

자원 제약이 있는 프로젝트 스케줄링을 위한 효율적인 유전알고리즘

Efficient Genetic Algorithm for Resource Constrained Project Scheduling Problem

이상욱

목원대학교 정보통신공학과

Sangwook Lee(slee@mokwon.ac.kr)

요약

자원 제약이 있는 프로젝트 스케줄링 문제는 자원의 양은 제한되어 있고 작업들 간에 선행조건이 있는 일정계획 문제로서 NP-hard 문제 중에 하나로 알려져 있다. 이러한 문제는 결정론적인 방법을 사용해서는 주어진 시간 내에 최적해를 구하기 어렵기 때문에 근사 최적해를 빠른 시간에 구할 수 있는 휴리스틱 방법을 이용한다. 본 논문에서는 자원 제약이 있는 프로젝트 스케줄링 문제를 효율적으로 해결할 수 있는 유전알고리즘을 소개한다. 제안한 유전알고리즘은 스키마 이론을 적용한 교차 연산자와 실세계 토너먼트 선택 전략을 이용하였다. 표준 문제에 실험한 결과는 제안한 알고리즘이 기존의 알고리즘 보다 우수함을 보여주었다.

■ 중심어 : | 자원 제약이 있는 프로젝트 스케줄링 문제 | 유전알고리즘 | 스키마 | 실세계 토너먼트 선택전략 |

Abstract

Resource constrained project scheduling problem with multiple resource constraints as well as precedence constraints is well-known as one of the NP-hard problem. Since these problems can't be solved by the deterministic method during reasonable time, the heuristics are generally used for getting a sub-optimal during reasonable time. In this paper, we introduce an efficient genetic algorithm for resource constrained project scheduling problem using crossover which is applying schema theory and real world tournament selection strategy. Experimental results showed that the proposed algorithm is superior to conventional algorithm.

■ keyword : | Resource Constrained Project Scheduling | Genetic Algorithm | Schema | Real World Tournament Selection |

I. 서론

프로젝트 관리의 의사결정은 정보의 질과 양에 의해 직접적으로 결정되며 스케줄링 방법에 의해 간접적으로 결정된다. 여기서 시간 요소는 프로젝트 스케줄링에 있어서 가장 큰 비중을 차지하며 프로젝트의 구성에도 항상 연계되어 프로젝트 관리의 가장 큰 영향을 미치는

요소이다. 프로젝트의 각 액티비티(activity)들은 이전에 특정한 액티비티가 완료된 후에만 진행할 수 있는 선행 제약 조건을 가지고 있다. 일반적으로 프로젝트 스케줄링의 목적은 주어진 선행 제약 조건 아래 프로젝트 완료 시간을 최소화 하는데 있다.

조금 더 현실적으로 얘기하면, 프로젝트를 수행하는

* 이 논문은 2009년 목원대학교 신입교원 정착연구 지원사업에 의하여 연구되었음.

접수번호 : #110503-007

접수일자 : 2011년 05월 03일

심사완료일 : 2011년 06월 13일

교신저자 : 이상욱, e-mail : slee@mokwon.ac.kr

데 있어서 기구, 장비, 기계 및 인부들과 같은 자원이 필요하며 각 액티비티들은 이러한 자원들을 공유하여 사용한다. 프로젝트에 사용할 수 있는 각 자원들은 한정되어 있으며, 따라서 선행 제약 조건을 만족하여 시작할 준비가 되어 있는 액티비티일지라도 필요한 자원이 부족하다면 다른 액티비티가 종료되어 필요한 만큼의 자원이 남을 때 까지 시작할 수 없게 된다. 이러한 형태의 문제가 바로 익히 잘 알려진 자원 제약이 있는 프로젝트 스케줄링 문제 (Resource Constrained Project Scheduling Problem, RCPSP) 이다.

RCPSP 모델은 1966년 Kelly [1]에 의해 처음 소개된 이후 소프트웨어, 군 산업 프로젝트 등과 같이 대규모 프로젝트를 개발하거나 디자인하는 경우에 많이 적용되어 연구되어 왔다. 연구 초기에는 수학적 논리를 적용한 결정론적인 방법으로 해결하려 노력하였으나, Blazewicz [2]에 의해 RCPSP가 NP-hard 문제임이 알려진 이후에는 발견적 (Heuristics, 휴리스틱) 기법으로 해결하려는 노력이 많이 이루어져 왔다 [3]. 본 논문에서는 대표적인 메타 휴리스틱 기법 중에 하나인 유전 알고리즘을 이용하여 RCPSP를 해결하기 위한 효율적인 유전 연산자 및 선택전략을 소개하고, 기존 알고리즘과 비교를 통해 우수성을 보여주고자 한다.

