



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0071683

(43) 공개일자 2025년05월22일

(51) 국제특허분류(Int. Cl.)

G06Q 50/20 (2012.01) G06Q 10/06 (2012.01)

(52) CPC특허분류

G06Q 50/205 (2013.01)

G06Q 10/06311 (2013.01)

(21) 출원번호 10-2023-0158493

(22) 출원일자 2023년11월15일

심사청구일자 2023년11월15일

(71) 출원인

국립부경대학교 산학협력단

부산광역시 남구 용소로 45, 부경대학교내 (대연동)

(72) 발명자

김창수

부산광역시 남구 분포로 113, 214동 603호 (용호동, 엘지메트로시티)

(74) 대리인

김정수

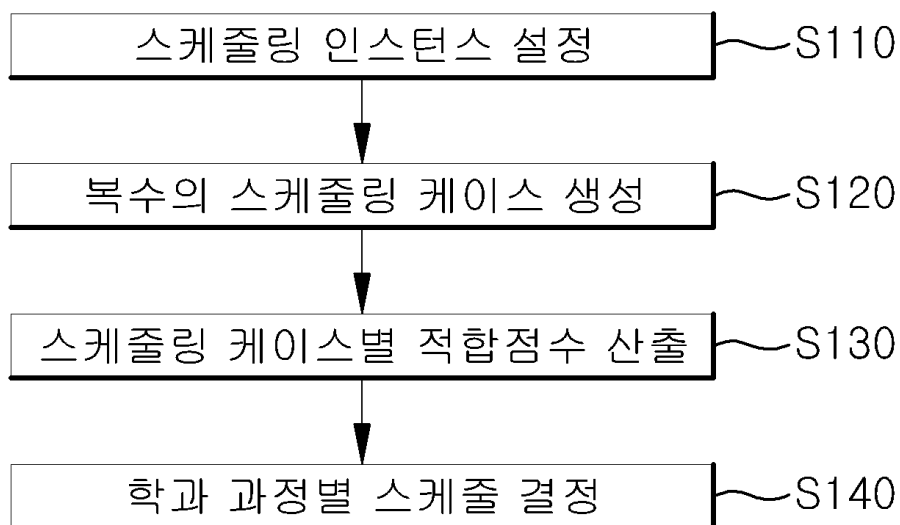
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 학과 과정 스케줄링 장치 및 방법

(57) 요약

학과 과정 스케줄링 장치 및 방법이 개시된다. 학과 과정 스케줄링 방법은, 학과 과정 스케줄링을 위한 스케줄링 인스턴스를 설정하는 단계, 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계, 생성된 복수의 스케줄링 케이스 별로 적합점수를 산출하는 단계 및 학과 과정별로 적합점수가 가장 큰 스케줄링 케이스를 선택하여 학과 과정별 스케줄을 결정하는 단계를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1345365715
과제번호	2021R1I1A3052605
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학학술연구기반구축
연구과제명	제조업 생산성 향상을 위한 AI기반 실시간 생산 스케줄링 동적 정보시스템 개발
기 여 율	1/1
과제수행기관명	부경대학교
연구기간	2023.03.01 ~ 2024.02.29

명세서

청구범위

청구항 1

학과 과정 스케줄링 장치가 수행하는 학과 과정 스케줄링 방법에 있어서,
 학과 과정 스케줄링을 위한 스케줄링 인스턴스를 설정하는 단계;
 상기 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계;
 상기 생성된 복수의 스케줄링 케이스 별로 적합점수를 산출하는 단계; 및
 상기 학과 과정별로 적합점수가 가장 큰 스케줄링 케이스를 선택하여 학과 과정별 스케줄을 결정하는 단계를 포함하는 학과 과정 스케줄링 방법.

청구항 2

제1항에 있어서,
 상기 스케줄링 인스턴스는 강의실(R), 타임슬롯(TS) 및 학과과정(C)을 포함하고,
 상기 강의실(R)은 식별자, 과정명, 기능 유형, 수용가능인원, 소유자의 변수를 포함하고,
 상기 타임슬롯(TS)은 요일, 강의 시작 및 종료 시각의 변수를 포함하고,
 상기 학과과정(C)은 과목명, 교수진, 학과, 학생 그룹, 지원 번호, 과정 유형, 교수 선호 시간, 교수 선호 시간에 상응하는 타임슬롯, 강의실의 변수를 포함하는 것을 특징으로 하는 학과 과정 스케줄링 방법.

