

<https://doi.org/10.7236/JIIBC.2023.23.5.151>
JIIBC 2023-5-23

일반화된 배정 문제의 k-opt 교환 최적화 알고리즘

Optimization Algorithm for k-opt Swap of Generalized Assignment Problem

이상운*

Sang-Un Lee*

요약 NP-난제로 다항시간으로 최적 해를 찾는 알고리즘이 제안되지 않고 있는 일반화된 배정 문제에 대해 기존에는 전적으로 메타휴리스틱 기법들에 치중하여 연구가 진행되었다. 반면에, 본 논문에서는 해를 찾아가는 규칙을 가진 휴리스틱 탐욕 알고리즘을 제안한다. 첫 번째로, m 대의 기계(용기)에 n 개의 작업(물품)을 담을 수 있도록 $l = n/m$ 개가 되도록 각 기계의 용량 b_i 에 대해 가중치 $w_{ij} \leq b_i/l$ 데이터로 축소시킨다. 축소된 데이터들을 대상으로 각 작업의 최대 이득 작업을 해당 기계에 배정하였다. 두 번째로, 각 기계에 배정된 가중치 합이 기계 용량을 초과하지 않도록 배정을 조정하였다. 마지막으로 이득을 최대화시키기 위해 k-opt 교환 최적화를 수행하였다. 제안된 알고리즘을 50개 벤치마킹 데이터들에 적용한 결과 약 1/3 데이터에 대해서는 알려진 최적 해를 찾을 수 있었으며, 나머지 2/3 데이터에 대해서는 메타휴리스틱 기법들과 견줄만한 결과를 보였다. 따라서 제안된 알고리즘은 GAP에 대해 다항시간으로 해를 찾아가는 규칙이 존재할 가능성을 보여 NP-난제에서 P-문제로 될 수 있음을 실험을 통해 증명하였다.

Abstract The researchers entirely focused on meta-heuristic method for generalized assignment problem(GAP) that is known as NP-hard problem because of the optimal solution within polynomial time algorithm is unknown yet. On the other hand, this paper proposes a heuristic greedy algorithm with rules for finding solutions. Firstly, this paper reduces the weight matrix of original data to $w_{ij} \leq b_i/l$ in order to n jobs(items) pack m machines(bins) with $l = n/m$. The maximum profit of each job was assigned to the machine for the reduced data. Secondly, the allocation was adjusted so that the sum of the weights assigned to each machine did not exceed the machine capacity. Finally, the k-opt swap optimization was performed to maximize the profit. The proposed algorithm is applied to 50 benchmarking data, and the best known solution for about 1/3 data is to solve the problem. The remaining 2/3 data showed comparable results to metaheuristic techniques. Therefore, the proposed algorithm shows the possibility that rules for finding solutions in polynomial time exist for GAP. Experiments demonstrate that it can be a P-problem from an NP-hard.

Key Words : Generalized assignment problem, Reduced data, Bin packing, Profit optimization, k-opt

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 2023년 2월 1일, 수정완료 2023년 9월 3일
게재확정일자 2023년 10월 6일

Received: 1 February, 2023 / Revised: 3 September, 2023 /
Accepted: 6 October, 2023

*Corresponding Author: sulee@gwnu.ac.kr

Dept. of Multimedia Eng., Gangneung-Wonju National
University, Korea

I. 서 론

용량(capacity)에 서로 다른 m 개 상자(기계)에 각 상자에 저장될 때 크기(높이)와 이득(또는 비용)이 서로 다른 n 개의 물품(작업)을 이득 최대화(또는 비용 최소화)를 달성하면서 m 개 상자 모두에 채워 넣는 문제를 일반화된 배정 문제(generalized assignment problem, GAP)이라 한다.^[1]

이 문제는 배정(할당) 문제(assignment problem, AP)와 상자 채우기 문제(bin packing problem, BPP)가 결합된 하이브리드 문제로 정의될 수 있다. 일반적으로 AP와 BPP 자체도 다항시간으로 정확한 해를 찾는 규칙을 발견하지 못해 NP-완전(non-deterministic polynomial time-complete, NP-complete)로 분류된 난제이다. 하물며, GAP=AP+BPP로 이 역시 NP-난제(NP-hard)로 분류된 난제 중의 난제라고 할 수 있다.^[2-4]