본 논문은 제2장에서 자원 제약이 있는 프로젝트 스케줄링의 수학적 모형에 대해 설명하고, 제3장에서는 제안하는 유전 알고리즘을 구체적으로 기술한다. 그리고 제4장에서 기존의 기법들과의 비교 실험을 통해 제안하는 방법의 성능을 보이며, 마지막으로 결론을 맺는다.

II. 자원 제약이 있는 프로젝트 스케줄링

프로젝트는 액티비티의 집합으로 이루어져 있다. RCPSP 식 문제 표현에서 프로젝트가 n 개의 액티비티로 이루어져 있으면, 액티비티 1과 n 은 가상 액티비티로 정의하며 프로젝트의 시작과 끝을 나타낸다. 각 액티비티들은 ‘선행제약조건’과 ‘자원제약조건’ 이라는 2 종류의 제약조건과 결부되어 있다. 첫 번째 타입의 선

행제약조건은 어떤 액티비티가 시작되기 위해서 다른 액티비티들이 완료되어야 한다는 조건이며, 액티비티 i 가 시작되기 위해서 선행되어 완료되어야 하는 액티비티들의 집합을 액티비티 i 의 선행자라고 부르며 S_i 로 표현하기로 한다. 액티비티 i 는 S_i 에 속한 액티비티들이 완료된 이후에 시작할 수 있다. 두 번째 타입의 자원 제약조건은 한정된 자원 아래 각 액티비티들이 자원을 사용해야 하는 조건이다. m 개의 재사용 가능한 자원들이 $b_k (k=1, 2, \dots, m)$ 양만큼 제한되어 있다. 만약 액티비티 i 가 필요한 자원이 r_{ik} 라면, 액티비티 i 가 수행되는 시간 (d_i) 동안 그만큼의 자원을 할당 받아야 하며, 그 조건을 만족하지 못한다면 선행제약조건을 만족하더라도 액티비티 i 는 수행될 수 없다.

RCPSP에서 변수 d_i, r_{ik}, b_k 는 미리 결정되어 있으며 정수 값을 가지고 있다. 더미 액티비티인 1과 n 에 대한 액티비티 소요시간은 $d_1 = d_n = 0$ 이며 필요한 자원 또한 0이다. RCPSP 문제는 각 액티비티들이 선행 제약조건과 자원제약조건을 만족시키면서 수행되어 모든 액티비티들이 완료되는데 걸리는 시간을 최소화 하는 것을 목표로 한다.

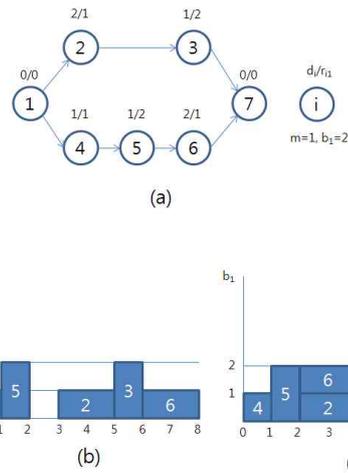


그림 1. RCPSP 간단한 예제 및 답

[그림 1]은 RCPSP 문제의 간단한 예제와 그에 대한 답을 보여준다. 이 프로젝트는 7개의 액티비티들로 이

루어져 있으며, 한계 값이 2인 자원 1개로 이루어져 있다. [그림 1](a)는 액티비티들 간의 선행조건과 각 액티비티들에 대해 수행하는데 걸리는 시간과 필요한 자원의 크기를 보여주고 있다. [그림 1](b)는 이 문제에 대한 한가지의 유효한 답을 보여주고 있으며, [그림 1](c)는 최적화된 답을 보여주고 있다.