청구항 3

제1항에 있어서,
 상기 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계는,
 미리 설정된 휴리스틱 알고리즘을 이용하여, 상기 학과 과정 별로 타임슬롯 및 강의실을 할당하여, 학과 과정 대 타임슬롯 및 학과 과정 대 강의실의 조합을 생성함으로써, 상기 학과 과정별로 복수의 스케줄링 케이스를 생성하는 것을 특징으로 하는 학과 과정 스케줄링 방법.

청구항 4

제1항에 있어서,
 상기 적합점수는 하기 수학식으로 이용하여 산출되는 것을 특징으로 하는 학과 과정 스케줄링 방법.

$$\begin{aligned} Score &= S_{hard} + S_{soft} \\ &= 0 + S_{soft} \\ &= a_1 S_{S1} + a_2 S_{S2} + a_3 S_{S3} + a_4 S_{S4} + a_5 S_{S5} \end{aligned}$$

여기서, S_{hard} 는 기 설정된 하드 제약조건의 만족여부에 따라 산출되는 점수로, 기본적으로 0이고, S_{soft} 는 기 설정된 소프트 제약조건에 만족여부에 따라 산출되는 점수로, $S1 \sim S5$ 의 5개 소프트 제약조건의 만족여부에 따라 설정된 점수의 합이고, 계수 $a1 \sim a5$ 는 각 소프트 제약조건의 보상 및 패널티 레벨을 나타냄

청구항 5

학과 과정 스케줄링 장치에 있어서,

명령어를 저장하는 메모리; 및

상기 명령어를 실행하는 프로세서를 포함하되,

상기 명령어는,

학과 과정 스케줄링을 위한 스케줄링 인스턴스를 설정하는 단계;

상기 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계;

상기 생성된 복수의 스케줄링 케이스 별로 적합점수를 산출하는 단계; 및

상기 학과 과정별로 적합점수가 가장 큰 스케줄링 케이스를 선택하여 학과 과정별 스케줄을 결정하는 단계를 포함하는 학과 과정 스케줄링 방법을 수행하는 것을 특징으로 하는 학과 과정 스케줄링 장치.

발명의 설명

기술 분야

[0001] 본 발명은 학과 과정 스케줄링 장치 및 방법에 관한 것이다.

배경 기술

[0002] 시간표 작성이라고 하는 대학 과정 스케줄링 문제(UCSP: University course scheduling problem)는 효율성을 향상시키고 인건비를 줄이기 위하여 지능형 교육 시스템을 구현하는데 필수적이다. 그 목적은 교수, 학생, 강의실 등의 교수 자원을 고정된 시간 내에 배치하여 수 개의 제약 조건을 만족시키는 것이다. 이전에는 대학 직원이 이 문제를 직접 해결하였다. 우선, 간단한 대학 과정을 정리한 다음 완성된 대학 과정을 마련한다. 이는 대학 과정 스케줄링 문제의 지속 이후로 쉽게 간히고 빠져나오기 어려운 과정으로 많은 시간이 소요된다. 즉, 하나가 잘못되면 직원이 다른 모든 것을 조정해야 한다. 또한, 교수진, 학생, 강좌 수가 증가함에 따라 실제 대학 과정 스케줄링 문제는 점점 더 복잡해진다. 대학 과정 스케줄링 문제의 복잡한 인스턴스를 직접 해결하는 것은 어렵고 심지어 불가능하다. 그러므로, 자동적이고 정확한 대학 과정 스케줄링 방법이 필요하다.

선행기술문헌

특허문헌

[0003] (특허문헌 0001) 대한민국등록특허공보 제10-1958918호(2019.03.11)

발명의 내용

해결하려는 과제

[0004] 본 발명은 학과 과정 스케줄링을 위하여 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하고, 생성된 복수의 스케줄링 케이스에 대하여 적합점수를 산출하고, 산출된 적합점수에 따라 학과 과정별로 스케줄을 결정하는 학과 과정 스케줄링 장치 및 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0005] 본 발명의 일 측면에 따르면, 학과 과정 스케줄링 장치가 수행하는 학과 과정 스케줄링 방법이 개시된다.

[0006] 본 발명의 실시예에 따른 학과 과정 스케줄링 방법은, 학과 과정 스케줄링을 위한 스케줄링 인스턴스를 설정하는 단계, 상기 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계, 상기 생성된 복수의 스케줄링 케이스 별로 적합점수를 산출하는 단계 및 상기 학과 과정별로 적합점수가 가장 큰 스케줄

링 케이스를 선택하여 학과 과정별 스케줄을 결정하는 단계를 포함한다.

