

Generation of Pareto Sets based on Resource Reduction for Multi-Objective Problems Involving Project Scheduling and Resource Leveling

Woo-Jin Jeong · Sung-Chul Park · Dong-Soon Yim[†]

Department of Industrial Engineering, Hannam University

프로젝트 일정과 자원 평준화를 포함한 다목적 최적화 문제에서 순차적 자원 감소에 기반한 파레토 집합의 생성

정우진 · 박성철 · 임동순[†]

한남대학교 산업공학과

To make a satisfactory decision regarding project scheduling, a trade-off between the resource-related cost and project duration must be considered. A beneficial method for decision makers is to provide a number of alternative schedules of diverse project duration with minimum resource cost. In view of optimization, the alternative schedules are Pareto sets under multi-objective of project duration and resource cost. Assuming that resource cost is closely related to resource leveling, a heuristic algorithm for resource capacity reduction (HRCR) is developed in this study in order to generate the Pareto sets efficiently. The heuristic is based on the fact that resource leveling can be improved by systematically reducing the resource capacity. Once the reduced resource capacity is given, a schedule with minimum project duration can be obtained by solving a resource-constrained project scheduling problem. In HRCR, VNS (Variable Neighborhood Search) is implemented to solve the resource-constrained project scheduling problem. Extensive experiments to evaluate the HRCR performance are accomplished with standard benchmarking data sets, PSPLIB. Considering 5 resource leveling objective functions, it is shown that HRCR outperforms well-known multi-objective optimization algorithm, SPEA2 (Strength Pareto Evolutionary Algorithm-2), in generating dominant Pareto sets. The number of approximate Pareto optimal also can be extended by modifying weight parameter to reduce resource capacity in HRCR.

Keywords : Resource-Constrained Project Scheduling, Resource Leveling, Pareto Set, Heuristic Algorithm

1. 서론

자원 소요 비용을 최소화 하면서 동시에 프로젝트 종료일을 최소화하는 스케줄을 구하는 것은 프로젝트 관리에서 매우 중요한 과제이다. 일반적으로 자원 소요 비용

과 종료일은 상충 관계를 나타낸다. 종료일을 단축시키기 위해서는 많은 자원의 투입이 요구되고, 자원 비용을 줄이기 위해서는 종료일이 늘어나는 상황이 발생할 수 있다. 이러한 상황은 의사결정자로 하여금 쉽게 결정할 수 없는 문제에 봉착하게 만든다. 의사결정에 도움을 줄 수 있는 방법 중의 하나는 다양한 종료일에 따른 최소 비용의 스케줄을 제공하는 것이다. 최적화의 관점에서는 종료일과 자원 비용의 최소화라는 두 가지 목적을 고려하여 파레토 집합에 속하는 스케줄들을 구하는 것이다.

Received 7 May 2020; Finally Revised 25 June 2020;
Accepted 26 June 2020

[†] Corresponding Author : dsyim@hnu.kr

2000년 대 이후 파레토 집합을 효과적으로 구하기 위한 다목적 최적화 알고리즘들이 개발되어 왔다. 대부분의 알고리즘들은 유전자 알고리즘(GA : Genetic Algorithm)이나 입자 군집 최적화(PSO : Particle Swarm Optimization)와 같은 군집 기반의 메타 휴리스틱에 기초하고 있다. 우수한 성능을 가진 것으로 알려진 NSGA2(Non-dominated Sorting Genetic Algorithm)[3]와 SPEA2(Strength Pareto Evolutionary Algorithm)[21]는 유전자 알고리즘에 기초하고 있고, SMPSO(Speed-constrained Multi-objective Particle Swarm Optimization)[13]는 PSO에 기초하고 있다. 최근에는 부분 최적화에 기초한 다목적 GRASP(Greedy Randomized Adaptive Search Procedures)[16], 다목적 가변 이웃 탐색(VNS : Variable Neighborhood Search)[5]과 Pareto-iterated Local Search[4] 등이 개발되었다.

프로젝트 스케줄링에서 종료일과 자원 비용의 최소화는 보통 자원 제약 프로젝트 스케줄링(RCPSP : Resource-Constrained Project Scheduling Problem)[2,10]와 자원 평준화(resource leveling) 문제로 정의된다. RCPSP는 자원의 제약을 고려하여 종료일의 최소화를 모색하는 것이다. 자원 소요 비용의 최소화는 자원 평준화와 밀접한 관계를 가지고 있다. 자원 소요 비용을 감소시키는 방법 중의 하나는 자원 소요의 불균형을 억제하는데 있기 때문이다. 따라서, 종료일과 자원 소요비용의 최소화를 목적으로 하는 프로젝트 스케줄링 분야의 연구에 적용된 군집 기반 다목적 최적화 방법들은 RCPSP와 자원 평준화 문제를 복합적으로 고려하고 있다.

