https://doi.org/10.7840/kics.2023.48.6.755

심층 강화학습 기반 조세 및 경제 활동 에이전트 정책 최적화 시뮬레이션 환경 분석 및 실험

허 주 성•, 최 요 한*, 석 영 준*, 유 태 완**, 이 연 희**, 한 연 희°

Deep Reinforcement Learning-Based Tax and Economic Agents Policy Optimization Simulation Environment Analysis and Experiment

Joo-Seong Heo•, Yo-Han Choi*, Yeong-Jun Seok*,
Taewan You**, Yeonhee Lee**, Youn-Hee Han°

요 약

4차 산업 혁명으로 AI가 사회 전반에 걸쳐 상용화되고 꾸준히 발전하고 있지만, 경제 분야는 여전히 데이터 부족, 다양한 환경, 변수 등으로 AI 적용이 어렵다. 실제 사회 경제적 문제를 해결하기 위해선 경제 주체 간 다양한 환경과 상호작용 요인들을 확인하며 경제 활동 및 정책 수립 과정을 설계하고 테스트해야 하지만, 경제 관련 데이터가 부족하고, 실제 정책을 실험하기 위한 환경 구성이 어렵다. 본 논문에서는 Salesforce 팀의 AI 기반 경제 시뮬레이션 환경인 AI Economist를 활용하여 심층강화학습 기반 조세 및 경제 활동 에이전트 정책 최적화 실험 및 분석을 진행하였다.

키워드 : 조세 정책 최적화, 경제 시뮬레이션 분석, 강화학습
Key Words : Tax Policy Optimization, Economic Simulation Analaysis, Reinforcement

ABSTRACT

With the fourth industrial revolution, AI has been commercialized and continuously developed throughout society. However, the economic sector still faces challenges in applying AI due to a lack of data, various environments, and variables. To address real economic problems, it is necessary to identify and test various environmental and interactive factors among economic entities in the process of designing and testing economic activities and policies. However, economic data is lacking and it is difficult to create an environment for experimenting with actual policies. In this paper, we utilize the AI Economist, an AI-based economic simulation environment developed by the Salesforce team, to conduct experiments and analysis on tax and economic activity agent policy optimization based on deep reinforcement learning

※ 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구(No.2022-0-00857, AI·데이터 기반 재정·경제 디지털트윈 플랫폼 개발)이며, 또한 2023년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 지원을 받아 수행된 연구(No. NRF-2023R1A2C1003143)임.
• First Author : THINKONWEB, js.heo@thinkonweb.com, 정회원
° Corresponding Author : Future Convergence Engineering, Korea University of Technology and Education, yhhan@koreatech.ac.kr, 종신회원
* Future Convergence Engineering, Korea University of Technology and Education, yowief@koreatech.ac.kr; dsb04163@koreatech.ac.kr, 학생회원
** Electronics and Telecommunications Research Institute, twyou@etri.re.kr; leeyh@etri.re.kr
논문번호 : KICS202303-047-0-SE.R1, Received March 14, 2023; Revised April 11, 2023; Accepted April 12, 2023

Ⅰ. 서 론

전통적인 경제학에서 경제 정책은 정부와 다양한 경제 주체들 간의 상호작용을 고려하여 많은 실험과 공식, 계산 등의 복잡한 절차를 통해 연구, 수립된다. 예를 들어, 경제 성장을 위한 조세 정책 수립을 한다면, 기본적으로 국가 재정 운용 계획에 따른 재정 운용 전략뿐만 아니라 세목별 조세 정책 방향, 소득 구간별 조세 부담 형평성 등 심지어 경제, 복지, 인구 구조, 통일, 환경까지 고려해야 할 사항이 많다. 실제로도 전례 없는 코로나 19 상황으로 인해 우리나라뿐만 아니라 세계 경제가 급변하고 있는 만큼 효율적인 경제 정책을 수립하는 것은 어려운 일이다.

4차 산업 혁명으로 AI가 사회 전반에 걸쳐 상용화되고 꾸준히 발전하고 있지만, 경제 분야는 여전히 데이터 부족, 다양한 환경, 변수 등으로 AI 적용이 어렵다. 실제 사회 경제적 문제를 해결하기 위해선 경제 주체 간 다양한 환경과 상호작용 요인들을 확인하며 경제 활동 및 정책 수립 과정을 설계하고 테스트해야 하지만, 경제 관련 데이터가 부족하고, 실제 정책을 실험하기 위한 환경 구성이 어렵기 때문이다.

예를 들어, 실제 경제 활동 인구는 개인마다 직업, 숙련도 등이 다르고 그에 따라 받는 소득 또한 다르다. 따라서, 자연스레 소득에 따른 부의 격차가 발생하게 된다. 이때, 정부의 소득 구간별 조세 정책은 일반적으로 소득이 높은 경제 활동 인구에게 세금을 걷어 소득이 낮은 구간의 경제 활동 인구에게 나눠 줌으로써 부의 재분배를 통해 불평등 해소에 중요한 역할을 한다. 하지만, 불평등 해소를 위해 무리하게 세율을 올리게 되면 오히려 능력이 높은 즉, 소득 구간이 높은 곳에 있는 경제 활동 인구들이 의욕을 잃고, 생산성이 감소하게 되는 문제가 발생할 수 있으며, 반대로 생산성을 걱정하여 세율을 낮게 책정하면 생산성은 올라가지만 여전히 불평등 격차는 다시 심해지게 된다.