[0007] 상기 스케줄링 인스턴스는 강의실(R), 타임슬롯(TS) 및 학과과정(C)을 포함하고, 상기 강의실(R)은 식별자, 과정명, 기능 유형, 수용가능인원, 소유자의 변수를 포함하고, 상기 타임슬롯(TS)은 요일, 강의 시작 및 종료 시각의 변수를 포함하고, 상기 학과과정(C)은 과목명, 교수진, 학과, 학생 그룹, 지원 번호, 과정 유형, 교수 선호 시간, 교수 선호 시간에 상응하는 타임슬롯, 강의실의 변수를 포함한다.

[0008] 상기 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계는, 미리 설정된 휴리스틱 알고리즘을 이용하여, 상기 학과 과정 별로 타임슬롯 및 강의실을 할당하여, 학과 과정 대 타임슬롯 및 학과 과정 대 강의실의 조합을 생성함으로써, 상기 학과 과정별로 복수의 스케줄링 케이스를 생성한다.

[0009] 상기 적합점수는 하기 수학식으로 이용하여 산출된다.

$$\begin{aligned} \text{Score} &= S_{hard} + S_{soft} \\ &= 0 + S_{soft} \\ &= a_1 S_{S1} + a_2 S_{S2} + a_3 S_{S3} + a_4 S_{S4} + a_5 S_{S5} \end{aligned}$$

[0010]

[0011] 여기서, S_{hard} 는 기 설정된 하드 제약조건의 만족여부에 따라 산출되는 점수로, 기본적으로 0이고, S_{soft} 는 기 설정된 소프트 제약조건에 만족여부에 따라 산출되는 점수로, S1 ~ S5의 5개 소프트 제약조건의 만족여부에 따라 설정된 점수의 합이고, 계수 $a_1 \sim a_5$ 는 각 소프트 제약조건의 보상 및 패널티 레벨을 나타낸다.

[0012] 본 발명의 다른 측면에 따르면, 학과 과정 스케줄링 장치가 개시된다.

[0013] 본 발명의 실시예에 따른 학과 과정 스케줄링 장치는, 명령어를 저장하는 메모리 및 상기 명령어를 실행하는 프로세서를 포함하되, 상기 명령어는, 학과 과정 스케줄링을 위한 스케줄링 인스턴스를 설정하는 단계, 상기 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하는 단계, 상기 생성된 복수의 스케줄링 케이스 별로 적합점수를 산출하는 단계 및 상기 학과 과정별로 적합점수가 가장 큰 스케줄링 케이스를 선택하여 학과 과정별 스케줄을 결정하는 단계를 포함하는 학과 과정 스케줄링 방법을 수행한다.

발명의 효과

[0014] 본 발명의 실시예에 따른 학과 과정 스케줄링 장치 및 방법은, 학과 과정 스케줄링을 위하여 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성하고, 생성된 복수의 스케줄링 케이스에 대하여 적합점수를 산출하고, 산출된 적합점수에 따라 학과 과정별로 스케줄을 결정함으로써, 학과 과정을 자동적이고 정확하게 스케줄링 할 수 있다.

도면의 간단한 설명

[0015] 도 1은 본 발명의 실시예에 따른 학과 과정 스케줄링 장치가 수행하는 학과 과정 스케줄링 방법을 개략적으로 예시하여 나타낸 흐름도.

도 2 내지 도 6은 도 1의 본 발명의 실시예에 따른 학과 과정 스케줄링 방법을 설명하기 위한 도면.

도 7은 본 발명의 실시예에 따른 학과 과정 스케줄링 장치의 구성을 개략적으로 예시하여 나타낸 도면.

발명을 실시하기 위한 구체적인 내용

[0016] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다. 또한, 명세서에 기재된 "...부", "모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다.

[0017] 이하, 본 발명의 다양한 실시예들을 첨부된 도면을 참조하여 상술하겠다.

[0018] 도 1은 본 발명의 실시예에 따른 학과 과정 스케줄링 장치가 수행하는 학과 과정 스케줄링 방법을 개략적으로

예시하여 나타낸 흐름도이고, 도 2 내지 도 6은 도 1의 본 발명의 실시예에 따른 학과 과정 스케줄링 방법을 설명하기 위한 도면이다. 이하, 도 1을 중심으로, 본 발명의 실시예에 따른 학과 과정 스케줄링 방법에 대하여 설명하되, 도 2 내지 도 6을 참조하기로 한다.