이러한 다목적 최적화 문제의 파레토 집합을 구하기 위해서는 자원의 제약을 입력으로 최소 종료일의 스케줄을 구하고 스케줄은 자원 평준화 방법에 의해 수정되는 기본적인 절차에 따른다. Ballestin and Blanco[1]의 연구에서는 군집을 구성하는 각 개체를 활동 리스트(activity list)[8]로 표현하고, S-SGS(Serial Schedule Generation Scheme)에 의해 디코딩 된 스케줄에 보다 빠른 종료일의 스케줄로 변환하기 위해 double justification[17]을 적용하였다. Hu and Flood[9]의 연구에서는 활동 리스트 표현과 S-SGS 디코딩을 사용하였고, 생성된 스케줄에 자원 소요비용을 줄이기 위한 자원 평준화 방법을 적용하였다. 두 연구 모두 SPEA2 알고리즘을 사용하여 어카이브 개체군(archive population)을 구성하는 다양한 개체들이 종료일과 자원의 비용 면에서 파레토 집합에 속하도록 유지한다. 종료일과 각 활동 시작 시간의 가중합의 다목적 최적화를 목적으로 한 Gomes[6]의 연구에서는 부분 최적화에 기초한 방법을 적용하였다.

그러나, 위에서 언급한 연구에서 생성되는 파레토 집합에 속하는 스케줄은 모두 동일한 자원의 용량 제약에서 구한 것으로 다양한 자원 사용과 종료일을 가지는 파레토 집합들을 구성하는데 한계가 있다. 동일한 자원 용량

에서는 서로 비슷한 스케줄만이 생성되기 때문이다. 의사결정자에게 도움을 주기 위해서는 다양한 종료일과 자원 비용을 가지는 스케줄들을 제공하여야 한다.

RCPSP는 자원의 용량을 입력으로 최소 종료일을 가져올 수 있는 스케줄의 생성을 목적으로 한다. 자원의 용량은 자원 비용과 직접적인 관계가 있다. 자원의 용량이 감소할수록 RCPSP의 해에는 자원 평준화가 이루어지고 이에 따라 자원 비용은 줄어드는 반면 종료일은 증가한다. 종료일의 증가로 인하여 자원 비용이 반드시 감소하지 않는 경우에도 자원 평준화에 따른 효과는 증가한다. 이러한 특성을 감안하면 서로 다른 자원의 용량 하에서 RCPSP에 의한 해들을 구하여 다양한 스케줄을 생성할 수 있다.

본 연구에서는 프로젝트 종료일과 자원 소요 비용의 최소화를 위한 다목적 최적화 문제에 대한 다양한 파레토 집합을 보다 효율적으로 구할 수 있는 휴리스틱 방법인 HRCR(Heuristic for Resource Capacity Reduction)을 소개한다. HRCR은 RCPSP를 해결하는 가변 이웃탐색 알고리즘과 자원 평준화를 이루기 위한 자원의 용량 감소에 기반한다. 논문의 구성은 다음과 같다. 제 2장에서는 자원평준화를 위해 사용된 공통된 목적함수를 정의한다. 제 3장에서는 HRCR을 소개하고 알고리즘의 구현 방식에 대해 설명한다. 마지막으로 제 4장에서는 HRCR의 성능평가를 수행한 결과를 설명한다.

2. 자원 평준화 목적 함수

자원 평준화는 고정된 프로젝트 종료일을 제약으로 하여 자원 사용량의 변동을 최소화하는 기본적인 목적을 가진다. 자원 평준화에 대한 연구에서 이러한 기본적인 목적을 다양한 목적 함수로 표현하였다. 종료일이 T 인 어느 스케줄 S 하에서 시각 t 에서의 자원 k 의 사용량이 U_{kt} 일 때 사용량의 최대, 평균, 그리고 분산은 각각 M_k , A_k , V_k 이라고 하자.

$$M_k = \max \{ U_{kt}, t = 1, \dots, T \}$$

$$A_k = \sum_{t=1}^T U_{kt} / T$$

$$V_k = \sum_{t=1}^T (U_{kt} - A_k)^2 / T$$

본 연구에서는 다음과 같은 목적 함수가 고려된다. W_k 는 목적 함수의 정의에서 자원 k 에 대한 가중치를 의미한다.

Resource Investment(Resource Availability Cost)

$$f_1 = \min \sum_{k=1}^m w_k M_k$$

Difference between actual usage and desired usage

$$f_2 = \min \sum_{k=1}^m w_k \sum_{t=1}^T \left| U_{kt} - \sum_{j=1}^n r_{jk} / T \right|$$

Variance

$$f_3 = \min \sum_{k=1}^m w_k V_k$$

Step difference

$$f_4 = \min \sum_{k=1}^m w_k \sum_{t=2}^T |U_{kt} - U_{kt-1}|$$

Resource Improvement Coefficient

$$f_5 = \min \sum_{k=1}^m w_k \left\{ \left(T \sum_{t=1}^T U_{kt}^2 \right) / \left(\sum_{t=1}^T U_{kt} \right)^2 \right\}$$

위 함수들은 자원평준화의 선행연구에서 자주 언급되는 목적 함수이다[2, 12, 14]. 자원 투자비용을 나타내기 위해 개발된 f_1 은 실제로 값 비싼 자원을 구입해야 하는 경우에 사용된다[14]. f_2 는 평균 자원 효율로부터 얻어진 자원 사용량 편차의 절대값을 고려한다. 분산의 가중치 합계인 f_3 은 자원 평준화에서 비정규 목적 함수에 일반적으로 사용된다. f_4 는 연속적인 기간에서 자원 사용량 간의 단계 차이를 고려한다. 이때, 자원이 다른 여러 유형의 인력을 대표하는 경우에 사용되어야 하며 기간별 인력규모는 평활화 되어야 한다. f_5 는 Harris[7]가 개발한 자원 투자 계수(RIC)의 가중치 합계이다. RIC는 직사각형 형태를 가진 이상적 자원 히스토그램과 선택된 자원 히스토그램과의 차이를 측정하는데 사용된다[15].