본 논문에서는 이러한 다양한 경제 활동, 정책 수립에 따른 상호 요인 분석 등을 실험하기 위하여 Salesforce 팀에서 개발한 AI Economist 환경을 사용하여 경제 활동 에이전트들을 학습하고 조세 정책을 최적화하기 위한 실험을 진행하였다.

Ⅱ. 관련 연구

일반적으로 정부의 조세 정책은 부의 재분배를 통해 불평등 해소에 중요한 역할을 한다. 최적의 조세 정책을 수립하기 위한 핵심 과제는 과도한 세금 징수로 인한 생산성 감소 즉, 노동자들의 일할 동기에 영향을 미칠 수 있어 부의 재분배를 통한 평등과 생산성 사이의 상충관계를 적절히 조율하는 것으로 불평등 해소를 위해 세율을 올리게 되면 생산성이 감소하고 반대로 세율이 낮아지면 생산성은 올라가지만 불평등 격차는 심해지게 된다[1,2].

조세 정책을 위해선 경제학에서 고려해야 할 다양한 환경과 이론이 있지만, 기본적인 정책은 소득의 양에 따른 소득 구간별 세금 징수로 이루어진다. 2018년 미국의 소득 구간과 세율은 소득이 높아질수록 점점 증가하는 형태로 고소득자가 세금을 더욱 많이 납부하는 계급별 세율 구조를 나타내는 반면 Saez의 조세 정책은 오히려 높은 소득에 따라 세율이 감소하는 구조를 나타낸다. 이에 따라 개인의 행동 또한 달라지게 된다. 즉, 소득 구간별 납부 해야하는 세금의 양에 따라 생산성이 차이나게 된다.

Salesforce는 고객 관계 관리를 위한 플랫폼 서비스를 제공하는 IT 기업으로, 최근 Salesforce Research 팀은 주요 글로벌 이슈 중 하나인 경제적 불평등 개선을 목적으로 AI 기반 경제 정책 설계에 도움을 줄 수 있는 프레임워크를 개발하였다[3].

AI Economist는 경제 활동 에이전트와 정부 에이전트를 통해 강화학습 알고리즘을 적용하여 학습한 조세 정책이 실제 경제적 불평등을 개선할 수 있는지 평가하기 위한 시뮬레이션 환경을 제공하며, 실제 AI Economist가 학습한 조세 정책은 기존의 잘 알려진 조세 정책과 비교했을 때 개선된 성능을 보여준다[4]. 또한, 강화학습을 사용하여 동적인 경제에서 경제 모델의 가정이 아닌 관찰된 데이터를 활용하여 주어진 환경에서 불평등과 생산성의 균형을 위한 최적의 조세 정책을 시뮬레이션할 수 있다[5].

Ⅲ. 경제 시뮬레이션 환경

AI Ecomist에선 경제 시뮬레이션을 위한 25X25 크기의 2차원의 그리드 공간의 Gather and Build 환경을 제공한다. 환경에서 Resident(경제 활동 에이전트)들은 이동하며 자원인 돌과 나무를 수집하고, 자원을 활용하여 집을 지어 코인을 얻거나, 다른 에이전트 간의 거래를 통해 코인으로 교환을 하는 다양한 경제 활동을 할 수 있다. 또한, Planner(정부 에이전트)는 생산성 향상과 불평등 해소의 균형을 맞추기 위해 소득 구간별 세율을 조정할 수 있다. 본 논문에서는 그림 1과 같이 AI Economist 환경을 분석하고, Unity를

활용하여 3D 기반 동적 모니터링을 위한 시각화 모듈을 개발하였다[6,7].

3.1 시나리오

초기 Resident들은 서로 다른 스킬 레벨을 가진 채 4구역으로 분리된 공간에 각자 위치한다. 맵 전체 임의의 공간에서 자원인 돌과 나무가 생성되며, Resident는 자원을 수집 혹은 거래를 통해 수집한 자원을 소모하여 집을 지을 수 있다. 이때, 돌과 나무 하나를 사용하여 집 한 채를 짓고 보상으로 코인을 받으며, Resident별 스킬 레벨에 따라 자원의 수집량과 집을 짓고 받는 코인은 다르게 지급된다. 하나의 에피소드는 총 1000 time-step으로 진행되며, Planner는 100 time-step마다 소득 구간별 세율을 조정하여 세금을 걷고, 재분배를 통해 사회복지 실현을 위해 행동한다.

그림 1. Unity 기반 경제 시뮬레이션 환경
Fig. 1. Unity-based Economic Simulation Environment

3.2 행동 및 관찰 정보

본 절에서는 경제 시뮬레이션 환경 구성을 위해 설계된 환경의 행동과 관찰 정보에 대해 설명한다. 시뮬레이션 환경은 Resident와 Planner 에이전트가 각각 다른 목표를 갖고 학습을 하는 2-level 강화학습으로 각 에이전트의 행동과 관찰 정보가 다르다.

