

AI Agent for Scheduling CSE Classes

노승현(2013136033) and 임찬규(2014136106) and 이슬빈(2016136098)

한국기술교육대학교
changu13@naver.com

1 Introduction

1.1 개요

본 과목의 팀 프로젝트 목표는 한국기술교육대학교 컴퓨터공학부의 2019년도 1학기 수업 시간표를 만들 수 있는 인공지능 에이전트를 구현하고 그 결과를 분석하는 것이다. 제공된 컴퓨터공학부의 모든 강의와 강의실에 대한 데이터를 이용하여 인공지능 에이전트가 제약 조건을 만족하는 최적의 수업 시간을 편성할 수 있도록 수행한다.

1.2 시스템 구성

시스템 구성은 다음과 같다.

① 검색 창

인공지능 3팀

교수자	정구철				
검색...	검색	과목 담기			
	과목코드	과목명	분반	교수자	학점
1	CPA131	C프로그래밍 I	01	정구철	3
2	CPA110	컴퓨터공학입문	01	정구철	3
3	CPA110	컴퓨터공학입문	02	정구철	3

Fig. 1. 검색 창

시스템 왼쪽 상단에 검색 창이 배치되어있다. 첫 번째 상자로 전체, 과목명, 교수자로 분류할 수 있으며, 이어서 두 번째 상자가 활성화되어 선택할 수 있다. 검색 창에 관련 키워드를 검색하여 원하는 과목을 찾을 수도 있으며, 과목에 커서를 두고 과목 담기 버튼을 누르면 해당 과목이 시간표에 담기게 된다.

② 담긴 과목 목록 창

과목 빼기					
	과목코드	과목명	분반	교수자	학점
1	CPA110	컴퓨터공학입문	02	정구철	3

Fig. 2. 담긴 과목 목록 창

시스템 왼쪽 하단에 담긴 과목 목록 창이 있다. 과목 담기를 통해 담은 과목의 정보를 표시해준다. 원하는 과목에 커서를 두고 과목 빼기 버튼을 통해 시간표에서 과목을 제거할 수 있다.

③ 데이터 제어 및 에이전트 기능 버튼

데이터 가져오기	에이전트 시작	초기화	데이터 내보내기
----------	---------	-----	----------

Fig. 3. 데이터 제어 및 에이전트 기능 버튼

담긴 과목 목록 창 하단에 시스템의 중요 기능을 담당하는 데이터 제어 및 에이전트 기능 버튼들이 있다. 시스템을 처음 시작하면 데이터 가져오기를 제외한 모든 버튼들이 비활성화 되어 있다. 강의 데이터를 가져오게 되면 인공지능 에이전트를 시작할 수 있게 되고 데이터 내보내기를 통해 수업 시간 및 강의실 배치가 완료된 데이터를 저장하고 확인할 수 있다.

④ 시간표 창

	월	화	수	목	금
01A					
01B			컴퓨터공학입문, 01,정구철,공학2 관/121호		
02A			컴퓨터공학입문, 01,정구철,공학2 관/121호		
02B			컴퓨터공학입문, 01,정구철,공학2 관/121호		
03A					
03B	컴퓨터공학입문, 01,정구철,공학2 관/121a호				
04A	컴퓨터공학입문, 01,정구철,공학2 관/121a호				
04B	컴퓨터공학입문, 01,정구철,공학2 관/121a호				
05A					

Fig. 4. 시간표 창

시스템의 오른쪽 화면에는 시간표 창이 배치되어 있다. 담은 과목 목록에 있는 과목들이 시간표 창에 표시된다. 과목명, 분반, 교수자, 강의실 정보를 확인할 수 있다.

1.3 개발환경

CPU : AMD Ryzen 7 1700(16, CPUs), ~3.0GHz
 RAM : 16GB
 Language : Python 3.6.8 :: Anaconda
 Libraries : OpenPyXL, PyQt
 Tools : Visual Studio Code, Pycharm

1.4 팀 활동 내용

프로젝트 제출까지 총 5번의 모임이 있었고, 활동 내용은 다음과 같다.

05.08	05.23	06.06	06.09	06.11
주제 확인 및 계획 편성	알고리즘 전략 공유	그룹 코딩		검토 및 자료 작성

Table 1. 팀 활동 내용

2 Point

2.1 설계 내용

① 객체 설계

시스템에 구현된 객체는 크게 수업 클래스와 강의실 클래스로 나뉜다.