3. 자원 용량 감소 기반의 휴리스틱

본 연구에서 다루는 다목적 최적화 문제는 자원 소요 평준화와 종료일의 최소화를 달성하기 위한 것이다. 종료일의 최소화는 하나의 목적함수로 표현할 수 있으나 평준화의 경우에는 다양한 목적함수가 존재한다. 모든 평준화 목적 함수를 동시에 고려하여 파레토 집합을 구

하는 것은 매우 어려운 문제이다. 본 연구에서는 모든 평준화 함수를 직접적으로 고려하기 보다는 평준화라는 기본적인 목적을 달성하여 다양한 평준화 목적 함수의 최적화를 간접적으로 이루기 위한 휴리스틱 방법의 개발을 목적으로 한다.

파레토 집합을 구하기 위한 휴리스틱 방법은 자원 용량에 대한 대안들의 효과적인 탐색을 목적으로 한다. 초기에 무한대의 자원 용량을 가정하고, 용량 감소의 대상이 되는 자원의 집합 P 에 모든 자원이 포함되도록 한다. 무한대 용량 제약을 가지는 RCPSP의 최적해는 용량 제약이 없는 경우의 중요 경로(critical path)의 종료일과 동일한 종료일을 갖게 된다. 이 해에서의 각 자원 형태에 대한 최대 자원 사용량을 초기 용량으로 하고, 용량들을 점차적으로 감소시켜 RCPSP의 해를 구한다. 이 같은 자원 용량 감소와 RCPSP 알고리즘의 적용을 반복하여 일련의 해를 구한 후 파레토 집합에 해당하는 해들을 선정한다. RCPSP 문제를 해결하기 위해서는 VNS에 기반한 알고리즘[19, 20]을 사용하였다.

자원의 용량을 감소할 때 어느 자원을 얼마만큼 감소하여야 하는지를 결정하여야 한다. 용량을 감소시켜야 할 자원은 가장 변동이 큰 소요를 나타내는 자원이다. 평준화된 소요를 나타내는 자원에 대한 용량의 감소는 종료일의 큰 증가를 의미한다. 이는 평준화가 이루어진 자원은 현재 스케줄의 병목 역할을 하는 치명적인 자원이기 때문이다. 종료일의 큰 증가 없이 자원의 평준화를 가져올 수 있기 위해서는 가장 변동이 큰 소요를 나타내는 자원의 용량을 우선적으로 감소시키는 것이 타당하다. 변동의 척도로는 각 자원 사용량의 분산을 사용한다.

s 번째 탐색에서 주어진 자원 용량($R_k, k \in K$)으로 구한 RCPSP의 현재 스케줄에서 각 자원 사용량의 최대, 평균, 그리고 분산을 구했다고 하자. 자원의 집합 P 에 속한 자원 중 분산이 가장 큰 자원 k' 을 선택한다.

$$k' = \underset{k}{\operatorname{argmax}} \{ V_k \}, k \in P$$

선정된 자원의 용량 감소는 자원의 최대 사용량과 평균 사용량 간의 차이를 이용한다. $s+1$ 번째 탐색에서 선택된 자원의 용량은 다음 식에 의한다.

$$R_{k'} = M_{k'} - |\alpha_{k'}(M_{k'} - A_{k'})|, 0 < \alpha_{k'} < 1$$

단, $R_{k'}$ 는 최소 용량 $R_{k'}^{\min}$ 보다 크거나 같아야 하고, 이전 단계에서의 $R_{k'}$ 보다 작아야 한다.

파레토 집합을 효과적으로 구하기 위하여 허용되는 최대 종료일을 설정할 수 있다. 최대 종료일을 초과하는 스케줄을 생성하면 선정되었던 자원 용량의 감소량을 적게 하고, 다시 RCPSP의 알고리즘을 실행한다. 만약, 더 이상의 감소가 불가능하면 이 자원을 P 에서 제거하고, 이전의 용량으로 재설정하는 backtracking 작업을 수행한 후 P 에서 사용량 분산이 가장 큰 자원을 선정하여 용량을 감소하는 작업을 반복한다.

탐색은 $P = \emptyset$ 로 모든 자원의 감소가 불가능하게 되면 종료된다.

Heuristic for Resource Capacity Reduction

- | | |
|---------|---|
| 0 | <u>Initialization</u> |
| 0.1 | $P = K$ |
| 0.2 | $S = \emptyset$ |
| 0.3 | $R_k = \infty, k \in K$ |
| 0.4 | apply RCPSP algorithm with resource capacity
$R_k, k \in K$ and compute M_k, A_k, V_k |
| 0.5 | if the schedule is infeasible, then stop |
| 0.6 | $R_k = RP_k = M_k, k \in K$ |
| 0.7 | $VP_k = V_k, k \in K$ |
| 0.8 | $\Delta R_k = M_k - A_k, k \in K$ |
| 0.9 | add current schedule and $R_k, k \in K$ to solution set S |
| 1 | <u>Repetitive resource capacity reduction</u> |
| 1.1 | while ($P \neq \emptyset$) do |
| 1.1.1 | find resource type k' where $V_{k'} = \max\{VP_k\}, k \in P$ |
| 1.1.2 | $R_{k'} = RP_{k'} - \alpha_{k'}, \Delta R_{k'}$ |
| 1.1.3 | if ($R_{k'} = RP_{k'}$) then $R_{k'} = RP_{k'} - 1$ |
| 1.1.4 | if ($R_{k'} < R_{k'}^{\min}$) then $R_{k'} = R_{k'}^{\min}$ |
| 1.1.5 | apply RCPSP algorithm with resource capacity
$R_k, k \in K$ and compute M_k, A_k, V_k |
| 1.1.6 | if the schedule is feasible, then |
| 1.1.6.1 | $VP_k = V_k, k \in K$ |
| 1.1.6.2 | $RP_k = M_k, k \in K$ |
| 1.1.6.3 | $\Delta R_k = M_k - A_k, k \in K$ |
| 1.1.6.4 | add current schedule and $R_k, k \in K$ to solution set S |
| 1.1.7 | if ($R_{k'} = RP_{k'} - 1$) or ($R_{k'} = R_{k'}^{\min}$), then $P = P - k'$,
$R_{k'} = RP_{k'}$ else $\Delta R_{k'} = \alpha_{k'} \Delta R_{k'}$ |
| 2 | <u>Find Pareto optimal from solution set S</u> |