이론적으로 자원 제약이 있는 프로젝트 스케줄링 문제 (RCPSP)는 아래와 같이 표현할 수 있다.

$$\min f_n \tag{1}$$

s.t.

$$f_j - f_i \geq d_i, \quad \forall_j \in S_i \tag{2}$$

$$\sum_{i \in A_k} r_{ik} \leq b_k, \quad k = 1, 2, \dots, m \tag{3}$$

$$t_i \geq 0, \quad i = 1, 2, \dots, n \tag{4}$$

여기서 t_i 는 액티비티 i 의 시작시간, d_i 는 액티비티 i 의 작업시간, S_i 는 액티비티 i 의 선행 액티비티들의 집합, r_{ik} 는 액티비티 i 가 요구하는 자원의 양, b_k 는 이용할 수 있는 k 자원의 양, A_k 는 시간 t_i 시점에서 진행되고 있는 액티비티들의 집합, 그리고 m 은 자원의 개수이다. (1)은 프로젝트 스케줄링 시간을 최소화 하는 목적함수를 나타내고, (2)는 선행 제약 조건을 나타내고, (3)은 자원 제약 조건을 나타내며, (4)는 각 액티비티 시간은 음수가 될 수 없음을 나타낸다.

III. 유전 알고리즘

유전 알고리즘 (Genetic algorithm, GA)은 자연세계의 진화과정을 모방하여 1975년 John Holland에 의해 개발된 최적화 알고리즘이다 [4]. GA는 자연 선택의 원리와 생물 유전학에 기본 이론을 두고 병렬적이고 진역적인 탐색을 통해 해를 찾는다. 다윈의 적자생존 (Survival of the fittest) 이론에 토대를 두고 선택전략을 통해 회를 거듭할수록 우수한 해를 찾는 알고리즘이다. [표 1]은 GA의 슈도코드를 보여주고 있다.

표 1. GA 슈도코드

```

t=0, 초기 해집단 Pt 생성
해집단 Pt 평가
중요조건을 만족할때까지
{
선택된 Pt의 몇몇 해를 Pt+1 복사
선택된 Pt의 몇몇 해에 교차연산을 하여 Pt+1 이동
선택된 Pt의 몇몇 해에 변이연산을 하여 Pt+1 이동
새로운 해집단 Pt+1 평가
Pt = Pt+1
}
    
```

1. 해 표현 (Representation)

유전 알고리즘을 이용한 최적화 문제에서 문제에 적합한 해(염색체) 표현을 찾는 것은 매우 중요하다. 프로젝트 스케줄링 문제의 해 표현에는 많은 방법들이 있다 [5]. 본 논문에서는 순서가 정해진 작업들의 조합을 하나의 해로 표현하였으며 표현법은 다음과 같다.

$$s_i = [g_1, g_2, \dots, g_j, \dots, g_c]$$

여기서 해 s_i 는 현재 집단에서 i 번째 개체를 나타내고, g_j 는 j 번째 위치한 작업을 나타낸다. g_j 는 각 액티비티들의 인덱스를 나타내며, 식(4)의 i 값을 뜻한다. 따라서 g_j 는 1에서 n 까지의 값을 가지며, 서로 중복될 수 없다. [그림 1](a)와 같이 7개의 액티비티들은 가진 RCPSP 문제에 대한 [그림 1](b)의 해를 염색체로 표현하면 다음과 같다.

$$s_i = [4, 5, 2, 3, 6]$$

이 표현법의 탐색 공간은 액티비티들의 모든 조합들이다. 그러나 선행 제약 조건 때문에 가능한 공간은 그 중 일부뿐이다.

2. 초기 해집단 생성 (Initialization)

일반적으로 GA에서 초기 해집단 구성은 무작위 방법을 통해 이루어진다. 만약 해집단 초기화 시점에서 문제 해결에 유리하게 작용할 수 있는 사전정보를 가지고 있다면 그 내용을 해집단 초기화 방법에 접목하여 해집