[0019] S110 단계에서, 학과 과정 스케줄링 장치는, 학과 과정 스케줄링을 위한 스케줄링 인스턴스를 설정한다. 즉, 스케줄링 인스턴스는 사용자로부터 입력받아 설정될 수 있다.

[0020] 여기서, 스케줄링 인스턴스는 도 2에 도시된 바와 같이, 강의실(R), 타임슬롯(TS) 및 학과과정(C)의 리스트를 포함한다. 이러한 스케줄링 인스턴스는 하기 표와 같이 정의될 수 있다.

표 1

Object	Type	Variables
Room <i>R</i>	Fact	{id: int, name: str, room_type: str, size: str, department: str}
Timeslot <i>TS</i>	Fact	{id: int, dayofweek: str, start_time: datetime, end_time: datetime}
Course <i>C</i>	Entity	{id: int, subject: str, faculty: str, department: str, student_group: str, s_course_type: str, prefer_time: Timesolt, timesolt: Timesolt, room: R
Timetable <i>Ttable</i>	Decision	{room_list, course_list, score}

[0021]

[0022] 즉, 강의실(R)은 식별자, 과정명, 기능 유형, 수용가능인원, 소유자 등의 변수를 포함할 수 있다. 그리고, 타임슬롯(TS)은 요일, 강의 시작 및 종료 시각 등의 변수를 포함할 수 있다. 그리고, 학과과정(C)은 과목명, 교수진, 학과, 학생 그룹, 지원 번호, 과정 유형, 교수 선호 시간, 교수 선호 시간에 상응하는 타임슬롯, 강의실 등의 변수를 포함할 수 있다. 이로부터 산출되는 타임테이블(T)은 스케줄 결정을 위한 인스턴스로서, 강의실(R) 리스트, 학과과정(C) 리스트, 적합점수 등의 변수를 포함할 수 있다.

[0023] 예를 들어, 스케줄링 인스턴스는 하기 수학적식과 같이 설정될 수 있다.

수학적식 1

$$UCSP_{instance} = \begin{cases} TS = \{TS_1, TS_2, \dots, TS_m, \dots, TS_M\} \\ R = \{R_1, R_2, \dots, R_O, \dots, R_O\} \\ C = \{C_1, C_2, \dots, C_n, \dots, C_N\} \end{cases}$$

[0024]

[0025] 즉, 스케줄링 인스턴스는 M개의 타임슬롯(TS), O개의 강의실(R) 및 N개의 학과과정(C)으로 정의될 수 있다.

[0026] S120 단계에서, 학과 과정 스케줄링 장치는, 설정된 인스턴스를 조합하여 학과 과정별로 복수의 스케줄링 케이스를 생성한다.

[0027] 즉, 학과 과정 스케줄링 장치는 도 2에 도시된 바와 같이, 미리 설정된 휴리스틱 알고리즘을 이용하여 학과 과정 별로 타임슬롯 및 강의실을 할당하여 학과 과정 대 타임슬롯 및 학과 과정 대 강의실의 조합을 생성함으로써, 학과 과정별로 복수의 스케줄링 케이스를 생성할 수 있다.

[0028] 예를 들어, 설정된 인스턴스는 미리 설정된 휴리스틱 알고리즘에 입력되어 초기 스케줄링 케이스가 생성될 수 있다. 초기 스케줄링 케이스는 한번에 하나의 학과 과정에만 할당되어 생성될 수 있으며, 재할당될 수 없다. 알고리즘은 모든 학과 과정이 할당된 후에 종료되므로, 최적의 솔루션을 보장할 수 없더라도 더 나은 솔루션을 찾는 데 많은 시간을 절약할 수 있다. 더 나은 솔루션을 찾기 위해 다른 알고리즘 안내하는 빠르게 사용가능한 솔루션이 제공될 수 있다.

[0029] 성능을 증가시키는 솔루션의 다양성을 유지하기 위하여, 원래 솔루션의 새로운 이웃 솔루션을 생성할 수 있다. 예를 들어, TS₁에 C₁을 할당한 원래의 스케줄링 케이스에 대하여, TS₂에 C₁을 할당하여 이웃 스케줄링 케이스를 생성할 수 있다.

[0030] 즉, 학과과정(C)과 타임슬롯(TS)의 조합은 하기 수학적식으로 나타낼 수 있다.