4. 실험 및 분석

다목적 프로젝트 스케줄링 문제의 파레토 집합을 구

하기 위한 휴리스틱 방법의 성능 분석을 위해 표준 문제인 PSPLIB[11]에서 작업 수가 120개인 J120 문제 집합을 대상으로 하였다. HRCR을 포함한 알고리즘들은 자바 프로그래밍 언어로 구현되어 2.6GHz의 Xeon CPU와 32GB RAM의 PC에서 실행하였다. HRCR 방법에서 자원 k 의 감소 가중치인 α_k 값을 모든 자원 형태에 대해 0.5로 설정하였다.

4.1 HRCR 실행 절차

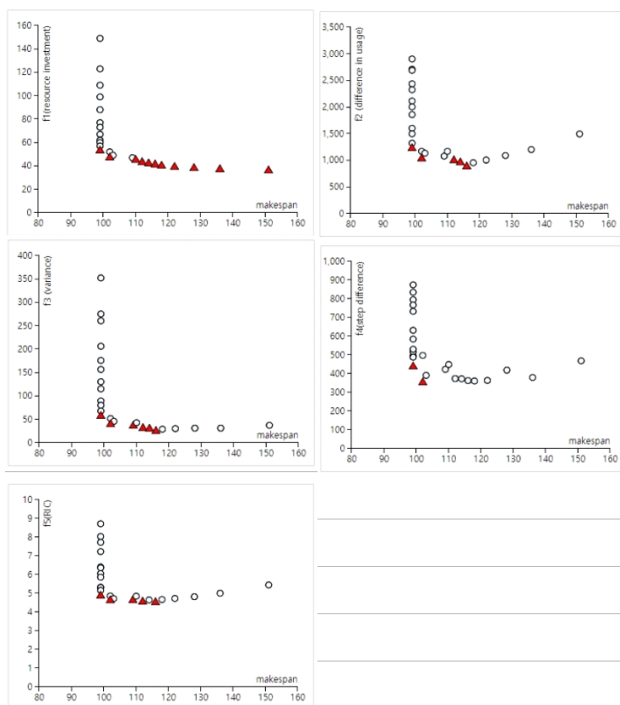
<Table 1>은 자원 형태의 수가 4인 X1_1 문제에 HRCR을 적용하여 순차적으로 구한 25개의 해를 나타낸다. 각 자원 용량 R을 입력으로 VNS에 기반한 알고리즘을 실행하여 RCPSP 문제에 대한 스케줄을 구하였고, 자원 사용량에 대한 최대(M), 평균(A), 분산(V), 종료일(T)을 구하였다. 각 단계에서 집합 P에 속한 자원 중 가장 큰 분산(굵게 나타낸 값)을 가지는 자원을 선택하여 용량을 감소시킨다. 허용되는 최대 종료일을 무한대로 설정하였으므로 종료일에 의해 유효하지 않은 해를 생성하는 경우는 발생하지 않는다. 그러나, 각 자원의 최소 용량이 있어 이 미만으로는 더 이상의 용량 감소가 이루어지지 않는다. 최소용량은 각 활동에서의 자원 요구량 중 가장 큰 값으로 설정하였다.

Step 0에서는 자원의 용량을 무한대로 하여 VNS 알고리즘을 적용하였고, 그 결과 가장 분산이 큰 자원 1의 용량이 $32(= 49 - 0.5 \times (49 - 13.7))$ 로 감소된다. 단계 15에서는 분산이 가장 큰 자원 3을 선택하여야 하지만 이미 최소 용량 10이 되어 P에서 자원 3이 제거하고, 그 다음으로 분산이 큰 자원 1을 선택하였다.

<Figure 1>은 종료일을 x축, f_1 부터 f_5 까지의 자원 평균화 목적 함수를 y축으로 하여 HRCR에 의한 24개 해(초기해 제외)를 나타낸 것으로 삼각형으로 표현된 해는 파레토 해이다. 그림에 따르면 $f_1, (f_2, f_3, f_5)$, 그리고, f_4 들은 서로 다른 패턴을 보인다. f_1 은 용량감소에 따라 종료일이 증가하고, 평균화 값이 감소하는 전형적인 특성을 보여 적지 않은 수의 파레토 해가 생성되었음을 알 수 있다. 반면, f_4 는 파레토 해가 2개로 적은 수이다. 이는 용량의 감소로 종료일은 증가하지만, 평균화 값이 반드시 감소하지 않는다는 사실에 기인한다. f_2, f_3, f_5 역시 이와 유사한 결과를 보인다. 대체로 종료일이 0일 보다 크게 되는 자원 용량에서는 오히려 평균화 값이 증가하는 현상을 보인다. 때문에 f_1 을 제외한 나머지 평균화 목적함수에서는 파레토 해의 수가 축소됨을 의미한다. 모든 평균화 목적함수를 고려한 파레토 해는 f_4 에서의 두 해에 해당한다.