단을 구성할 수도 있다. 또는 해집단을 무작위로 구성하면 유효한 해를 만들 수 없는 경우도 있다. 본 논문에서 다루는 RCPSP 문제는 선행 제약 조건으로 인하여, 무작위적인 방법으로는 유효한 해를 만들어내기 어렵다. 이런 경우 일반적으로 유효한 해를 만들기 위해 2가지 방법이 사용된다. 첫 번째는 무작위로 만들어진 해에 보정 (Repair) 연산을 하는 것이고, 두 번째는 항상 제약 조건을 만족하는 해를 만들어 내는 초기화 전략을 사용하는 것이다. 본 논문에서는 초기 해집단 생성시 단일 패스 전략을 사용하여 항상 유효한 해를 만들도록 하였다 [6]. 해 표현을 해석하여 일정 계획을 세울 때 탐색체의 첫 번째 액티비티부터 시작하여 오른쪽으로 하나씩 채워 나감으로써 프로젝트 스케줄링이 이루어진다. 해를 구성하는 액티비티를 할당할 때 선행 제약 조건을 만족하는 액티비티들의 집합을 만든 다음 그 중에 무작위로 하나의 액티비티를 선택하여 할당한다. 그런 후 다음 상황에 맞는 선행 제약 조건을 만족하는 액티비티들의 집합을 다시 구성하고 위의 과정을 반복한다. 모든 액티비티들의 할당이 끝나면 초기화 과정이 완료된다.

3. 해집단 평가 (Evaluation)

다윈의 적자생존 법칙에 입각한 GA의 선택전략을 적용하려면 해집단에 대한 평가가 선행되어야만 한다. 해에 대한 평가는 탐색체를 디코딩하여 문제에서 최적화 하고자 하는 목적에 본 탐색체가 얼마만큼의 값을 가지는지 계산하는 것이다. RCPSP 문제에서 평가는 탐색체로부터 해석하여 구해진 프로젝트 완료 시간 값이 된다. 일반적으로 GA의 평가 값을 적합도 값 (fitness value)이라 하고, 최대화 하는 것을 목표로 한다. 그러나 RCPSP 문제에서는 프로젝트 완료 시간을 최소화 하는 것이 목표이므로 목적함수를 최대화 하는 문제로 아래와 같이 조정한다.

$$fit(s_i) = \frac{1}{o(s_i)}$$

여기서 $fit(\cdot)$ 은 적합도 함수, $o(\cdot)$ 은 프로젝트 완료 시간을 구하는 목적 함수이다.

4. 스키마 교차 (Schema Crossover)

GA에서 교차 연산은 두 부모로의 유전자를 자식에게 골고루 물려주는 역할을 한다. 선택전략과 맞물려 우수한 형질을 물려받은 자식이 다음 세대에 살아남을 것이다.



그림 2. 스키마 교차

교차 연산의 가장 기본적인 것은 1점 교차와 2점 교차이다. 그러나 RCPSP와 같이 선행제약조건이 있는 문제에서는 1점, 2점 교차는 유효하지 못한 해를 생성할 확률이 높다. 그런 경우 보정 작업을 필히 수행하여야 하고 보정작업을 마치면 부모의 유전 형질을 전혀 반영하지 못한 자식이 나올 수도 있다.

이러한 문제점을 보완하기 위하여 본 논문에서는 John Holland가 GA의 탐색 능력을 설명할 때 거론한 스

키마 이론 [4]에 기반한 효율적인 교차 연산자를 제안한다. [그림 2]는 본 논문에서 제안하는 '스키마 교차'에 대한 내용을 보여주고 있다. 두 부모가 공통적으로 연속적인 유전자를 가지고 있으면, 그 배열을 스키마로 정의하고 저장한다. 두 부모의 유전자 배열을 비교 검색할 때 배열의 정방향 뿐만 아니라 역방향도 함께 검색하여 연속하는 배열을 찾는다. 만일 정방향, 역방향 모두 연속하지 않으면 하나의 유전자를 스키마로 저장한다.

이렇게 만들어진 스키마들을 이용해서 자식을 만든다. 스키마들의 조합을 해집단 생성에서 설명한 단일 패스 전략을 적용하여 다음과 같이 자식 염색체를 만드는 것이다. 자식 해를 구성하는 스키마를 할당할 때 선행 제약 조건을 만족하는 스키마들의 집합을 만든 다음 그 중에 무작위로 하나의 스키마를 선택하여 할당한다. 그런 후 다음 상황에 맞는 선행 제약 조건을 만족하는 스키마들의 집합을 다시 구성하고 위의 과정을 반복한다. 모든 스키마들의 할당이 끝나면 자식 해 생성 과정이 완료된다.