• 수업 클래스

- 과목코드
- 과목명
- 강의실
- 학점
- 이론시간
- 실습시간
- 수업속성
- 교수
- 트랙
- 분반
- 수강정원
- 수업 관계 목록
- 시간표

• 강의실 클래스

- 건물
- 강의실 번호
- 강의실 인원
- 강의실 명
- 강의실 속성
- 실습실 여부
- 시간표

※ Code

```
class Lecture:
    def __init__(self):
        self.classCode = '' # 과목코드
        self.className = '' # 과목명
        self.classRoom = [] # 강의실
        self.score = 0 # 학점
        self.classTime = 0 # 이론시간
        self.researchTime = 0 # 실습시간
        self.classProperty = [] # 수업속성(HW, SW, 이론)
        self.professor = '' # 교수
        self.track = '' # 트랙
        self.classNum = 0 # 분반
        self.classLimit = 0 # 수강정원
        self.relation = [] # 연관된 수업 목록
        self.timeTable = [[['Vacant'] for col in range(18)] for row in range(5)] # 시간표 (시간 0~17, 강

    def setClassRelation(self, relatedClass):
        self.relation.append(relatedClass) # 연관된 수업 목록

class ClassRoom:
    def __init__(self):
        self.Building = '' # 건물
        self.roomNum = '' # 강의실 번호
        self.accommodableLimit = 0 # 강의실 인원
        self.isResearch = False # 실습실 여부
        self.name = '' # 강의실명
        self.property = '' # 강의실 속성(HW, SW, 이론, 담현)
        self.timeTable = [[None for col in range(18)] for row in range(5)] # 시간표 (시간 0~17, 강
```

Fig. 5. 객체 설계 코드

② 제약 조건에 따른 수업 관계 목록 생성

제약 조건은 다음과 같이 두 가지로 선정하였다.

- 교수는 동일한 시간에 하나의 수업만을 진행해야 한다.
- 트랙별 전공선택 수업은 가능한 겹치지 않게 할당한다.

해당 수업의 교수자와 수업 속성을 모든 수업과 비교한다. 해당 수업의 교수자와 각 수업의 교수자가 같거나 해당 수업의 속성과 각 수업의 속성이 같을 경우, 두 수업의 관계를 연결시킨다. 수업은 클래스로 구현되어 하나의 객체이고, 고유의 수업 관계 목록(relation[])을 가지고 있다. 그러므로 수업 시간 배치를 할 때 수업 관계 목록에 있는 수업들만 확인하여 해당 수업 배치 유무를 결정할 수 있다.

③ 수업 시간 배치 전략

- 공학설계(I, II), 졸업설계(I, II) 수업은 수요일 오후 2시로 고정
- 월요일 오전 9시부터 순차적으로 탐색하며 수업을 배치함
- 3시간 강의는 1시간 강의보다 2시간 강의를 먼저 배치됨
- 4시간 강의는 앞 강의가 이론, 뒷 강의가 실습 강의임
- 교수는 하루에 수업을 4시간 이상 할 수 없음
- 수요일, 금요일 오후 2시 이후에는 수업이 존재하면 안됨
- 두 번째 배치하는 시간은 첫 번째 배치된 시간 다음 날부터 배치 가능
- 점심 시간(05A~05B)은 수업을 배치하지 않음

수업은 속성에 따라 수업시간이 다르다. 몇몇 특수한 수업을 제외하고, 이론 수업은 총 3시간으로 일주일에 2시간과 1시간으로 두 번 나누어 진행한다. 실습 수업은 총 4시간으로 일주일에 각각 2시간으로 두 번 나누어 진행한다. 이 과정에서 강의실 배치를 명확히 하기 위해 실습 수업을 크게 두 분류(SW, HW)로 분류하였다. 시간 배치 에이전트를 실행함과 동시에 앞에서 언급한 제약 조건에 따른 수업 관계 목록을 생성하기 위한 작업을 수행한다. 그 후 수업 시간을 배치하는데, 첫 번째 2시간 진행하는 수업을 먼저 배치하고 두 번째는 이론 수업일 경우 1시간, 실습 수업일 경우 2시간을 배치한다.