3.2.1 Resident 행동 및 관찰 정보

Resident의 행동은 상하좌우 이동을 위한 행동 4개와 집 짓기 1개, 거래를 위한 행동 44개 마지막으로 아무 행동도 하지 않는 것까지 총 50개로 스텝별 하나의 행동을 선택할 수 있다. 자원의 수집은 돌과 나무 위로 Resident가 위치하면 자동으로 수집되며, 표 1은 Resident의 행동 분류를 나타낸다.

Resident의 관찰 정보는 총 4가지를 포함한다. 첫 번째는 Resident의 주변 정보를 관리하며, 맵, 인덱스 맵, Resident의 위치, 시간 정보 등을 포함한다. 이때, 맵은 전체 맵에서 각 Resident를 중심으로 11X11 크기만큼 총 7개의 채널을 가진다. 각각의 채널은 자신 주변의 Object의 유무를 나타낸다. 이때, 각 채널의 값은 0 또는 1을 가지며, 0은 없음, 1은 있음을 의미한다. 그림 2는 전체 맵에서 11x11 크기만큼의 특정 맵 데이터를 의미한다. 정 가운데 Resident를 중심으로 값을 1로 표시한다.

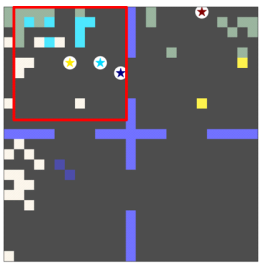
표 1. Resident의 행동 분류
Table 1. Resident's Behavioral Classification

행 동		개 수
이동		4
집 짓기		1
거래	Buy / Sell	2
	Wood / Stone	2
	Coin(0 ~ 10)	11
	거래 행동 합계	44 (=2*2*11)
건너뛰기		1
Resident 행동 합계		**50**

두 번째는 자원별 거래 시세와 내역, 거래된 매수, 매도 호가를 관리한다. Resident들은 수집한 자원을 소모하여 집을 지을 수도 있지만, 거래소에서 자원을 0~10 사이의 가격으로 거래할 수 있다.

세 번째는 Resident가 보유하고 있는 코인과 자원의 양을 관리하는 자산 그리고 현재의 세율, 마지막 소득 등 세율과 세금 징수에 관련된 정보를 관리한다.