수학식 2

$$\begin{aligned} &C_1TS_1, C_1TS_2, \dots, C_1TS_M \\ &C_2TS_1, C_2TS_2, \dots, C_2TS_M \\ &\dots \\ &C_NTS_1, C_NTS_2, \dots, C_NTS_M \end{aligned}$$

[0031]

[0032] 그리고, 학과과정(C)과 강의실(R)의 조합은 하기 수학식으로 나타낼 수 있다.

수학식 3

$$\begin{aligned} &C_1R_1, C_1R_2, \dots, C_1R_O \\ &C_2R_1, C_2R_2, \dots, C_2R_O \\ &\dots \\ &C_NR_1, C_NR_2, \dots, C_NR_O \end{aligned}$$

[0033]

[0034] 이와 같이 생성된 학과과정(C)과 타임슬롯(TS)의 조합과 학과과정(C)과 강의실(R)의 조합은 도 2에 도시된 바와 같이 결합될 수 있다. 이를 통해, 타임슬롯과 강의실 둘 다에서 솔루션의 다양성을 증가시켜 성능이 향상될 수 있다.

[0035] S130 단계에서, 학과 과정 스케줄링 장치는, 생성된 복수의 스케줄링 케이스 별로 적합점수를 산출한다.

[0036] S140 단계에서, 학과 과정 스케줄링 장치는, 학과 과정별로 적합점수가 가장 큰 스케줄링 케이스를 선택하여 학과 과정별 스케줄을 결정한다.

[0037] 적합점수는 하기 수학식을 이용하여 산출될 수 있다.

수학식 4

$$\begin{aligned} Score &= S_{hard} + S_{soft} \\ &= 0 + S_{soft} \\ &= a_1S_{S1} + a_2S_{S2} + a_3S_{S3} + a_4S_{S4} + a_5S_{S5} \end{aligned}$$

[0038]

[0039] 여기서, S_{hard} 는 하드 제약조건의 만족여부에 따라 산출되는 점수로, 기본적으로 0이다. 왜냐하면, 하드 제약조건을 불만족하는 스케줄링 케이스는 실행 불가능하도록 음수가 부여되기 때문이다. 이를 통해, 모든 스케줄링 케이스가 실행 가능하다는 것이 보장될 수 있다.

[0040] 그리고, S_{soft} 는 소프트 제약조건에 만족여부에 따라 산출되는 점수로, $S_1 \sim S_5$ 의 5개 소프트 제약조건의 만족여부에 따라 설정된 점수의 합이다. 계수 $a_1 \sim a_5$ 는 각 소프트 제약조건의 보상 및 패널티 레벨을 나타낸다.

[0041] 소프트 제약조건을 만족하는 경우, S_{soft} 는 계수를 달리하여 양수값으로 보상이 주어질 수 있다. 반대로, 소프트

제약조건을 불만족하는 경우, S_{soft} 는 상응하는 가중치의 음수값으로 패널티가 주어질 수 있다. 일반적으로, $a_1=a_2=a_3=a_4=a_5=1(\text{보상})/-1(\text{패널티})$ 이다.

[0042] 다음과 같이 5개의 하드 제약조건(H1 ~ H5)이 주어질 수 있다.

[0043] H1: 하나의 강의실은 동일한 타임슬롯에서 최대 하나의 학과 과정을 수용할 수 있음

[0044] H2: 강의실은 학과과정의 기능적 요구사항을 충족해야 함

[0045] 예를 들어, 일부 학과 과정은 다른 학과 과정에서는 필요 없는 컴퓨터와 같은 장치 및 기능성 기계가 필요함

[0046] H3: 강의실의 수용 가능 인원수는 신청 학생의 수 이상이어야 함

[0047] H4: 한 교수가 동일한 타임슬롯에서 최대 하나의 학과 과정을 가르칠 수 있음

[0048] H5: 한 학생이 동일한 타임슬롯에서 최대 하나의 학과 과정에 참석할 수 있음

[0049] 그리고, 다음과 같이 5개의 소프트 제약조건(S1 ~ S5)이 주어질 수 있다.

[0050] S1: 교수는 선호하는 강의실이 하나 있음

[0051] S2: 교수는 각 학과 과정에 대하여 선호하는 타임슬롯을 가지고 있음

[0052] S3: 교수는 순차 모드로 가르치는 것을 선호함

[0053] S4: 학생은 순차 모드로 공부하는 것을 싫어함

[0054] S5: 학과 과정과 강의실은 학과가 동일해야 함

[0055] 이와 같이 주어진 제약조건의 만족 여부가 복수의 스케줄링 케이스 각각에 대하여 판단될 수 있으며, 만족 여부에 따라 기 설정된 점수가 복수의 스케줄링 케이스 별로 부여될 수 있다.