따라서 GAP에 대해, 아직까지는 시행착오법의 컴퓨터 프로그램을 적용한 메타휴리스틱 기법이나 수학적 모델 연구가 주류를 이루고 있으며, 다항시간의 근사 알고리즘은 제안되지 않고 있다.^[5]

Lee^[6]은 GAP에 대해 $m \times n$ 시간과 비용행렬에 대해 시간행렬 최적화 결정(선택)과 비용행렬 최적화 결정(교환)의 2단계 최적화(TSO) 기법과 $m \times n$ 의 시간x비용행렬에 대해 결정하는 1단계 최적화(OSO) 기법의 $O(mn)$ 복잡도를 가진 시간-비용 최적화 알고리즘을 제안한 연구 결과가 있다.

본 논문에서는 Lee^[6]와는 차별화된 방법으로 GAP의 근사 해를 다항시간으로 찾아갈 수 있는 규칙을 가진 휴리스틱 탐욕 알고리즘을 제안한다. 2장에서는 GAP의 난제를 문제 정의와 더불어 예제 문제를 대상으로 고찰해 본다. 3장에서는 다항시간으로 근사 해를 찾아가는 휴리스틱 탐욕 알고리즘으로 데이터를 축소시켜 배정하고, 모든 상자의 용량을 충족시킨 후 이득을 극대화 시키는 최적화를 수행하는 기법을 제안한다. 4장에서는 GAP 벤치마킹 데이터들에 대해 제안된 알고리즘을 적용하여 다항시간으로 근사 해를 찾을 수 있는 휴리스틱 탐욕 알고리즘의 존재 가능성을 검증해 본다.

II. 일반화된 배정 문제의 난제

GAP은 기계(상자) $M, i=1,2,\dots,m$, 작업(물품) $J, j=1,2,\dots,n$ 에 대해 $m \times n$ 의 이득 행렬 P 와 가중치 행

렬 W 와 용량 $m \times 1$ 개의 용량 행렬 B 에 대해 다음과 같이 p_{ij} 또는 c_{ij} , w_{ij} 와 b_i 용어들이 사용된다.

- p_{ij} : 물품 j 를 상자 i 에 배정 시 획득 이득(profit)
- c_{ij} : 물품 j 를 상자 i 에 배정 시 소요되는 비용(cost)
- w_{ij} : 물품 j 를 상자 i 에 배정 시 가중치(높이, height)
- b_i : 상자 i 의 용량(높이, capacity)

이러한 용어들을 사용하여 GAP은 식 (1) 또는 (2)의 상자문제로 설명될 수 있다.^[1,7,8] n 개 물품을 m 개 상자에 모두 채워 넣는 경우, 각 물품은 정확히 하나의 상자에 놓여야 하며, 어떤 상자도 적재된 물품들의 높이가 상자 높이를 초과할 수 없다. 이러한 조건을 만족하면서 총 이득 최대화(총 비용 최소화)시켜야만 한다.

$$\text{Maximize Profit : } z^+ = \sum_{i=1}^m \sum_{j=1}^n p_{ij} x_{ij} \quad (1)$$

$$\text{Minimize Cost } z^- = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2)$$

$$\text{subject to } \sum_{j=1}^n w_{ij} x_{ij} \leq b_i, i \in M = \{1, 2, \dots, m\}$$

$$\sum_{i=1}^m x_{ij} = 1, j \in N = \{1, 2, \dots, n\}$$

$$x_{ij} = \{0, 1\}, i \in M, j \in N$$

where

$$x_{ij} = \begin{cases} 1 & \text{if item } j \text{ is assigned to knapsack } i; \\ 0 & \text{otherwise} \end{cases}$$

Andreas와 Kurt^[9]에서 인용된 그림 1의 GAP-1 사례를 통해 GAP의 문제 정의와 더불어 해를 얻는데 있어 난제를 이해하여 보자.