마지막으로, 수집, 집 짓기에 사용될 스킬 레벨 정보를 관리하고 있다. 집 짓기에 사용되는 Build Skill은 스킬 레벨이 높을수록 집을 지을 때 얻는 코인의 양도 높아지며, 자원 수집에 사용되는 Collect Skill은 스킬 레벨이 높을수록 자원을 수집했을 때, 추가로 자원을 얻을 확률이 높아진다. 표 2는 Resident별 관찰 정보 분류를 나타낸다.



```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

그림 2. Resident 관찰 정보 예시
Fig. 2. Example of Resident Observation

표 2. Resident의 관찰정보 분류
Table 2. Resident's Observation Classification

구 분	항 목
맵	수집 가능한 자원(돌, 나무)의 위치
	집, 물의 위치
	자원의 생성 위치
	맵의 경계
인덱스 맵	Resident별 집의 위치
	Resident별 위치
위치	전체 맵에서 Resident 위치
시간	에피소드 내 현재 시간
거래소	자원의 현재 시세
	가격별 자원의 거래 횟수
	각 자원에 대한 Resident별 매도 호가
	각 자원에 대한 Resident별 매수 호가
자산	현재의 세율, 한계 세율, 세금 징수일 등
	Resident별 보유하고 있는 자원 및 코인
스킬	Build, Collect 스킬 레벨

3.2.2 Planner 행동 및 관찰 정보

Planner는 7X22개의 행동이 있다. 사회복지를 실현하기 위한 조세 정책을 위한 행동으로 총 7개의 소득 구간과 각 구간별 세율을 위한 22개의 행동으로 나뉜다. (0.05 단위로 0부터 100을 나누기 위한 21개와 아무 행동도 하지 않는 건너뛰기 1개를 포함)

7개의 소득 구간은 2020 US Federal 미국의 소득 구간을 그대로 사용하며, 실제 소득(달러)과 시뮬레이션 환경에서 소득(coin)을 맞추기 위해 환율을 적용하여 사용한다. 표 3은 소득 구간별 달러와 코인을 나타낸다.

Planner의 관찰 정보는 Resident와 달리 부분이 아닌 전체 맵에 대한 정보를 관리한다. Resident와 비슷한 정보를 관리하고 있지만, Planner 입장에서 불필요한 부분을 제외한 전체 정보를 관리하고 있다. 즉, 전체 맵에서 Resident별 스킬 레벨 정보를 제외한 Resident의 위치, 집의 개수, 거래 정보 등을 포함한다.

표 3. 소득 구간별 달러와 코인
Table 3. Dollars and Coins by Tax Bracket

구간	1	2	3	4
달러	0	9700	39475	84200
코인	0	97	39	84
구간	5	6	7	
달러	160725	204100	510300	
코인	160	204	510	

3.3 보상

본 논문의 시뮬레이션 환경은 강화학습을 위한 두 종류의 에이전트가 있다. 첫 번째는 자원을 캐거나 집을 짓고 거래를 통해 자산을 최대로 늘리는 것을 목표로 하는 Resident이며, 두 번째는 생산성과 불평등을 조율하여 사회복지를 최대화하는 것이 목표인 Planner로 2단계의 학습 과정이 필요하며, 목표가 다른만큼 에이전트별 보상 또한 다르다.

3.3.1 Resident 보상

Resident의 보상은 단순히 자산(Coin)을 늘리는 것이 아니라 별도의 수식을 적용한 효용함수를 적용한 유틸리티를 활용한다. 현실을 잘 반영하기 위해 경제학 개념을 적용하였으며, 한계효용체감 법칙을 적용하여 효용함수를 정의하였다.

이때, 한계효용체감의 법칙이란 어떠한 재화를 소비함으로써 얻는 만족감을 수치로 나타내는 것으로 재화를 더욱 소비할 경우 발생하는 추가적인 만족감은 감소한다는 것을 말한다. 즉, 배가 고플 때, 밥을 먹는 것에 대한 만족감과 밥을 어느정도 먹고 난 뒤의 만족감은 다르다는 것으로 자산이 적을 때 Coin을 얻는 것과 자산이 많을 때 Coin을 얻는 것에 대한 차이를 고려하여 현실을 반영하였다.

효용함수는 수식 1을 사용하며, 각 Resident의 총 코인의 양을 crra 효용함수를 통해 구한 뒤 노동의 양을 뺀 것으로 소득 대비 노동의 양을 적용하여 Resident는 노동 대비 자산의 최대화를 목표로 한다.

$$u_{i,t} = crra(coin_{i,t}) - l_{i,t} \tag{1}$$

$$crra(x) = \frac{x^{1-\eta}-1}{1-\eta},\ \eta > 0\ (Default: 0.23) \tag{2}$$

수식 1에서 i는 Resident, t는 time-step, x^c는 자산, l은 노동을 의미하며, 수식 2의 crra는 Constant Relative Risk Aversion로 경제학에서의 효용함수를 나타낸다.

표 4는 Resident들이 행동을 할 때 각 행동마다 누

표 4. Resident 행동별 노동의 양