그림 3. 돌연변이 연산을 통해 얻은 이웃 해들

5. 돌연변이 (Mutation)

GA에서 돌연변이 연산은 교차와 선택에 의해 해가 성숙하지 못한 상태에서 빠르게 수렴하는 것을 방지하기 위한 연산이다. 본 연구에서는 제안한 스키마 교차 연산자를 Gen의 결과와 비교하기 위하여, Gen의 논문에서 사용된 것과 같은 변이 연산자를 사용하였다 [6]. 돌연변이 수행에 해당하는 염색체에 무작위로 돌연변이 지점을 잡고, 이웃한 두개의 유전자와의 조합을 이용해 이웃 해들을 만든다. 이웃 해들 중에서 제약 조건을 만족하는 해들을 모은 후 그 중 가장 적합도가 높은

해를 선택하여 부모와 대체한다. [그림 3]은 돌연변이 연산을 통해 얻은 이웃 해들을 보여준다.

6. 선택 (Selection)

본 실험에서 GA의 선택전략으로 실세계 토너먼트 선택 전략 (Real World Tournament Selection, RWTS)을 사용하였다 [7]. 실세계 토너먼트 선택전략은 현실 세계의 토너먼트 스포츠 게임처럼 해 집단에 있는 모든 해들을 이웃하는 해들끼리 쌍을 이루어 경쟁을 시키고, 경쟁에서 승리한 해들을 다음 세대의 해 집단에 포함시키며, 승리한 해들은 또 다시 이웃하는 다른 승리한 해들끼리 쌍을 이루어 경쟁을 시키는 과정을 반복 수행하여 해 집단을 형성하는 방법이다. 문헌의 많은 실험 결과, 실세계 토너먼트 선택전략은 룰렛 휠, 토너먼트 선택 전략 등과 같은 다른 선택 전략에 비해 성능이 우수함이 입증되었다 [7].

IV. 실험

본 논문에서 제안하는 알고리즘의 성능을 기존의 알고리즘과 비교 평가하기 위하여 Gen [6]과 Hartmann [8]이 제안한 알고리즘과 비교 실험하였다. Gen이 제안한 알고리즘은 제안한 알고리즘과 동일한 작업들의 조합을 해 표현으로 사용했으며, [그림 3]에서 설명한 돌연변이 연산을 사용하고 PMX(Partially Mixed Crossover) 교차 연산을 사용하였다. Hartmann이 제안한 알고리즘은 Holland [4]가 제안한 2점 교차 연산과 1점 swap 돌연변이 연산을 사용했으며, 선행제약조건 위반 문제를 해결하기 위해 해 표현법을 '우선순위 규칙 (Priority Rule Representation)'에 기반한 방법을 사용하였다. 보다 정확한 비교 실험을 위해 Beasley의 OR-Library [9]에 있는 patterson set 데이터를 이용하였으며, 총 10개의 소형 문제, 10개의 중형 문제와 10개의 대형 문제에서 실험을 수행하였다. 제안하는 알고리즘은 C++를 이용해서 프로그램 되었고 Pentium IV 3.4G CPU와 2.0 Gbyte Ram 컴퓨터 상에서 실험을 하였다.

표 2. patterson 문제 실험 결과

Inst.	n	m	opt.	Gen			Hartmann			The proposed algorithm		
				best	freq.	error	best	freq.	error	best	freq.	error
pat1	14	3	19	19	8	1	19	9	1	19	10	0
pat2	7	3	7	7	10	0	7	10	0	7	10	0
pat3	13	3	20	20	9	1	20	10	0	20	10	0
pat4	22	3	6	6	10	0	6	10	0	6	10	0
pat5	22	3	7	7	10	0	7	10	0	7	10	0
pat6	22	3	8	8	10	0	8	10	0	8	10	0
pat7	9	1	8	8	10	0	8	10	0	8	10	0
pat8	9	1	11	11	10	0	11	10	0	11	10	0
pat9	18	1	19	19	8	1	19	9	1	19	10	0
pat10	8	2	14	14	10	0	14	10	0	14	10	0
pat51	22	3	25	25	9	1	25	9	1	25	9	1
pat52	27	3	27	27	10	0	27	10	0	27	10	0
pat53	22	3	28	28	10	0	28	10	0	28	10	0
pat54	22	3	50	50	10	0	50	8	1	50	10	0
pat55	22	3	29	29	10	0	29	10	0	29	10	0
pat56	22	3	27	27	10	0	27	10	0	27	10	0
pat57	22	3	21	21	9	1	21	9	1	21	9	1
pat58	27	3	35	35	2	2	35	4	2	35	2	2
pat59	27	3	31	31	2	2	31	6	1	31	2	2
pat60	27	3	39	40	0	2	40	0	2	39	1	2
pat101	51	3	75	75	1	6	75	1	4	75	5	4
pat102	51	3	83	84	0	11	84	0	11	84	0	8
pat103	51	3	56	57	0	4	57	0	4	56	2	2
pat104	51	3	79	79	2	6	79	4	5	79	5	4
pat105	51	3	76	77	0	5	77	0	6	76	4	3
pat106	51	3	60	62	0	8	61	0	6	62	0	4
pat107	51	3	78	79	0	11	78	2	6	78	3	7
pat108	51	3	61	63	0	8	62	0	5	62	0	4
pat109	51	3	60	63	0	6	63	0	6	60	2	4
pat110	51	3	50	52	0	5	50	2	3	51	0	4

