] M.A.H. Dempster, C.M. Jones, A real-time adaptive trading system using genetic programming, Quant. Finance 1 (2001) 397–413.

[4] F.D. Freitas, A.F. DeSouza, A.R. deAlmeida, Prediction-based portfolio optimization model using neural networks, Neurocomputing 72 (2009) 2155–2170.

[5] Koza, John R. Genetic programming II. Vol. 17. Cambridge: MIT press, 1994.

[6] Hwang, T.-M., Oh, H., Choung, Y.-K., Oh, S., Jeon, M., Kim, J. H., et al. Prediction of membrane fouling in the pilot-scale microfiltration system using genetic programming. Desalination, 247 (2009): 285–294

[7] R.J. Bauer, Genetic Algorithms and Investment, Wiley, New York, 1994.

[8] F. Allen, R. Karjalainen, Using genetic algorithms to find technical trading rules,

[9] H. Etemadi, A.A. Anvary Rostamy, H. Farajzadeh Dehkordi, A genetic programming model for bankruptcy prediction: empirical evidence from Iran, Expert Syst. Appl. 36 (2) (2009) 3199–3207.

[10] Y.-S. Lee, L.-I. Tong, Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming, Knowledge Based Syst. 24 (2011) 66–72.

[11] C. Neely, Risk-adjusted, ex ante, optimal, technical trading

rules in equity markets, Int. Rev. Econ. Finance 12 (1) (2003) 69–87.
[12] J. Potvin, P. Soriano, M. Vall'ee, Generating trading rules on the stock markets with genetic programming, Comput. Oper. Res. 31 (2004) 1033–1047.
[13] C. Fyfe, J.P. Marney, H. Tarbert, Risk adjusted returns from technical trading: a genetic programming approach, Appl. Financ. Econ. 15 (2005) 1073–1077.
[14] A. Esfahanipour, S. Mousavi, A genetic programming model to generate risk adjusted technical trading rules in stock markets, Expert Syst. Appl. 38 (7) (2011) 8438–8445.
[15] Y. Chen, S. Mabu, K. Shimada, K. Hirasawa, Trading rules on stock markets using genetic network programming with sarsa learning, J. Adv. Comput. Intell. Intell. Inform. 12 (4) (2008) 383–392. J. Financ. Econ. 51 (1999) 245–271.
[16] S. Mousavi, A. Esfahanipour, M. H. F. Zarandi, A novel approach to dynamic portfolio trading system using multitree genetic programming, Knowledge-Based Systems, Volume 66, August 2014, Pages 68-81
[17] 장병탁. 유전 알고리즘 이론 및 응용 전자공학회지, Vol.22 (11), (1995) : 1290-1300
[18] Bae, H., Jeon, T. R., Kim, S., Kim, H. S., Kim, D., Han, S. S., et al. Optimization of silicon solar cell fabrication based on neural network and genetic programming modeling. Soft Computing, 14(2) (2010): 161–169.
[19] Etemadi, H., Rostamy, A. A. A., & Dehkordi, H. F. A genetic programming model for bankruptcy prediction: empirical evidence from Iran. Expert Systems with Applications, 36 (2009): 3199–3207.
[20] R. Poli, W.B. Langdon, N.F. McPhee, A Field Guide to Genetic Programming, 2008
[21] Eric Zivot and Jiahui Wang. Rolling analysis of time series.Modeling Financial Time Serieswith S-PLUSR©, pages 313–360, 2006
[22] Moon. 문병로 교수의 메트릭 스튜디오, pages 313–360, 2016

Abstract

# Dynamic Long Short Portfolio Optimization and Stock Market Pattern Search Using Genetic Programming

Kim Chang Gyoum
Computer Science and Engineering
The Graduate School
Seoul National University

A dynamic portfolio trading system is used to allocate capital among stocks dynamically over time in a manner that maximizes the return of the portfolio and minimizes the risk of the portfolio. Genetic programming, an artificial intelligence technique, has been successfully used in the field of finance, especially in the task of making short-term predictions about financial markets. In this paper, genetic programming is used to construct an optimal portfolio by predicting the flow of stock market prices over time. The approach proposed in this study is designed to determine the weight of the portfolio by considering the relative situation of the stocks in the portfolio when composing the weight of the portfolio. In order to consider the relative relationship of items within the tree structure of genetic programs, we searched the space by adding operators that can calculate rank and zscore as well as general numerical operators. In addition, the system was developed by considering the transaction costs of transactions that greatly affect the profitability of the

dynamic portfolio system. By comparing the portfolio considering transaction costs with the portfolio held after purchase without weighting, we checked how meaningful the profit was actually created by weighting. In order to confirm the significance of the developed dynamic long-short portfolio system, we tested whether it was possible to build stable portfolios in various periods by using the rolling method. Through the results of the experiment, it was confirmed that steady profits were generated through weight adjustment in various periods. In addition, an experiment was conducted to analyze recurring phenomena in the market by utilizing the characteristics of genetic programming, which allows you to know the operating principle of the model in more detail compared to general data-based models. To this end, an experiment was conducted by constructing a Boolean type tree that is easy to interpret, and the ratio between the performance during the test period and the performance during the train period was checked to determine the continuity of the pattern. Experimental results confirm that the frequency of a pattern found in the market has a strong influence on the persistence of that pattern.

Keywords : Genetic programming, Machine learning, Stock market pattern search
Student Number : 2019-24908