Table 4. Amount of Labor by Resident Behavior

행동 구분	노동의 양
이동	1.0
자원 수집	1.0
집 짓기	10.0
자원 거래	0.25

적되는 노동의 양을 수치화한 것으로 이동, 자원의 수집에 1씩 노동이 발생하며, 집을 지을 때는 10, 자원을 사고파는 거래 행동에는 0.25의 노동이 발생 된다.

3.3.2 Planner 보상

Planner의 학습 목표는 사회복지를 최대화 하는 것으로 사회복지 함수를 보상으로 활용한다. 이때, 사회복지함수는 다양한 것들이 있지만 본 논문에서는 생산성과 불평등 지수를 곱한 것을 사회복지함수로 활용하였다.

$$swf_t = eq(coin_t) \bullet \mathrm{Prod}(coin_t)$$
$$eq(coin_t) = 1 - gini(x^c), \mathrm{Prod}(coin_{i,t}) = \sum_{i=1}^{N} coin_{i,t} \quad (3)$$

수식 3은 Planner의 보상인 사회복지함수를 나타낸 것으로 불평등 지수는 1에서 전체 코인에 대한 지니 계수를 뺀 것을 활용하며, 생산성은 전체 코인의 양을 활용한다. 지니 계수는 소득의 불평등 정도를 나타내는 대표적인 소득분배지표로 0과 1사이의 값을 가지며, 소득이 완전 평등한 경우 0, 완전 불평등한 경우 1의 값을 가진다. 인구 누적비율과 해당 소득 누적 비율을 그래프화 하여 로렌츠 곡선을 정의한 뒤, 로렌츠 곡선을 활용하여 지니 계수를 구할 수 있다.

Planner는 이러한 사회복지 함수를 활용하여 사회적 불평등을 개선하고 생산성을 최대화하기 위한 것을 목표로 학습한다. 이때, 생산성과 불평등지수는 서로 트레이드오프 관계에 있기 때문에, 적절하게 세율을 조정하여 이 둘을 최적화하는 것이 목표이다.

Ⅳ. 실 험

본 논문에서는 다양한 에이전트들과 경제 활동에 대한 상호작용을 테스트하기 위해 2단계 학습을 진행하였다. 1단계 학습은 Resident만 학습하는 것으로 정부의 별도 조세 정책이 없는 시나리오에서 Resident들이 이동, 자원 수집, 거래, 집 짓기 등의 행동으로 개인 보상을 최대화하는 것을 목표로 학습한다.

학습에 사용한 강화학습 알고리즘은 데이터를 효과적으로 사용하고, 여러 번의 업데이트를 위해 PPO(Proximal Policy Optimization Algorithm)를 사용하였다[8]. PPO 알고리즘은 2017년 OpenAI에서 개발된 강화학습 기법으로 대표적인 정책 경사 기반의 강화학습 기법이다. 기존 정책 경사 기반 기법들의 학습 불안정성, 속도 및 성능 저하 등을 개선하기 위하여 정책 갱신 전후의 비율을 클리핑하여 정책 갱신의 양을 간접적으로 제한하는 방식을 활용한다.

2단계 학습은 1단계에 이어 진행되며, Planner의 조세 정책을 적용하여 세율에 따른 Resident들의 행동 양식 변화와 각 Resident들의 행동에 따른 소득, 불평등 지수를 토대로 사회복지 함수를 최대화하기 위한 Planner의 학습이 함께 진행된다.

1번의 에피소드는 총1000 time-step으로 이루어져 있으며, Planner는 100 time-step마다 한 번씩 세금을 걷게 된다. 실험은 AI Economist의 소스 코드를 내재화하여 진행했으며, python 버전 3.7, tensorflow 버전 1.14, ray[rllib] 버전 0.8.4 등의 환경에서 실험을 진행했다. 실험 결과를 모니터링 및 시각화하기 위하여 WanDB와Unity(유니티)의 3D 시각화 시뮬레이션 환경을 구성하여 실험 결과를 시각화하여 평가하였다.

4.1 1단계 학습

1단계 학습은 Resident에 대한 정책 최적화 실험으로 소득 구간별 별도의 조세 정책 학습이 없는 환경에서 진행되었으며, Resident의 다양한 행동 양식을 비교 분석하기 위해 총 세가지 시나리오에서 학습을 진행하였다. 세금이 없기 때문에 Planner는 학습하지 않는다.

첫 번째는 Free Market 즉, 자유시장으로 세금이 전혀 없고 Resident들이 오로지 본인의 Utility를 최대화 하기 위해 학습하는 시나리오이다.

두 번째는 Communism 시나리오로 자유시장과 반대로 세율이 100% 즉, 주기적으로 세금으로 모두를 걷어 동등한 양으로 분배한다.

마지막은 Dystopia로 세율은 0%지만, Resident들이 개인의 유틸리티를 최대화하는 것을 목표로 일하지 않고 사회복지를 최대화하기 위해 일을 하는 시나리오이다.

각 시나리오별 학습 시간은 12시간 정도 소요되었으며, 50,000번 에피소드를 학습하였고, Resident들이 세금이 없는 시나리오에서 개인의 유틸리티를 최대화하기 위해 학습하며, 각자 다른 행동 양식을 보이는 것을 확인할 수 있었다.

표 5는 자유시장 환경에서 Resident별 스킬 레벨과 지은 집의 개수를 나타낸다.