<Table 1> Solutions Generated by HRCR with X1_1 Problem

Step	R	M	A	V	P	T
0	($\infty, \infty, \infty, \infty, \infty$)	(49, 34, 37, 29)	(13.7, 8.5, 7.5, 6.5)	(130.6, 65.5, 102.7, 54.2)	{1, 2, 3, 4}	99
1	(32, 34, 37, 29)	(32, 25, 37, 29)	(13.7, 8.5, 7.5, 6.5)	(81.8, 46.3, 93.2, 54.2)	{1, 2, 3, 4}	99
2	(32, 25, 23, 29)	(32, 25, 23, 29)	(13.7, 8.5, 7.5, 6.5)	(77.2, 64.0, 70.4, 49.1)	{1, 2, 3, 4}	99
3	(23, 25, 23, 29)	(23, 24, 23, 29)	(13.7, 8.5, 7.5, 6.5)	(45.6, 46.3, 63.7, 50.6)	{1, 2, 3, 4}	99
4	(23, 24, 16, 29)	(23, 21, 16, 28)	(13.7, 8.5, 7.5, 6.5)	(43.1, 36.7, 27.4, 49.4)	{1, 2, 3, 4}	99
5	(23, 21, 16, 18)	(23, 21, 15, 18)	(13.7, 8.5, 7.5, 6.5)	(61.2, 49.3, 33.9, 31.5)	{1, 2, 3, 4}	99
6	(19, 21, 15, 18)	(19, 21, 15, 18)	(13.7, 8.5, 7.5, 6.5)	(25.0, 40.6, 31.0, 33.7)	{1, 2, 3, 4}	99
7	(19, 15, 15, 18)	(19, 15, 15, 18)	(13.7, 8.5, 7.5, 6.5)	(26.4, 24.6, 27.3, 37.0)	{1, 2, 3, 4}	99
8	(19, 15, 15, 13)	(19, 15, 15, 13)	(13.7, 8.5, 7.5, 6.5)	(24.8, 20.2, 23.5, 20.8)	{1, 2, 3, 4}	99
9	(17, 15, 15, 13)	(17, 15, 15, 13)	(13.7, 8.5, 7.5, 6.5)	(13.4, 22.5, 22.3, 22.0)	{1, 2, 3, 4}	99
10	(17, 12, 15, 13)	(17, 12, 15, 13)	(13.7, 8.5, 7.5, 6.5)	(11.2, 7.2, 30.3, 19.2)	{1, 2, 3, 4}	99
11	(17, 12, 12, 13)	(17, 12, 12, 13)	(13.7, 8.5, 7.5, 6.5)	(14.6, 11.8, 17.0, 16.0)	{1, 2, 3, 4}	99
12	(17, 12, 10, 13)	(17, 12, 10, 13)	(13.3, 8.3, 7.3, 6.3)	(13.5, 10.8, 11.2, 16.4)	{1, 2, 3, 4}	102
13	(17, 12, 10, 10)	(17, 12, 10, 10)	(13.3, 8.3, 7.3, 6.3)	(13.1, 8.2, 7.2, 6.2)	{1, 2, 3, 4}	103
14	(16, 12, 10, 10)	(16, 12, 10, 10)	(13.3, 8.3, 7.3, 6.3)	(8.5, 13.7, 10.9, 8.8)	{1, 2, 3, 4}	102
15	(16, 11, 10, 10)	(16, 11, 10, 10)	(12.4, 7.7, 6.8, 5.9)	(10.8, 7.3, 12.0, 8.4)	{1, 2, 4}	109
16	(15, 11, 10, 10)	(15, 11, 10, 10)	(12.3, 7.7, 6.8, 5.8)	(9.3, 8.7, 13.2, 11.7)	{1, 2, 4}	110
17	(15, 11, 10, 8)	(15, 11, 10, 8)	(12.1, 7.5, 6.6, 5.7)	(9.6, 7.0, 12.2, 4.7)	{1, 2, 4}	112
18	(14, 11, 10, 8)	(14, 11, 10, 8)	(11.9, 7.4, 6.5, 5.6)	(6.6, 7.6, 12.7, 5.2)	{1, 2, 4}	114
19	(14, 10, 10, 8)	(14, 10, 10, 8)	(11.7, 7.3, 6.4, 5.5)	(5.3, 5.3, 12.3, 4.1)	{1, 2, 4}	116
20	(13, 10, 10, 8)	(13, 10, 10, 8)	(11.5, 7.1, 6.3, 5.4)	(3.8, 7.4, 12.3, 5.5)	{1, 4}	118
21	(13, 10, 10, 7)	(13, 10, 10, 7)	(11.1, 6.9, 6.1, 5.2)	(4.6, 7.3, 13.7, 4.7)	{1}	122
22	(12, 10, 10, 7)	(12, 10, 10, 7)	(10.6, 6.6, 5.8, 5.0)	(4.2, 8.7, 14.4, 3.9)	{1}	128
23	(11, 10, 10, 7)	(11, 10, 10, 7)	(9.9, 6.2, 5.5, 4.7)	(1.7, 9.5, 13.7, 6.4)	{1}	136
24	(10, 10, 10, 7)	(10, 10, 10, 7)	(9.0, 5.6, 5.0, 4.2)	(2.2, 13.1, 15.6, 6.4)	{}	151