이 문제는 $n=3, m=8$ 로 이득을 극대화 시키도록 8개 물품을 3개의 상자에 넣는 문제이다. GAP은 p_{ij} 에 대한 이득(또는 비용)의 배정(할당) 문제(assignment problem, AP)와 더불어 w_{ij}, b_i 에 대한 상자 채우기 문제(bin packing problem, BPP)가 결합된 하이브리드 형태의 문제로 정의될 수 있다. 즉, GAP=AP+BPP의 2개 문제에 대한 조건을 모두 충족시켜야만 하는 난제이다.

쉽게 말하면, n 개의 각 물품은 m 개의 상자에 넣을 경우 이득이 서로 다르며, 높이가 서로 다른 m 개의 상자에 n 개의 각 물품이 m 개의 상자에 들어갈 경우 높이가 서로 다른 경우로 이득 극대화 측면의 AP나 n 개의 물품을 m 개 상자에 모두 넣기 위한 BPP의 어느 한 측면을 충족시켜서는 최적 해를 찾지 못하며, 반드시 양쪽 모두를 충족시켜야만 한다. 일반적으로 AP나 BPP도 다항시간으로 해를 찾아가는 규칙을 아직까지 발견하지 못한 난제

로 NP-완전(complete) 문제로 분류되고 있다. 하물며, NP-완전의 2 문제를 결합한 GAP은 더더욱 풀기 어려운 NP-난제(hard)로 분류된 난제 중의 난제라고 할 수 있다. [2-4]

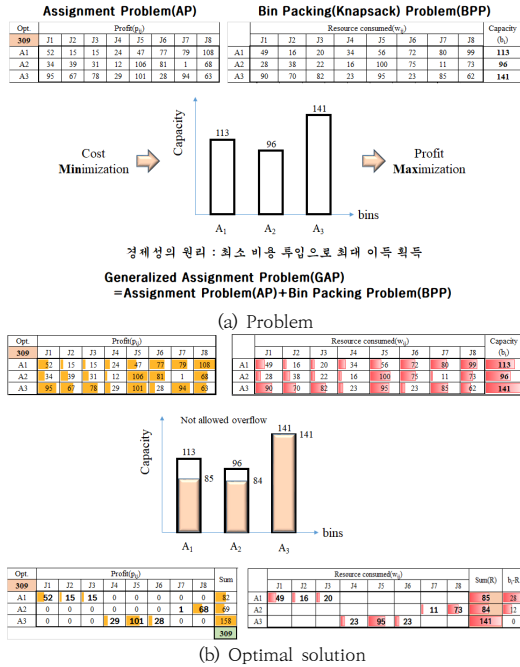


그림 1. GAP-1 예제
 Fig. 1. Example of GAP-1

GAP의 해를 구하기 위해 다양한 메타휴리스틱 기법이나 수학적 모델 등을 적용하고 있지만 정작 다항시간으로 해를 찾아가는 규칙을 가진 휴리스틱 탐색 알고리즘은 제안되지 않고 있는 실정이다. 따라서 3장에서는 GAP의 근사 해를 다항시간으로 찾을 수 있는 탐색 규칙을 제시한다.

III. k-opt 교환 최적화 알고리즘

GAP은 n개의 물품(items)을 m개 상자(bin)에 넣어야 하는 문제로, 동일한 크기의 상자과 물품이라면 각 상자에 n/m개씩 할당할 수 있다. 따라서 기본적으로는 $w_{ij} \leq b_i / \lfloor n/m \rfloor$ 데이터들만을 선택 대상으로 축소시킬 수 있다. 이는 m개 상자 크기가 동일하며, w_{ij} 가 모두 $b_i / \lfloor n/m \rfloor$ 크기인 경우에 한하며, w_{ij} 가 $w_{ij} \ll b_i / \lfloor n/m \rfloor$ 에 치우쳐 있는 경우 $w_{ij} > b_i / \lfloor n/m \rfloor$ 도 추가될 수 있다.

이러한 w_{ij} 의 분포 경우를 고려하여 $l = \lfloor n/m \rfloor - 1$ 로 설정하여 $w_{ij} < b_i/l$ 데이터들만으로 축소된 초기 배정 대상을 추출한다.