학습 결과에서 알 수 있듯이 스킬 레벨이 낮을수록 집을 지었을 때 얻을 수 있는 코인의 양이 적어지므로, 유틸리티 즉, 노동 대비 얻을 수 있는 코인의 양이 적다는 것을 Resident들이 깨닫고 스킬 레벨이 상대적

표 5. Resident별 스킬 레벨과 집의 개수 비교
Table 5. Comparison of Skill level by Resident and Number of Houses

구분	스킬 레벨	집의 개수
Resident 1	11.3	1
Resident 2	13.2	0
Resident 3	16.4	16
Resident 4	22.5	120

으로 높은 3번과 4번 Resident들은 집을 짓지만, 1번과 2번 Resident들은 집을 짓지 않는 것을 알 수 있다.

그림 3은 Resident의 스킬 레벨별 유틸리티를 나타낸 것으로 x축은 time-step, y축은 유틸리티 값을 나타낸다. 스킬 레벨이 높은 Resident일수록 유틸리티가 높았고, 각 Resident들이 자원 수집, 거래, 집 짓기를 통해 개인의 자산을 최대화하기 위해 학습되었으며 1번, 2번 Resident의 경우 상대적으로 3번 4번 Resident보다 스킬 레벨이 낮아 유틸리티가 낮은데, 집을 거의 짓지 않은 것에 비해 유틸리티가 있는 것은 집을 지을 때의 노동이 양이 상대적으로 거래보다 높기 때문에 집을 짓지 않고 거래를 통하여 자산을 얻은 것을 알 수 있다.

그림 4는 time-step별 학습 결과를 시각화한 것으로 초록색은 나무, 하얀색은 돌, 파랑, 하늘, 노랑, 빨강별은 Resident와 각 Resident 색의 사각형은 집을 의미한다. 마찬가지로 스킬 레벨이 높을수록 더 많은 집을 짓고 상대적으로 낮은 Resident는 집을 짓는 것보단 자원 수집과 거래를 통해 유틸리티를 높이는 쪽으로 학습되는 것을 확인하였다.

두 번째 Communism 시나리오는 Resident들이 자산을 벌어도 주기적으로 전부 거두어 다시 균등하게 재분배를 해주기 때문에 상대적으로 자유시장보다 생

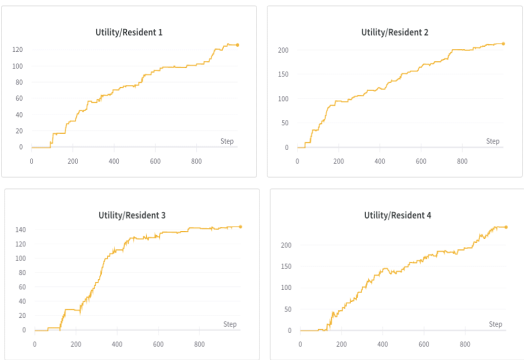


그림 3. Resident별 유틸리티 그래프
Fig. 3. Utility Graph by Resident

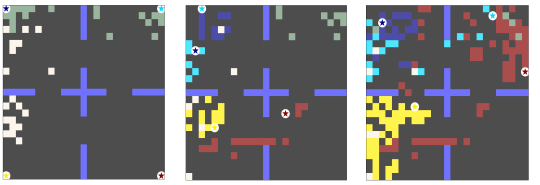


그림 4. time-step별 학습 결과(왼쪽부터 step-0, step-300, step-999)
Fig. 4. Learning results by time-step (step-0, step-300, step-999 from the left)

산 및 거래 활동이 적었다. 스킬 레벨이 높은 Resident는 대체적으로 활발하게 집을 짓지만, 자유시장 시나리오에 비해 아주 열심히 짓지 않고 최소한의 움직임으로 집을 짓는 행동 양식을 보였다. 또한, 사회복지 지수와 생산성, 불평등 지수 모두가 낮음을 알 수 있었다.

마지막 Dystopia 시나리오는 Resident가 사회복지를 최대화하기 위해 행동하는 시나리오로 사회복지 지수가 가장 높게 학습되었다. 생산성은 자유시장보다 적지만 불평등 지수가 가장 낮게 학습되었으며, 스킬 레벨이 높은 Resident들은 생산성을 높이기 위해 열심히 집을 짓고, 자원 또한 열심히 확보하고 스킬 레벨이 낮아 거래 위주로 코인을 얻는 Resident들의 불평등 지수를 낮추기 위해 의도적으로 자원을 높은 가격에 매입 하는 것을 확인할 수 있었다.

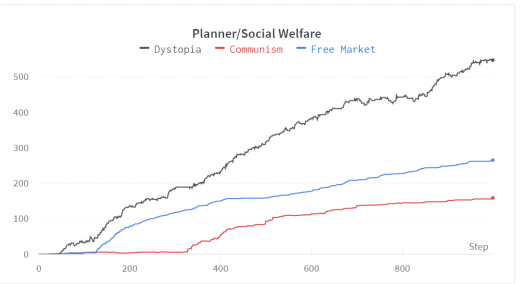


그림 5. 시나리오별 사회복지 지수 비교
Fig 5. Social Welfare Index Comparison by Scenario

4.2 2단계 학습

2단계 학습은 1단계 학습 이후에 진행되며, 정부의 조세 정책을 의미한다. Planner 에이전트는 각 Resident들에게 100 time-step마다 소득 구간별 세율을 통해 세금을 걷고, 이를 균등하게 다시 재분배 할 수 있다. 이때, 세율이 높아지면 평등해지지만 생산성이 감소하고, 반대로 세율이 낮아지면 생산성은 올라가지만 불평등 격차가 심해지는 문제가 발생하므로 Planner는 수식 3과 같이 보상으로 사회복지 함수를

활용하여 서로 트레이드오프 관계인 생산성과 불평등의 균형을 맞추는 것을 목표로 조세 정책을 학습한다. 본 논문에서는 AI Economist, Saez, US Federal, Free Market 총 4가지 조세 정책에 대해 2단계 학습을 진행하고 비교 분석하였다. 첫 번째는 Free Market으로 조세 정책이 없어 1단계 학습과 차이가 없다. 두 번째는 US Federal 정책으로 2020년 미국의 소득 구간별 세율을 그대로 적용하여 실험하였다. 세 번째는 Saez 조세 정책으로 경제 개념 중 Saez 공식을 통해 계산된 소득 구간별 세율을 적용하였다. 마지막 네 번째는 AI Economist로 세율 학습을 통해 소득 구간별 사회복지를 최대화하기 위해 학습된 모델로 실험을 진행하였다.