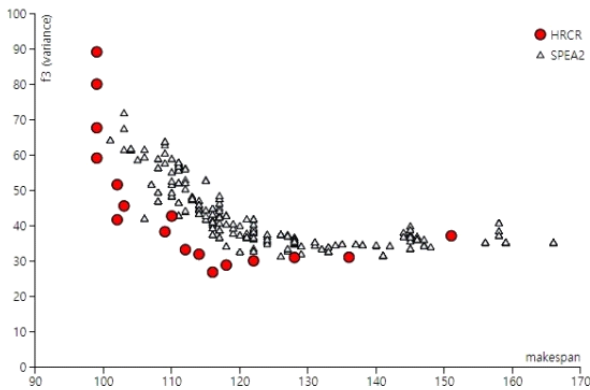
<Figure 1> Bi-Objective Views of Solutions Generated by HRCR

4.2 HRCR과 SPEA2의 성능 비교

<Figure 2>는 X1_1문제에 대해 HRCR과 SPEA2에 의한 해를 나타낸 것이다. SPEA2에서 목적 함수로는 종료 일과 자원 소요량 분산(f_3)을 적용하였다. SPEA2 알고리즘은 Ballestin and Blanco[1]의 방법을 사용하였다. 초기 해를 생성하기 위해 LFT(Latest Finish Time) 규칙을 적용한 후회 기반 편향 랜덤 샘플링[11]을 이용하였고, double justification[18]에 의한 부분 최적화를 적용하였다. 해에 대한 스케줄은 전진 스케줄링에 의해 구한다. 적자선택은 룰렛 휠 방식을 사용하고, 교배연산자로는 두 위치 전진과 두 위치 후진, 돌연변이 연산자로는 리스트에서 임의의 두 작업 위치를 바꾸는 교환 연산을 사용한다. 교배 확률과 돌연변이 확률은 각각 0.75와 0.05로 하였고, 개체수는 400, 세대수는 1,000으로 하였다.

SPEA2에서의 입력 자료인 자원 용량은 HRCR로 생성된 <Table 1>의 25개 해 중 step 0부터 step 10까지를 제외한 나머지 14개 해에서의 자원 용량으로 하였다. 이에 따라 SPEA2를 14번 실행하였다. 한 번의 실행에서 50개의 어카이브 개체를 구하였으므로 총 700개 해를 구하였다.

그림에서 원의 표식은 HRCR에 의한 해를 나타내고, 삼각형 표식은 SPEA2에 의한 700개 해를 나타낸다. SPEA2의 해들이 HRCR의 해에 의해 모두 지배(dominated)되어 HRCR 방법이 상대적으로 우수한 해를 생성하고 있음을 보여준다. 자원 소요의 분산이 아닌 다른 평준화 함수에서도 동일한 결과를 가져왔다. X1_1 문제 외 다른 문제에서도 동일한 결과를 가져와 HRCR의 해가 SPEA2의 해 보다 우수한 해를 생성하였다.



<Figure 2> Solutions Generated by HRCR and SPEA2

4.3 HRCR해와 파레토 최적해 비교

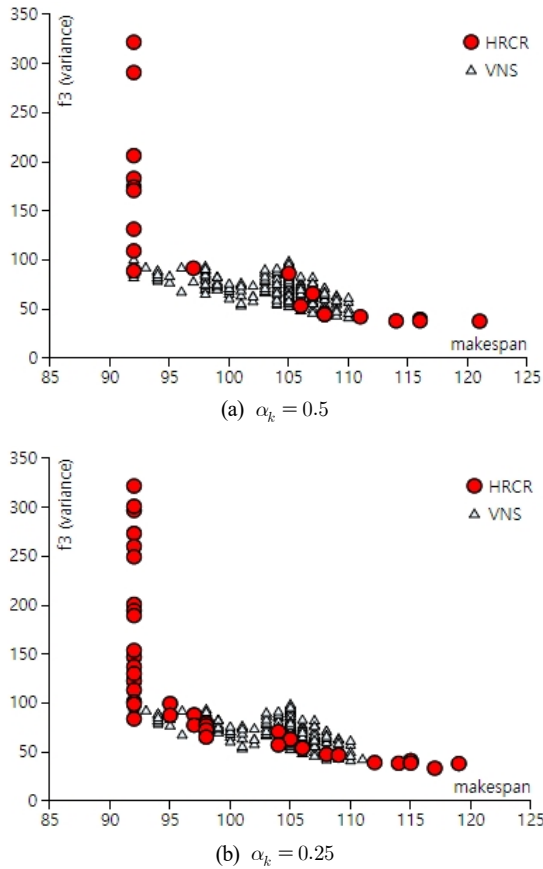
HRCR에 의한 해가 파레토 집합을 대표하는지를 조사하기 위해서는 모든 용량의 조합을 입력으로 구한 RCPSP 해들과 비교하여야 한다. X1_1에서 4개 자원의 용량의 범위는 최소 (10, 10, 10, 7)에서 최대 (16, 16, 16, 13)까지로 설정할 수 있다. 이러한 경우 모두 $7^4 (= 2401)$ 개의 용량 조합이 있게 된다. 실행시간을 단축시키기 위해 이 중 균등한 자원 용량 분포에 따르는 256개 조합을 선정하였다.