본 실험에 사용된 유전 연산자들의 파라미터는 다음과 같다. 제안한 알고리즘의 경우 여러 번의 반복 실험을 통해 경험적으로 최적화된 연산 확률을 얻었으며, 교차 확률을 0.5, 돌연변이 확률을 0.3 과 같이 나왔다. 개체 수는 50, 세대 수는 100으로 설정하였고 동일한 조건에서의 비교 실험을 위해 Hartmann과 Gen의 알고리즘에서도 동일한 개체 수 및 세대 수에서 실험하였다. Hartmann은 2점 교차 연산자와 1점 돌연변이 연산자를 사용하였으며, 문헌에서 제시된 바와 같이 교차 확

률 및 돌연변이 확률을 각각 0.6, 0.05를 사용하였다 [8]. Gen은 PMX 교차와 Gen 돌연변이 연산자를 사용하였으며, 문헌에서 제시된 바와 교차 확률 및 돌연변이 확률은 각각 0.3, 0.3을 사용하였다 [6]. 선택전략의 경우 3가지 알고리즘 모두에 실제계 토너먼트를 사용하였다.

[표 2]는 patterson 데이터에서의 실험 결과를 보여주고 있다. 표에서 n은 액티비티의 수, m은 자원의 수, opt.는 최적의 프로젝트 완료 시간, best는 알고리즘이 구한 최적의 해, freq.는 알고리즘이 10번의 반복 실험

중 최적해를 구한 빈도, error는 알고리즘이 구한 가장 나쁜 해와 최적해 간의 오차를 나타낸다.

실험 결과를 보면 Hartmann 알고리즘과 제안한 알고리즘은 pat54에서 Gen알고리즘이 Hartmann 알고리즘에 비해 우수한 것을 제외하고는 항상 Gen의 알고리즘에 비해 두 알고리즘이 우수하다는 것을 알 수 있다. 따라서 보다 간편한 비교를 위해 Hartmann과 제안한 알고리즘 2개만 사용하여 비교하도록 하겠다.

표에서 볼드체로 표시가 되어 있는 것은 제안한 알고리즘과 Hartmann 알고리즘을 비교하여 더 우수한 성능을 보여주는 것을 뜻한다. 최적해를 찾는 성능을 비교해 볼 때, Hartmann 알고리즘이 최적해를 구하지 못한 pat60, pat103, pat105, pat109 문제 등의 총 4문제에서 제안한 알고리즘은 최적해를 구하였고, 반대로 pat110 문제에서는 제안한 알고리즘은 최적해를 못 구했으나 Hartmann 알고리즘은 최적해를 구하였다. 최적해를 구한 빈도수를 비교하면, 둘 다 최적해를 찾았을 경우, 제안한 알고리즘이 최적해를 더 많이 구한 경우가 9문제, Hartmann 알고리즘이 더 많이 구한 경우가 3문제이다. 10번 반복실험을 통해 얻은 결과의 안정성을 측정하는 척도가 되는 error를 비교해 보면, 제안한 알고리즘이 더 안정적인 경우가 10문제, Hartmann 알고리즘이 더 안정적인 경우가 3문제이다.