※ Code

```
# 시간 배치 에이전트
def TimeAgent(self):
    self.relateLectures()
    isDone = False # 수업 배치 Flag
    # 첫번째 수업 시간 배치
    for lecture in self.LectureList:
        if '캡스톤디자인' in lecture.className: # 공학설계, 졸업설계는 제외
            continue
        for day in range(self.checkLectureTimeOfDay(lecture), 5): # 요일마다
            if isDone is True:
                isDone = False
                break
            for time in range(0, 18, 2): # 1시간 간격마다
                isAvailable = self.checkRelatedLecture_First(lecture, day, time)
                if isAvailable is True: # 이론 수업 배치가 가능하면 시간 할당
                    self.giveLectureFirstTime(lecture, day, time)
                    isDone = True
                    break
    isDone = False
    # 두번째 수업 시간 배치
```

Fig. 6. 수업 시간 배치 전략 코드

④ 강의실 배치 전략

- 강의실 / 실습실의 정원은 과목의 수강정원보다 같거나 커야함

먼저 수업 속성별 목록을 생성한다. 수업 속성은 이론, 실습(SW), 실습(HW) 3가지로 분류된다. 수업 속성별로 리스트가 구성이 되었다면, 시간 배치 전략 경우처럼 첫 번째 시간에 대해 강의실을 먼저 배치한다. 시간표에는 각 시간마다 비어있음(Vacant), 비어있지 않음(Occupied), 해당 수업 시간(Mine)으로 분류되어 있고, Mine이 부여된 시간을 탐색하게 되면 그 시간에 해당하는 수업에 비어있는 강의실을 부여한다. 두 번째 시간에 대한 강의실 배치도 동일하게 수행한다.

※ Code

```

# 강의실 배치 에이전트
def ClassroomAgent(self):
    # 수업 속성별 리스트 생성
    theoryList = []
    swList = []
    hwList = []
    for classroom in self.classRoomList:
        if '이론' in classroom.property:
            theoryList.append(classroom)
        if 'SW' in classroom.property:
            swList.append(classroom)
        if 'HW' in classroom.property:
            hwList.append(classroom)

    theoryList.sort(key=lambda classroom: classroom.accommodableLimit)
    swList.sort(key=lambda classroom: classroom.accommodableLimit)
    hwList.sort(key=lambda classroom: classroom.accommodableLimit)

    # 첫번째 수업을 배치
    isDone = False # 강의실 배치 완료 여부
    for lecture in self.LectureList:
        if '이론' in lecture.classProperty: # 수업 속성에 이론이 있는 경우
            for day in range(5):
                if isDone is True:
                    isDone = False
                    break
                for time in range(0, 18, 2):
                    if isDone is True:
                        break
                    for classroom in theoryList:
                        if '이론' in classroom.property:
                            if lecture.timeTable[day][time][0] is 'Mine':
                                if classroom.timeTable[day][time] is None and int(lecture.classLimit) <= classroom.accommodableLimit:
                                    lecture.classRoom.append(classroom)
                                    classroom.timeTable[day][time] = 'Used'
                                    classroom.timeTable[day][time + 1] = 'Used'

```

Fig. 7. 강의실 배치 전략 코드

2.2 수행 과정

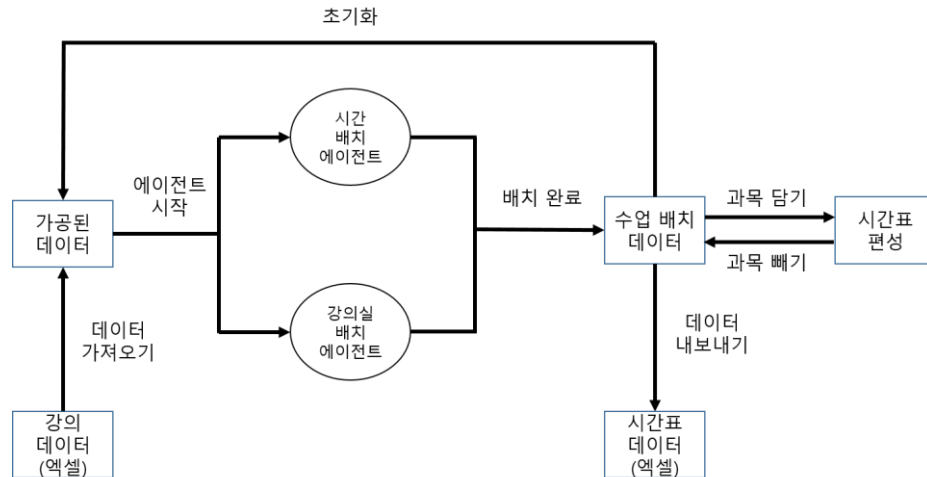


Fig. 8. 수행 과정

3 Conclusion

3.1 결과

인공지능 3팀

교수자: 오홍선

검색... 검색 과목 담기

과목코드	과목명	분반	교수자	학점
1 CP5460	인공지능	01	오홍선	3
2 CPA320	컴퓨터구조	04	오홍선	3
3 CPA320	컴퓨터구조	03	오홍선	3