<Table 2>는 30개 문제에 대해 HRCR에 의해 구해진 해의 수(no sols)와 각 평준화 함수에서 HRCR에 의한 해가 256개 용량 조합으로 구한 해를 지배하는 비율을 나타낸다. 100%에 가까운 지배비율을 보인다면 HRCR에 의한 해가 파레토 집합을 효과적으로 생성한다는 의미를 갖는다. 평균적으로는 종료일과 자원 평준화 함수의 두 목적함수에서 256개 해 중 HRCR 해에 의해 지배되는 비율이 89% 부터 94% 사이에 있다. 지배비율이 가장 작은 문제는 X5_1으로 f_3 의 경우 40%에 불과하다. 작은 지배비율은 문제의 복잡성에 비해 HRCR에 의한 해의 수가 적은 것에 기인한다. 보다 많은 해를 생성한다면 지배비율이 증가될 수 있다. 이를 위해서는 자원 용량 감소 가중치인 α_k 를 0.5 대신에 보다 적은 수로 설정할 수 있다. 즉, 용량의 감소량을 적게 하여 보다 많은 자원 용량 조합을 생성하는 것이다.

<Table 2> Ratio of 256 VNS Solutions Dominated by HRCR Solutions

problem	no sols	f1	f2	f3	f4	f5
X1_1	25	0.91	0.91	0.90	0.91	0.93
X2_1	30	0.79	0.91	0.79	1.00	0.99
X3_1	29	0.91	0.93	0.92	0.94	0.93
X4_1	26	0.85	0.89	0.87	0.90	0.92
X5_1	26	0.41	0.68	0.40	0.46	0.72
X6_1	21	0.93	0.92	0.93	0.94	0.93
X7_1	49	0.97	0.99	0.97	0.99	0.99
X8_1	43	0.94	0.94	0.93	0.97	0.93
X9_1	43	0.97	0.98	0.98	0.99	0.99
X10_1	36	0.96	0.99	0.96	0.97	0.99
X11_1	52	0.98	0.98	0.97	0.98	0.99
X12_1	50	0.98	0.97	0.97	1.00	0.99
X13_1	45	0.98	0.96	0.96	0.98	0.97
X14_1	61	0.98	0.97	0.97	0.99	0.98
X15_1	58	0.97	0.99	0.99	1.00	0.99
X16_1	74	0.98	0.99	0.98	1.00	0.99
X17_1	72	0.99	1.00	0.99	0.99	1.00
X18_1	67	0.98	0.99	0.98	0.99	1.00
X19_1	61	0.91	0.90	0.71	0.99	0.92
X20_1	62	0.96	0.99	0.98	0.98	1.00
X21_1	19	0.64	0.64	0.58	0.49	0.76
X22_1	20	0.92	0.87	0.85	0.91	0.90
X23_1	23	0.86	0.91	0.83	0.84	0.87
X24_1	22	0.80	0.84	0.82	0.82	0.82
X25_1	22	0.87	0.84	0.81	0.81	0.89
X26_1	36	0.95	0.98	0.95	0.96	0.98
X27_1	43	0.95	0.96	0.94	0.96	0.98
X28_1	40	0.95	0.94	0.95	0.93	0.96
X29_1	41	0.98	0.98	0.98	0.98	0.98
X30_1	33	0.85	0.89	0.84	0.88	0.89
ave		0.90	0.93	0.89	0.92	0.94
std		0.12	0.08	0.13	0.13	0.07

<Figure 3(a)>는 HRCR에서 생성된 21개의 해와 256개의 RCPSP 해를 (작업소요기간, f_3)의 다목적적으로 나타낸 것으로 매우 적은 숫자의 HRCR 해가 낮은 지배비율의 원인으로 분석된다. 보다 많은 해를 생성한다면 지배비율을 높일 수 있는 기회가 부여된다. HRCR에서 더 많은 해를 생성하기 위해서는 용량감소의 가중치인 α_k 값을 줄일 수 있다. α_k 의 값이 작을수록 더 많은 자원용량 조합을 생성할 수 있다. <Figure 3(b)>는 α_k 값이 0.5 대신에 0.25로 설정되었을 때, 41개의 HRCR이 생성된다는 것을 나타낸 것이다. 이때의 지배비율은 87%로 증가한다. 본 결과는 HRCR에서 용량감소를 위한 가중치를 적절히 조정하여 파레토 집합의 대표성을 높일 수 있음을 의미한다.



<Figure 3> HRCR Solutions Generated by Varying Weight for Resource Reduction

5. 결론

본 연구에서는 프로젝트 일정의 다목적 최적화 문제에 대한 파레토 집합을 효율적으로 도출하기 위한 경험적 알고리즘 HRCR을 소개하였다. 이 알고리즘은 RCPSW와 자원평준화의 다목적 문제를 해결하기 위해 개발되었다. RCPSW의 해를 구하기 위해 VNS를 사용하였고, 자원평준화 문제는 직접적으로 해결하는 대신에 자원 용량 감소 절차를 적용하여 평준화 수준을 개선하도록 하였다. 프로젝트 일정관리 분야에서 개발된 기존의 다목적 최적화 알고리즘은 자원 용량을 고정된 값으로 간주하는 반면에 HRCR에서는 자원 용량이 변수로 간주된다. HRCR의 성능을 분석하기 위해 표준 문제인 PSPLIB를 이용한 실험을 수행하였다. 실험 결과 HRCR은 작업소요시간에 대한 다양한 옵션과 자원 평준화의 정도를 통해 대안 일정을 효율적으로 생성한다는 것을 입증하였다. 또한 생성된 해들은 효과적으로 파레토 집합을 대표한다. HRCR의 용량 절감을 위한 가중치를 변경하면 생성된 해의 수를 조절할 수 있다. 가중치를 적게 하여 HRCR 해의 수를 늘리면 파레토 집합의 대표성이 향상될 수 있다.