그림 6은 정책별 사회복지를 비교한 그래프로 60만 번 에피소드를 학습했을 때, AI Economist 정책의 사회복지 지수가 가장 높았고, Saez 정책이 두 번째, US Federal과 Free Market이 비슷하게 수렴하는 것을 알 수 있었다. 그림 7은 정책별 생산성과 평등 지수를 비교한 그래프로 Free Market이 세금이 없기 때문에 생산성이 가장 높고 평등 지수가 가장 낮으며, AI Economist 정책이 생산성, 평등 지수가 모두 적절히 높게 학습된 것을 확인하였다.

그림 8은 조세 정책별 소득 구간별 세율을 나타낸 그래프이다. Free Market은 세금이 없는 자유시장으로 제외하였으며, US Federal과 Saez는 각 정책에 대한 세율을 그대로 표현하였다. US Federal은 소득이 높을수록 세율이 올라가는 경향을 보이고, Saez는 반대로 소득이 높을수록 세율이 낮아지는 경향을 보이고 있다. 반면, 학습된 AI Economist는 소득 구간별 세율이 지그재그 형태로 세율이 높았다가 낮아지는 경향을 보였다. 특히, 가장 소득이 높은 구간에서 세율이 높고, 중간에 다시 세율이 낮도록 학습이 되었다.

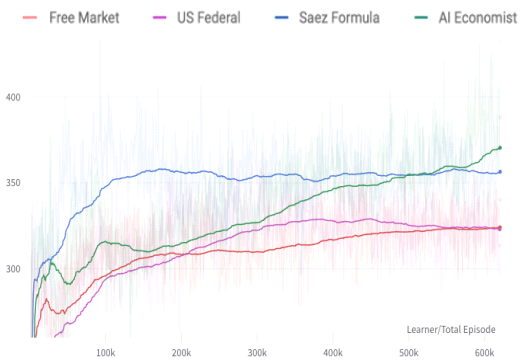


그림 6. 조세 정책별 사회복지 비교 그래프
Fig. 6. Social Welfare Comparison by Tax Policy

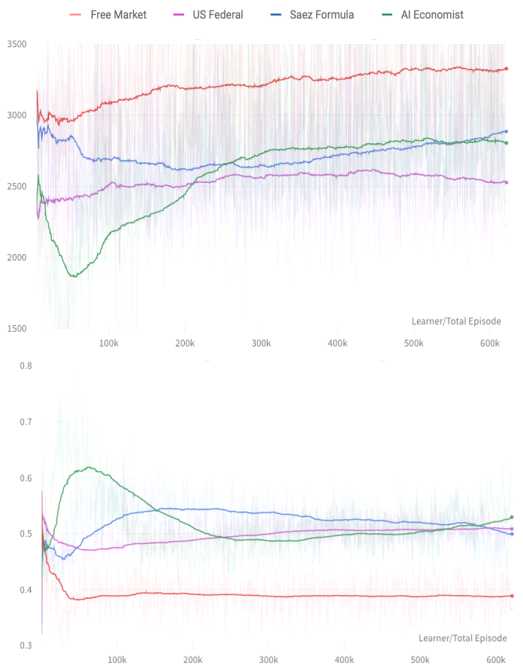


그림 7. 조세 정책별 생산성(위), 평등지수(아래) 비교
Fig. 7. Comparison of Productivity (above) and Equality Index (below) by Tax Policy

그림 8. 소득 구간 및 조세 정책별 세율
Fig. 8. Tax Rate by Tax Bracket and Tax Policy

그림 9는 학습한 AI Economist 정책에서 Resident의 소득과 납부한 세금을 비교한 그래프로, 파란색은 실제 소득과 실제 납부한 세금을 나타내고 검정색은 전체 소득의 평균 값과 해당 소득일 때 납부해야 할 세금을 나타낸다.

AI Economist 조세 정책은 그림 8과 같이 세율이 소득 구간에 따라 지그재그로 높았다 낮아지기를 반복하고 있는데 Resident는 총 1000번의 time-step에서 100 step 마다 세금을 걷는 것을 알고, 100 step마다 열심히 노동을 통해 소득을 늘려 세금을 많이 납부하고 다음 100 step 동안은 일을 거의 하지 않아 세금을 적게 납부하는 행동 양식을 보이는 것을 확인하였다. 이러한 행동은 Resident의 전체 수입을 기준으로 세금을 계산해보면 확연한 차이가 나는 것을 알 수 있는데 Resident가 세금을 최대한 적게 납부하기 위하여 AI Economist 정책에서의 세율이 낮은 소득 구간에 해당

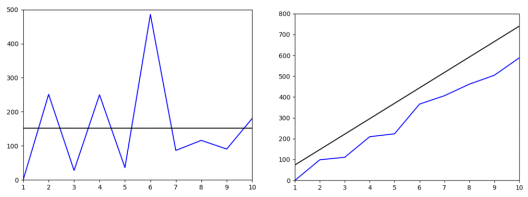


그림 9. AI Economist 정책에서 Resident의 소득(왼쪽)과 납부한 세금(오른쪽) 비교 그래프
Fig. 9. Comparison of Residents' income(left) and Taxes Paid(right) in the AI Economist Policy

하기 위해 또한, 본인의 유틸리티를 최대화하기 위하여 행동하는 것을 유추할 수 있다.

V. 결 론

본 논문에서는 전통적인 경제 개념과 공식을 적용하여 다양한 경제 주체들 간의 상호작용 속에서 조세 정책을 실험 및 평가하기 위해 AI Economist를 활용하여 심층 강화학습 기반 조세 정책 최적화 시뮬레이션 환경을 분석 및 실험하였다. 경제 활동 에이전트인 Resident와 조세 정책을 통해 사회복지를 실현하기 위한 Planner 에이전트를 2단계로 나누어 학습하였으며, 실험 결과 각 Resident의 스킬 레벨과 정부의 조세 정책에 따라 각 에이전트들이 서로 다른 행동 양식을 보이고, 실제 현실의 경제 환경을 일부 반영할 수 있다는 것을 확인하였다.

또한, 향후 소득 구간, 보상, 경제 활동 등 각종 변수들과 강화학습 알고리즘을 개선하여 다양한 경제 정책을 적용할 수 있도록 개선할 계획이다.

References

[1] N. G. Mankiw, M. Weinzierl, and D. Yagan, "Optimal taxation in theory and practice," *J. Econ. Perspectives*, vol. 23, no. 4, pp. 147-174, 2009.