[0056] 도 3은 하드 제약조건의 만족 여부를 판단하는 의사 코드를 나타내고, 도 4는 소프트 제약조건의 만족 여부를 판단하는 의사 코드를 나타낸다.

[0057] 도 3 및 도 4를 참조하면, 제약 조건을 불만족하면, 미리 설정된 하나의 음수값이 더해짐으로써 패널티를 받게 된다. 제약 조건을 만족하면, 전술한 수학식 4와 같이, 하나의 양수값으로 보상을 받게 된다. 하드 제약조건을 불만족 한다는 것은 솔루션이 실행 불가능함을 의미한다.

[0058] 도 3 및 도 4에 도시된 바와 같이, 학과과정(C)의 인스턴스를 이용하여 상응하는 제약조건의 제어가 이루어질 수 있다.

[0059] 도 5 및 도 6은 복수의 스케줄링 케이스 각각에 대하여 순차적으로 연산을 수행하여 허용되지 않는 케이스를 필터링하여 적합점수가 가장 큰 스케줄링 케이스를 산출하는 알고리즘을 의사 코드로 나타낸 것이다. 특히, 도 5는 Tabu search를 이용한 필터링 알고리즘을 나타내고, 도 6은 Simulated Annealing을 이용한 필터링 알고리즘을 나타낸다. Tabu search는 로컬 최적에 빠지는 것을 회피하는 전역 솔루션을 찾는 데 사용될 수 있다. 그리고, Simulated Annealing은 여러 단계를 거쳐 강한 솔루션을 찾는 데 사용될 수 있다. 이와 같은 방식은 솔루션이 Tabu search와 Simulated Annealing의 장점을 모두 고려하고 잠재적인 다음 단계의 조합을 선택할 수 있는 다양성을 유지하도록 만든다.

[0060] 본 발명의 실시예에 따른 학과 과정 스케줄링 장치는 도 5 및 도 6의 두 필터링 알고리즘을 병렬로 동시에 수행하여 복수의 스케줄링 케이스 각각에 대한 연산 수행을 통해 학과 과정 별 최적의 스케줄을 도출할 수 있다.

[0061] 도 7은 본 발명의 실시예에 따른 학과 과정 스케줄링 장치의 구성을 개략적으로 예시하여 나타낸 도면이다.

[0062] 도 7을 참조하면, 본 발명의 실시예에 따른 학과 과정 스케줄링 장치는 프로세서(10), 메모리(20), 통신부(30) 및 인터페이스부(40)를 포함한다.

[0063] 프로세서(10)는 메모리(20)에 저장된 처리 명령어를 실행시키는 CPU 또는 반도체 소자일 수 있다.

[0064] 메모리(20)는 다양한 유형의 휘발성 또는 비휘발성 기억 매체를 포함할 수 있다. 예를 들어, 메모리(20)는 ROM, RAM 등을 포함할 수 있다.

[0065] 예를 들어, 메모리(20)는 본 발명의 실시예에 따른 학과 과정 스케줄링 방법을 수행하는 명령어들을 저장할 수

있다.

[0066] 통신부(30)는 통신망을 통해 다른 장치들과 데이터를 송수신하기 위한 수단이다.

[0067] 인터페이스부(40)는 네트워크에 접속하기 위한 네트워크 인터페이스 및 사용자 인터페이스를 포함할 수 있다.

[0068] 한편, 전술된 실시예의 구성 요소는 프로세스적인 관점에서 용이하게 파악될 수 있다. 즉, 각각의 구성 요소는 각각의 프로세스로 파악될 수 있다. 또한 전술된 실시예의 프로세스는 장치의 구성 요소 관점에서 용이하게 파악될 수 있다.

[0069] 또한 앞서 설명한 기술적 내용들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예들을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 하드웨어 장치는 실시예들의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

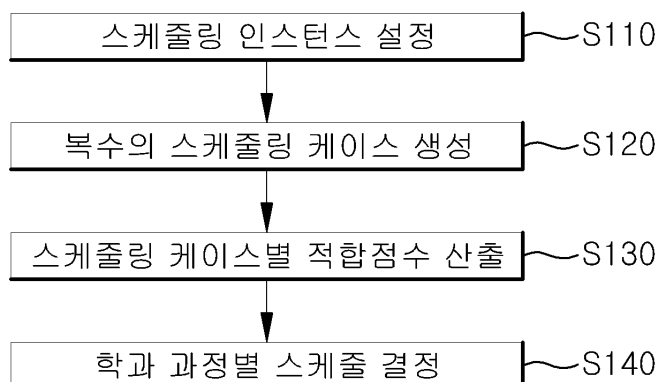
[0070] 상기한 본 발명의 실시예는 예시의 목적을 위해 개시된 것이고, 본 발명에 대한 통상의 지식을 가지는 당업자라면 본 발명의 사상과 범위 안에서 다양한 수정, 변경, 부가가 가능할 것이며, 이러한 수정, 변경 및 부가는 하기의 특허청구범위에 속하는 것으로 보아야 할 것이다.