위와 같은 실험 결과를 이용해 분석한 내용을 보면 제안한 알고리즘이 기존의 알고리즘보다 우수한 성능을 보인다는 것을 알 수 있다. 이렇게 제안한 알고리즘이 기존의 알고리즘보다 우수한 성능을 보이는 이유는 '스키마 교차' 연산이 부모의 우수한 형질을 자손에게 효율적으로 전달하기 때문으로 생각된다. 일반적으로 RCPSP에 적용된 기존의 다른 유전 알고리즘들은 교차 연산을 수행하면 선행제약조건을 위반하여 유효한 해를 만들지 못하므로 보정 과정을 반드시 거쳐야 한다. 이러한 보정 과정은 돌연변이 연산과 같은 역할을 하여 탐색 공간에서 지역 탐색 (local search) 기능을 수행해야 할 교차 연산이 제 역할을 다하지 못하기 때문으로 판단된다. 이에 반해 본 논문에서 제안하는 스키마 교차는 보정 과정 없이 두 부모가 공통적으로 지닌 유전자 블록을 자손에게 그대로 전달하기 때문에 교차 연산

이 수행해야 할 지역 탐색 역할을 충실히 수행한다고 판단된다.

V. 결 론

본 논문에서는 자원 제약이 있는 프로젝트 스케줄링 문제를 효율적으로 해결할 수 있는 새로운 진화 알고리즘을 제안하였다. 제안하는 알고리즘은 스키마 이론을 바탕으로 하여 부모의 공통 유전자 블록을 자손에게 전달할 수 있는 교차 연산자와 현실 세계의 스포츠 토너먼트 제도를 본 딴 실세계 토너먼트 선택전략을 적용하였다. 기존의 방법들과의 비교 실험을 통해 상대적으로 우수한 결과 값을 찾았다는 것을 확인할 수 있었다.

참 고 문 헌

- [1] J. Kelley, "The critical path method: Resource planning and scheduling," In Muth, J. and G. Thompson, editors, *Industrial Scheduling*, pp.347-365, Prentice Hall, Englewood Cliffs, New Jersey, 1963.
- [2] J. Blazewicz, "Complexity of computer scheduling algorithms under resource constraints," In Proc. First meeting AFCE/SMF on Applied Mathematics, pp.169-178, 1978.
- [3] R. Alvarez-Valdes and J. Tamarit, "Heuristic algorithms for resource constrained project scheduling: A review and an empirical analysis," In Slowinski, R. and J. Weglarz, editors, *Advances in Project Scheduling*, pp.113-134, Elsevier Science Publishers, Amsterdam, 1989.
- [4] H. J. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, 1975.
- [5] R. Cheng, M. Gen, and Y. Tsujimura, "A tutorial survey of job-shop scheduling

problems using genetic algorithms: part I. Representation," International Journal of Computers and Industrial Engineering, Vol.30, No.4, pp.983-997, 1996.

- [6] R. Cheng and M. Gen, "Resource constrained project scheduling problem using genetic algorithms," International Journal of Intelligent Automation and Soft Computing, Vol.3, No.3, pp.273-286, 1997.
- [7] S. Lee, S. Soak, K. Kim, H. Park, and M. Jeon, "Statistical properties analysis of real world tournament selection in genetic algorithms," Applied Intelligence, Vol.28, No2, pp.195-205, 2008(4).
- [8] S. Hartmann, "A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling," Naval research logistics, Vol.45, No.7, pp.733-750, 1998.
- [9] <http://people.brunel.ac.uk/~mastjjb/jeb/info.html>

<관심분야> : 휴리스틱 알고리즘, 최적화, 데이터 마이닝, 유비쿼터스 컴퓨팅, 스마트폰 프로그래밍

저 자 소 개

이 상 욱(Sangwook Lee)

중신회원



- 2000년 2월 : 한국과학기술원 기계공학과(공학사)
- 2002년 2월 : 광주과학기술원 기전공학과(공학석사)
- 2007년 8월 : 광주과학기술원 정보기전공학부(공학박사)

- 2007년 9월 ~ 2008년 9월 : 조지아공과대학교 전산학과 박사후연구원
- 2008년 11월 ~ 2009년 2월 : 삼성전자 통신연구소 책임연구원
- 2009년 3월 ~ 2011년 3월 : 목원대학교 정보통신공학과 전임강사
- 2011년 4월 ~ 현재 : 목원대학교 정보통신공학과 조교수