[2] P. Diamond and E. Saez, "The case for a progressive tax: From basic research to policy recommendation," *J. Econ. Perspectives*, vol. 25, no. 4, pp. 165-190, 2011.

[3] AI Economist, *GitHub Repository*, https://github.com/salesforce/ai-economist.

[4] S. Zheng, et al., "The AI economist: Improving equality and productivity with ai-driven tax policies," *arXiv preprint arXiv:2004.13332*, 2020. (https://doi.org/10.48550/arXiv.2004.13332)

[5] S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher, "The AI economist: Optimal economic policy design via two-level deep reinforcement learning," *arXiv preprint arXiv: 2108.02755*, 2021. (https://doi.org/10.48550/arXiv.2108.02755)

[6] J. S. Heo, Y. H. Choi, Y. J. Seok, and Y. H. Han, "Experiment and analysis of policy optimization for ai-based economic agents," in *Proc. 2022 KICS Fall Conf.*, pp. 238-239, Gyeongju, Korea, Nov. 2022.

[7] J. S. Heo, Y. H. Choi, Y. J. Seok, J. S. Youn, and Y. H. Han, "Deep reinforcement learning-based tax policy optimization simulation environment analysis and experiment," in *Proc. 2023 KICS Winter Conf.*, Pyeongchang, Korea, Feb. 2023.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017. (https://doi.org/10.48550/arXiv.1707.06347)

허 주 성 (Joo-Seong Heo)

2016년 2월 : 한국기술교육대학교 학사
2019년 2월 : 한국기술교육대학교 석사
2019년 3월~현재 : 한국기술교육대학교 박사과정 (수료)
2016년 11월~현재 : 씽크온웹 대표
<관심분야> 딥러닝, 강화학습, 빅데이터
[ORCID:0000-0002-2486-9515]

최 요 한 (Yo-Han Choi)

2021년 2월 : 한국기술교육대학교 학사
2021년 3월~현재 : 한국기술교육대학교 석사과정
<관심분야> 딥러닝, 강화학습, 심층강화학습
[ORCID:0009-0003-6861-3030]

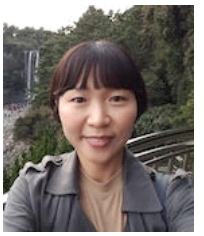
석 영 준 (Yeong-Jun Seok)

2022년 2월 : 한국기술교육대학교 학사
2022년 3월 : 한국기술교육대학교 석사과정
<관심분야> 심층강화학습, 조합최적화, 네트워크 슬라이싱, 양자컴퓨터
[ORCID:0009-0007-6942-3596]

유 태 완 (Taewan You)

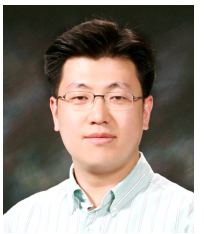
2001년 2월 : 전북대학교 컴퓨터과학과 졸업 (학사)
2004년 2월 : 서울대학교 전기컴퓨터공학부 졸업 (석사)
2009년 2월 : 서울대학교 전기컴퓨터공학부 수료 (박사)
2015년~현재 : 한국전자통신연구원 인공지능연구소
<관심분야> 인공지능, 행위자 기반 시뮬레이션, 계량경제모형, 등

이 연 희 (Yeonhee Lee)



2012년 2월 : 충남대학교 컴퓨터공학과 (석사)
2015년 2월 : 충남대학교 컴퓨터공학과 (박사)
2016년 : 아파치 오픈 소스 커미터
2014~현재 : 한국전자통신연구원 인공지능연구소
<관심분야> 경제 디지털트윈, 디지털트윈&AI 플랫폼 기술, 빅데이터, 클라우드, 컴퓨터 네트워크, 네트워크 트래픽 분석

한 연 희 (Youn-Hee Han)



1996년 2월 : 고려대학교 수학과 (학사)
1998년 2월 : 고려대학교 컴퓨터학과 (석사)
2002년 2월 : 고려대학교 컴퓨터학과 (박사)
2002년 3월~2006년 2월 : 삼성종합기술원 전문연구원
2013년 9월~2014년 8월 : SUNY at Albany, Department of Computer Science 방문교수
2006년~현재 : 한국기술교육대학교 컴퓨터공학부 교수
<관심분야> 사물인터넷, 5G/6G, 딥러닝, 강화학습, 조합최적화
[ORCID:0000-0002-5835-7972]