축소된 데이터들에 대해 각 작업(물품, 열)에서 $\max p_j$ 를 선정하여 $w_{ij}x_{ij} (x_{ij} = 1)$ 로 각 상자(기계, 행)에 채워진(fill) 물품의 총 크기 $f_i = \sum_{j=1}^n w_{ij}$ 를 계산하여 여유량(clearance or remaining, r) $r_i = b_i - f_i$ 을 계산한다. 만약 상자 밖으로 흘러넘친(overflow) 상자가 존재하면 여유가 있는 상자와 물품 교환으로 모든 상자의 용량을 충족시킨다. 이 과정에서 z^+ 또는 z^- 의 최적 해를 얻기 위해 그림 2의 k-opt 교환 최적화를 식 (3)으로 수행한다. 여기서 출발(s)과 최종 도착(t)의 물품(작업 j)은 $(s_1, t_n) = \max(\max p_j - p_s)$ 인 작업으로 결정한다.

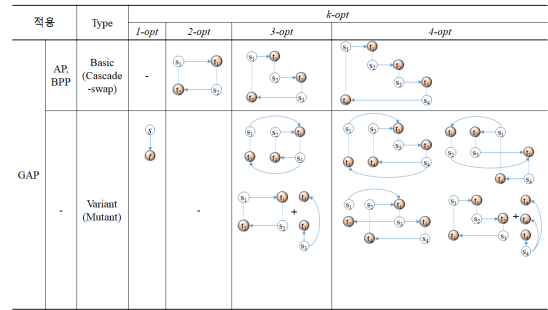


그림 2. k-opt 교환 최적화
 Fig. 2. k-opt swap optimization

$$z^+ : \text{if } (\Sigma p_t - \Sigma p_s) > 0, \forall_i, (b_i - f_i) \geq 0 \text{ then} \quad (3)$$

$$w_s, x_{ij} = 1 \rightarrow 0, \quad w_t, x_{ij} = 0 \rightarrow 1.$$

$$z^- : \text{if } (\Sigma c_s - \Sigma c_t) > 0, \forall_i, (b_i - f_i) \geq 0 \text{ then}$$

$$w_s, x_{ij} = 1 \rightarrow 0, \quad w_t, x_{ij} = 0 \rightarrow 1.$$

AP, BPP를 포함한 대부분의 최적화 문제는 하나의 $m \times n$ 행렬 데이터를 다루기 때문에 기본형(basic type)인 k-opt($k \geq 2$)를 적용한다. 반면에 GAP은 $m \times n$ 행렬의 P와 W의 2개를 결합하여 다루는 복잡한 문제로 기본형 뿐 아니라 변종(variant or mutant) k-opt도 함께 적용해야만 하는 어려움이 있다.

제안된 알고리즘은 GAP의 최적 해인 z^+ 또는 z^- 모두에 적용되는 방법으로 k-opt 교환 최적화 알고리즘(k-opt swap optimization algorithm, k-opt SOA)라 하며 다음과 같이 수행된다.

[z^+ k-opt SOA]

Step 1. 초기 배정

- $l = \lfloor \frac{n}{m} \rfloor - 1, (l \geq 2), \begin{cases} w_{ij} < b_i/l, p_{ij} = p_{ij} \\ w_{ij} \geq b_i/l, p_{ij} = 0 \end{cases}$
- if $|p_j > 0| = 0$ then $\max p_j$ 복원(z^+)
- 각 작업(j 열)에서 $\max p_j$ 선택, $w_{ij} = w_{ij}(x_{ij} = 1)$
- if $|\max p_j| \geq 2$ then $\min w_{ij}$ 선택
- $r_i = b_i - f_i$ 계산

Step 2. 상자 용기 용량 충족과 이득 최적화 수행

- 상자 용량 충족 : $r_i < 0$ 기체가 존재하면, $r_i > 0$ 기체와 w_{ij} 를 상호 교환하는 2-opt 균형 최적화 수행, 모든 기체의 $\sum_{j=1}^n w_{ij}x_{ij} \leq b_i$ 를 충족
- 최적화 : if $(\Sigma p_t - \Sigma p_s) > 0, \forall_i, (b_i - f_i) \geq 0$ then $w_s, x_{ij} = 1 \rightarrow 0, w_t, x_{ij} = 0 \rightarrow 1$.
- 출발점 : $s_1 = \min p, (s_1, t_n) = \max(\max p_j - p_s)$