과목 빼기

과목코드	과목명	분반	교수자	학점
1 CP5460	인공지능	01	오홍선	3
2 CPA320	컴퓨터구조	04	오홍선	3
3 CPA320	컴퓨터구조	03	오홍선	3

데이터 가져오기 에이전트 시작 초기화 데이터 내보내기

	월	화	수	목	금
01A			컴퓨터구조 04, 오홍선, 공학2관/1218호		
01B	컴퓨터구조 04, 오홍선, 공학2관/314호	컴퓨터구조 03, 오홍선, 공학2관/314호	컴퓨터구조 03, 오홍선, 공학2관/1218호		
02A					
02B					
03A					
03B					
04A					
04B					
05A					
05B					
06A					
06B	인공지능 01, 오홍선, 공학2관/1218호				
07A					
07B					
08A					
08B					
09A			인공지능 01, 오홍선, 공학2관/1218호		
09B					

Fig. 9. 결과

3.2 결과 분석

① 시간 복잡도

- 시간 배치 에이전트 → $O(n)$
- 강의실 배치 에이전트 → $O(n)$

② 공간 복잡도

- 시간 배치 에이전트 → $O(n)$
- 강의실 배치 에이전트 → $O(n)$

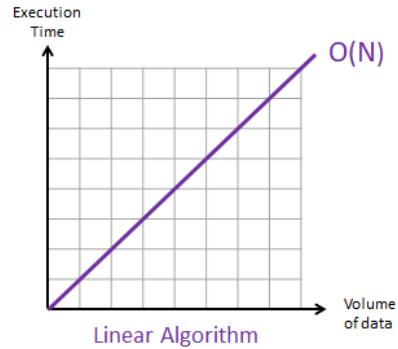


Fig. 10. $O(n)$

③ 제약을 변경했을 경우 결과

- 수업의 트랙을 전부 제거 후 수행 → 오류 없음

	A	B	C	D	E	F	G	H	I	J	K
1	과목	분반	교수	강의실	시간						
2	물리설계101	박승철	수/06A, 수/06B,								
3	물리설계101	박승철	수/06A, 수/06B,								
4	신입재학101	박승철	담현실학관, 월/01A, 월/01B,								
5	공학설계101	박승철	수/05A, 수/05B,								
6	컴퓨터네트워크2	박승철	공학2관/12, 화/02A, 화/02B, 화/03A, 화/03B, 화/03A, 화/03B,								
7	컴퓨터네트워크1	박승철	공학2관/12, 화/01A, 화/01B, 화/02A, 화/02B, 목/01A, 목/01B,								
8	공학설계101	박승철	수/06A, 수/06B,								
9	창의력공학101	윤환경	공학2관/4C, 월/01A, 월/01B, 월/02A, 월/02B, 수/01A, 수/01B, 수/02A, 수/02B,								
10	기초전기공학101	윤환경	공학2관/11, 화/03A, 화/03B, 화/04A, 화/04B, 수/03A, 수/03B, 수/04A, 수/04B,								
11	컴퓨터네트워크4	주영복	공학2관/4C, 월/01A, 월/01B, 월/02A, 월/02B, 수/01A, 수/01B, 수/02A, 수/02B,								
12	컴퓨터네트워크4	무하마드	공학2관/12, 화/01A, 화/01B, 화/02A, 화/02B, 수/01A, 수/01B,								
13	컴퓨터네트워크3	무하마드	공학2관/12, 화/03A, 화/03B, 화/04A, 화/04B, 수/02A, 수/02B,								
14	알고리즘102	무하마드	공학2관/4C, 화/01A, 화/01B, 화/02A, 화/02B, 수/03A, 수/03B, 수/04A, 수/04B,								
15	컴퓨터시스템102	주영복	공학2관/4C, 화/03A, 화/03B, 화/04A, 화/04B, 수/03A, 수/03B, 수/04A, 수/04B,								
16	자료구조101	무하마드	공학2관/4C, 화/03A, 화/03B, 화/04A, 화/04B, 목/01A, 목/01B, 목/02A, 목/02B,								
17	디지털공학104	주영복	공학2관/11, 화/01A, 화/01B, 화/02A, 화/02B, 목/01A, 목/01B, 목/02A, 목/02B,								
18	디지털공학103	주영복	공학2관/11, 화/03A, 화/03B, 화/04A, 화/04B, 목/03A, 목/03B, 목/04A, 목/04B,								
19	컴퓨터공학104	윤환경	공학2관/12, 화/01A, 화/01B, 화/02A, 화/02B, 목/01A, 목/01B, 목/02A, 목/02B,								
20	컴퓨터공학103	윤환경	공학2관/12, 화/03A, 화/03B, 화/04A, 화/04B, 목/03A, 목/03B, 목/04A, 목/04B,								
21	자바프로그래밍101	김상진	공학2관/4C, 화/01A, 화/01B, 화/02A, 화/02B, 화/06A, 화/06B, 화/07A, 화/07B,								
22	C프로그래밍101	조재수	공학2관/12, 화/01A, 화/01B, 화/02A, 화/02B, 화/06A, 화/06B, 화/07A, 화/07B,								
23	컴퓨터비전101	조재훈	공학2관/4C, 화/01A, 화/01B, 화/02A, 화/02B, 화/01A, 화/01B, 화/02A, 화/02B,								
24	정보보호기초101	김상진	담현실학관, 목/03A, 목/03B, 목/04A, 목/04B, 수/01A, 수/01B,								
25	인공지능101	오홍선	공학2관/31, 월/01A, 월/01B, 화/02A, 화/02B, 화/04A, 화/04B,								
26	스마트IoT101	김원태	공학2관/24, 월/01A, 월/01B, 화/02A, 화/02B, 화/06A, 화/06B, 화/07A, 화/07B,								
27	멀티코어프로그래밍101	김원태	공학2관/24, 월/01A, 월/01B, 화/02A, 화/02B, 화/03A, 화/03B, 화/04A, 화/04B,								
28	네트워크프로그래밍101	서희석	공학2관/12, 화/01A, 화/01B, 화/02A, 화/02B, 화/03A, 화/03B, 화/04A, 화/04B,								
29	컴퓨터구조104	오홍선	공학2관/31, 월/03A, 화/03B, 화/04A, 화/04B, 수/01A, 수/01B,								
30	컴퓨터구조103	오홍선	공학2관/31, 월/03A, 화/03B, 화/04A, 화/04B, 수/01A, 수/01B,								