부호의 설명

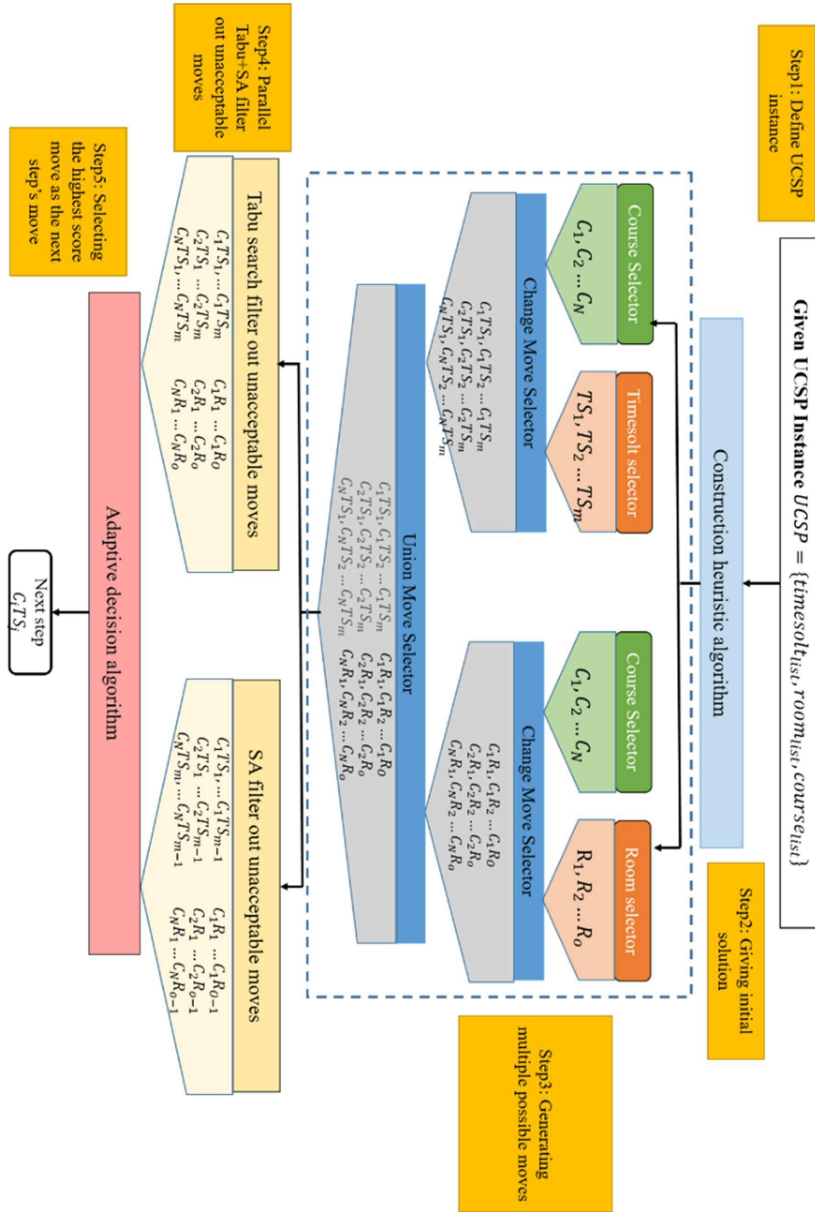
[0071] 10: 프로세서
20: 메모리
30: 통신부
40: 인터페이스부