[z^- k-opt SOA]

Step 1. 초기 배정

- $l = \lfloor \frac{n}{m} \rfloor - 1, (l \geq 2), \begin{cases} w_{ij} < b_i/l, c_{ij} = c_{ij} \\ w_{ij} \geq b_i/l, c_{ij} = \infty \end{cases}$
- if $|c_j = \infty| = m$ then $\min c_j$ 복원(z^-)
- 각 작업(j 열)에서 $\min c_j$ 선택, $w_{ij} = w_{ij}(x_{ij} = 1)$
- if $|\min c_j| \geq 2$ then $\min w_{ij}$ 선택
- $r_i = b_i - f_i$ 계산

Step 2. 상자 용기 용량 충족과 비용 최적화 수행

- 상자 용량 충족 : $r_i < 0$ 기체가 존재하면, $r_i > 0$ 기체와 w_{ij} 를 상호 교환하는 2-opt 균형 최적화 수행, 모든 기체의 $\sum_{j=1}^n w_{ij}x_{ij} \leq b_i$ 를 충족
- 최적화 : if $(\Sigma c_s - \Sigma c_t) > 0, \forall_i, (b_i - f_i) \geq 0$ then $w_s, x_{ij} = 1 \rightarrow 0, w_t, x_{ij} = 0 \rightarrow 1$.
- 출발점 : $s_1 = \max c, (s_1, t_n) = \max(\max c_j - c_s)$

그림 1의 GAP-1 사례에 대해 제안된 z^+ k-opt SOA를 적용한 결과는 그림 3과 같다.

GAP-1에 대해 $l = \lfloor n/m \rfloor - 1, (l \geq 2)$ 을 계산하면 $\lfloor 8/3 \rfloor = 2, 2 - 1 = 1, l \geq 2$ 로 $l = 2$ 를 얻는다. 따라서 $w_{ij} > b_i/2$ 인 데이터가 삭제된 상태에서 각 작업(j 열)의 $\max p_{ij}/w_{ij}$ 를 선택하여 초기 배정한 결과 A_3 상자 용기 용량을 초과(-37)되었음을 알 수 있다. 따라서 A_3 상자 용기의 초과 용량(-37)을 여유가 있는 $A_1(8)$ 과 $A_2(63)$

중 $w_{32} = 70$ 을 $w_{22} = 38$ 로 이동시키는 1-opt를 수행하여 용량을 충족시켰다. 이 때 얻은 총 이득은 290으로, 보다 많은 이득을 얻기 위해 이득 p_{ij} 를 극대화시키는 변형 4-opt 최적화를 수행한 결과 $z^+ = 309$ 를 얻었다.

The figure displays four stages of the k-opt SOA algorithm for the GAP-1 example problem. Each stage consists of a Profit table, a Resource consumed table, and a Sum value.

- Stage 1:** Profit table shows values for jobs 21, 22, 23, 24, 25, 26, 27, 28. Resource consumed table shows values for machines A1, A2, A3. Sum(p) = 165.8.
- Stage 2:** Profit table shows values for jobs 21, 22, 23, 24, 25, 26, 27, 28. Resource consumed table shows values for machines A1, A2, A3. Sum(p) = 178.37.
- Stage 3:** Profit table shows values for jobs 21, 22, 23, 24, 25, 26, 27, 28. Resource consumed table shows values for machines A1, A2, A3. Sum(p) = 198.1.
- Stage 4:** Profit table shows values for jobs 21, 22, 23, 24, 25, 26, 27, 28. Resource consumed table shows values for machines A1, A2, A3. Sum(p) = 309.0.

그림 3. GAP-1 문제에 대한 k-opt SOA
Fig. 3. k-opt SOA for GAP-1 example problem

IV. 적용 및 결과 분석

본 장에서 실험에 사용된 데이터는 Beasley^[9]의 벤치마킹 데이터에서 인용된 gap1 ~ gap10 데이터들로 작업 수는 15개부터 48개까지이다.