Fig. 11. 트랙 제거 후 수행

- 임의의 강의실 제거 후 수행 → 오류 없음

	A	B	C	D	E	F
1	건물	강의실번호	인원	실습실	실습실명	수업속성
2	공학2관	117	48 o		기초전기전자 실습실	HW
3	공학2관	119	48 o		IoT실습실	HW
4	공학2관	124	48 o		임베디드시스템 실습실	SW
5	공학2관	125	48 o		시스템 S/W 실습실	SW
6	공학2관	249	45 o		공학설계 실습실	SW
7	공학2관	401	40 o		프로그래밍 실습실	SW
8	공학2관	402	25 o		지능형 영상처리 실습실	SW
9	공학2관	403	48 o		데이터베이스 실습실	SW
10	공학2관	408	40 o		지능형 S/W 실습실	SW
11	공학2관	409	40 o		가상현실 실습실	SW
12	공학2관	410	42 o		네트워크 시스템 실습실	SW
13	공학2관	121	42 x		이론 강의실	이론
14	공학2관	314	45 x		이론 강의실	이론
15	공학2관	316	45 x		이론 강의실	이론
16	담헌 실학관	102	80 x		이론 강의실	담헌
17						
18						
19						

Fig. 11. 강의실 제거 후 수행

④ 인공지능 에이전트의 장점

- 하루에 수업을 4시간 이상 할 수 없도록 하여 스케줄 관리 가능
- 수요일, 금요일 오후 2시 이후 수업을 배치하지 않아 여유 시간 확보
- 두 번째 수업 시간을 첫 번째 수업 시간과 같은 날에 배치하지 않음
- 점심 시간을 고려하여 식사 시간 확보
- 과목의 수강정원보다 큰 강의실로 배치

⑤ 인공지능 에이전트의 단점

- 금요일에 수업이 거의 배치되지 않아 비어있는 시간 활용 미흡
- 9시부터 순차적으로 탐색하며 배치하기 때문에 첫 수업이 9시인 경우가 대부분임

⑥ 인공지능 에이전트의 개선점

- 시간을 효율적으로 사용하기 위해 금요일에 수업 배치 하는 방안을 모색해야 함
- 순차적인 탐색 방법이 아닌 다른 효율적인 방법을 모색해야 함