도면

도면1



도면2



도면3

1. Given the UCSP instance solution course list $course_{list} = \{C_1, C_2, \dots, C_i, \dots, C_N\}$
2. If $C_i, course_{type} = C_i, room, room_{type}$:
3. Break **H2 CourseTypeHardConflict**.
4. If $C_i, student_{number} > C_i, room, size$:
5. Break **H3 RoomSizeHardConflict**.
6. Doing join operation $[course_{list}, course_{list}] = \{C_1 C_2, C_1 C_3, \dots, C_1 C_N, \dots, C_i C_j, \dots, C_{N-1} C_N\}$
7. Do $C_i C_j$ in $[course_{list}, course_{list}]$:
8. If $C_i, timesolt = C_j, timesolt$ and $C_i, room = C_j, room$ and $C_i, id < C_j, id$:
9. Break **H1 RoomTimeHardConflict**.
10. If $C_i, timesolt = C_j, timesolt$ and $C_i, faculty = C_j, faculty$ and $C_i, id < C_j, id$:
11. Break **H4 TeachingHardConflict**.
12. If $C_i, timesolt = C_j, timesolt$ and $C_i, student_{group} = C_j, student_{group}$ and $C_i, id < C_j, id$:
13. Break **H5 StudentHardConflict**.

도면4

1. Given the UCSP instance solution content course list $course_{list} = \{C_1, C_2, \dots, C_i, \dots, C_N\}$
2. If $C_i.prefer_{time}.dayofweek = C_i.timesolt.dayofweek$ or $C_i.prefer_{time}.start_{time} \neq C_i.timesolt.start_{time}$:
3. Break **S2 FacultyTimePreference**.
4. If $C_i.department == C_i.room.department$:
5. Satisfy **S5 CourseDepartmentConsistency**.
6. Doing join operation $[course_{list}, course_{list}] = \{C_1C_2, C_1C_3, \dots, C_1C_N, \dots, C_iC_j, \dots, C_{N-1}C_N\}$
7. Do C_iC_j in $[course_{list}, course_{list}]$:
8. If $C_i.faculty = C_j.faculty$ and $C_i.id < C_j.id$:
9. If $C_i.room \neq C_j.room$:
10. Break **S1 FacultyRoomStability**.
11. If $C_i.faculty = C_j.faculty$ and $C_i.timesolt.dayofweek = C_j.timesolt.dayofweek$:
12. If $C_i.timesolt.end_{time} - C_j.timesolt.start_{time} < 12$ hours:
13. Satisfy **S3 FacultyTeachingMode**.
14. If $C_i.subject = C_j.subject$ and $C_i.student_{group} = C_j.student_{group}$ and $C_i.course_{type} = C_j.course_{type}$ and $C_i.timesolt.dayofweek == C_j.timesolt.dayofweek$:
15. If $C_i.timesolt.end_{time} - C_j.timesolt.start_{time} < 12$ hours:
16. Break **S4 StudentStudyingMode**.

도면5

Input: Possible next steps solution $PS = \{C_1TS_1, C_1TS_2 \dots C_1TS_N, \dots, C_1R_1 \dots C_1R_o\}$

Output: Acceptable next step solution $AP \in PS$.

Defined current solution i ,

1. Randomly generate one solution $i \in PS$, set best solution $s = i$, Tabu list $H = \{\}$ with a 2% Entity Object length, move step $ms = 7$, and set iteration $k = 0$. Calculate the fitness score $f(s)$ using Eq. (1).
2. While not stop:
3. $A = N(i, H)$ #generate the neighbors of the solution set A from H.
4. $i = \text{SelectionBestSolution}(A)$ #set the current solution i
5. Calculate the fitness score $f(i)$ using Eq. (1).
6. Update the H considering Tabu list length and move step ms . # The Tabu list will be full as a length of 2% of Entity Object, and one Tabu object cannot be used before seven steps.
7. if $f(i) < f(s)$:
8. $f(s) = f(i)$
9. End if
10. $k = k + 1$
11. End While
12. $AP = s$

도면6

Input: Possible next steps solution $PS = \{C_1TS_1, C_1TS_2 \dots C_1TS_N, \dots, C_1R_1 \dots C_1R_o\}$

Output: Acceptable next step solution $AP \in PS$.

Define starting temperature $T_s = 1000$, and the cooling ratio is 0.99.

1. Randomly generate one solution $i \in PS$, setting best solution $s = i$, and calculate its fitness score $f(s)$ by Eq. (1).
2. While not stop
3. Generate a random neighbor s' , and calculate $f(s')$ by Eq. (1)
4. $\Delta E = f(s) - f(s')$
5. If $\Delta E < 0$ then $s = s'$
6. Else accept s' with probability $e^{\frac{-\Delta E}{T_s}}$
7. Update $T_s = G(T_s)$
8. Until $T_s < 1$
9. $AP = s$

도면7