References

- [1] Ballestin, F. and Blanco, R., Theoretical and practical fundamentals for multi-objective optimization in resource-constrained project scheduling problems, *Computers and Operations Research*, 2011, Vol. 38, No. 1, pp. 51-62.
- [2] Brucker, P., Drexl, A., Mohring, R., Neumann, K., and Pesch, E., Resource-constrained project scheduling : Notation, classification, models, and methods, *European Journal of Operations Research*, 1999, Vol. 112, No. 1, pp. 3-41.
- [3] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., A fast and elitist multi objective genetic algorithm : NSGA-II, *IEEE Transactions on Evolutionary Computing*, 2002, Vol. 6, No. 2, pp. 182-197.
- [4] Geiger, M.J., Foundation of the pareto iterated local search metaheuristic, in *Proceedings of the 18th MCDM in Chania*, Greece, 2006.
- [5] Geiger, M.J., Randomized variable neighborhood search for multi-objective optimization, in *4th EU/ME : Design and Evaluation of Advanced Hybrid Meta-Heuristics*, 2008, pp. 34-42.
- [6] Gomes, H.C., Neves, F.A., and Souza, M.J.F., Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations, *Computers and Operations Research*, 2014, Vol. 44, pp. 92-104.
- [7] Harris, R., Precedence and arrow networking techniques for construction, Wiley, NY., 1978.
- [8] Hartmann, S. and Kolisch, R., Experimental evaluation of state-of-the art heuristics for the resource-constrained project scheduling problem, *European Journal of Operational Research*, 2000, Vol. 127, No. 2, pp. 394-407.
- [9] Hu, J. and Flood, I., A multi-objective scheduling model for solving the resource-constrained project scheduling and resource leveling problem, *ASCE International Conference on Computing in Civil Engineering*, 2012, pp. 49-56.
- [10] Kolisch, R. and Hartmann, S., Experimental investigation of heuristics for resource-constrained project scheduling : An update, *European Journal of Operational Research*, 2006, Vol. 174, No. 1, pp. 23-37.
- [11] Kolisch, R. and Sprecher, A., PSPLIB-A project scheduling library : OR software-ORSEP operations research software exchange program, *European Journal of Operational Research*, 1996, Vol. 96, No. 1, pp. 205-216.

- [12] Kyriklidis, C. and Dounias, G., Evolutionary computation for resource leveling optimization in project management, *Integrated Computer Aided Engineering*, 2016, Vol. 23, No. 2, pp. 173-184.
- [13] Nebro, A.J., Durillo, J.J., Garcia-Nieto, J., Coello, C.A., Luna, F., and Alba, E., SMPSO : A new PSO-based metaheuristic for multi-objective optimization, in 2009 *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision Making*, 2009, pp. 66-73.
- [14] Neumann, K. and Zimmermann, J., Resource leveling for projects with schedule-dependent time windows, *European Journal of Operational Research*, 1999, Vol. 117, No. 3, pp. 591-605.
- [15] Ponz-Tienda, J.L., Yepes, V., and Pellicer, E., Moreno-Flores, J., The resource leveling problem with multiple resource using an adaptive genetic algorithm, *Automation in Construction*, 2013, Vol. 29, pp. 161-172.
- [16] Reynolds, A.P. and De la Iglesia, B., A multi-objective GRASP for partial classification, *Soft Computing*, 2009, Vol. 13, pp. 227-243.
- [17] Valls, V., Ballestin, F., and Quintanilla, S., A population-based approach to the resource-constrained project scheduling problem, *Annals of Operations Research*, 2004, Vol. 131, pp. 305-324.
- [18] Valls, V., Ballestin, F., and Quintanilla, S., Justification and RCPSP : a technique that pays, *European Journal of Operational Research*, 2005, Vol. 165, No. 2, pp. 375-386.
- [19] Yim, D.-S., Dynamic decision using variable neighborhood search for stochastic resource-constrained project scheduling problem, *Journal of the Korean Institute of Industrial Engineers*, 2017, Vol. 43, No. 1, pp. 1-11.
- [20] Yim, D.-S., Performance analysis of local optimization algorithms in resource-constrained project scheduling problem, *Journal of the Korean Institute of Industrial Engineers*, 2011, Vol. 37, No. 4, pp. 408-414.
- [21] Zitzler, E., Laumanns, M., and Thiele, L., SPEA2 : improving the strength pareto evolutionary algorithm, Swiss Federal Institute of Technology, Zurich, Switzerland, 2001.

ORCID

Woo-Jin Jeong | <http://orcid.org/0000-0002-3330-1729>

Sung-Chul Park | <http://orcid.org/0000-0002-4618-1244>

Dong-Soon Yim | <http://orcid.org/0000-0003-4968-9091>