본 논문에서 k-opt SOA를 얻은 결과에 대해 기존에 알려진 메타휴리스틱 기법들과 비교하여 알고리즘 성능 비교를 한 결과는 표 1에 제시하였다. 또한 최적 해를 찾은 18개 데이터에 대해 z^+ k-opt SOA로 얻은 해는 그림 4에 제시되어 있다. 참고로, 표 1에서 알려진 최적 해를 100% 찾은 GA^[1]는 각 데이터에 대해 10회 수행 결과 얻은 최대 이득 값을 기록한 것이다.

50개의 GAP 벤치마킹 데이터들 중에서 18개 데이터에 대해서는 기존에 알려진 최적 해를 찾는데 성공하였다. 나머지 32개 데이터에 대해서는 기존에 메타휴리스틱 기법들로 찾은 최적 해에 편차를 가진 근사한 결과를 얻을 수 있었다. 결론적으로 메타휴리스틱 기법의 MMAS와 GRASP에 비해서는 보다 좋은 결과를 나타내어 10개 알고리즘 중 8위로 메타휴리스틱 기법들과 경쟁할 수 있는 수준의 결과를 보였다.

V. 결론 및 추후 연구과제

본 논문은 NP-난제로 분류된 GAP에 대해 다항시간으로 근사 해를 찾아가는 규칙인 휴리스틱 탐욕 알고리즘을 제안하였다.

표 1. 실험 데이터의 해
 Table 1. Solution for experimental data

Benchmarking data		Profit Maximization(z^+)											휴리스틱 k-opt SOA	
		Best known ^[37]	메타휴리스틱, 수학적 모델											
			GA ^[11]	ABF+AS ^[6]	MMAS ^[25]	GRASP ^[25]	ASH+TS ^[25]	GRAH+TS ^[25]	ASH+LS+TS ^[25]	GRAH+LS+TS ^[25]	ASH+LS+CTS ^[25]	SAS ^[36]		
gap1 (5×15)	c515-1	336	336	336	-	-	-	-	-	-	-	-	-	336
	c515-2	327	327	327	-	-	-	-	-	-	-	-	-	327
	c535-3	339	339	339	-	-	-	-	-	-	-	-	-	339
	c515-4	341	341	341	-	-	-	-	-	-	-	-	-	341
c515-5	326	326	326	-	-	-	-	-	-	-	-	-	326	
gap2 (5×20)	c520-1	434	434	434	-	-	-	-	-	-	-	-	-	434
	c520-2	436	436	436	-	-	-	-	-	-	-	-	-	436
	c520-3	420	420	418(-2)	-	-	-	-	-	-	-	-	-	420
	c520-4	419	419	416(-3)	-	-	-	-	-	-	-	-	-	419
	c520-5	428	428	428	-	-	-	-	-	-	-	-	-	428
gap3 (5×25)	c525-1	580	580	574(-6)	-	-	-	-	-	-	-	-	-	578(-2)
	c525-2	564	564	558(-6)	-	-	-	-	-	-	-	-	-	563(-1)
	c525-3	573	573	572(-1)	-	-	-	-	-	-	-	-	-	571(-2)
	c525-4	570	570	570	-	-	-	-	-	-	-	-	-	570
	c525-5	564	564	562(-2)	-	-	-	-	-	-	-	-	-	564
gap4 (5×30)	c530-1	656	656	656	-	-	-	-	-	-	-	-	-	656
	c530-2	644	644	644	-	-	-	-	-	-	-	-	-	640(-4)
	c530-3	673	673	673	-	-	-	-	-	-	-	-	-	669(-4)
	c530-4	647	647	647	-	-	-	-	-	-	-	-	-	647
	c530-5	664	664	661(-3)	-	-	-	-	-	-	-	-	-	660(-4)
gap5 (8×24)	c824-1	563	563	563	-	-	-	-	-	-	-	563	-	563
	c824-2	558	558	558	-	-	-	-	-	-	-	-	-	558
	c824-3	564	564	564	-	-	-	-	-	-	-	-	-	564
	c824-4	568	568	567(-1)	-	-	-	-	-	-	-	-	-	568
	c824-5	559	559	558(-1)	-	-	-	-	-	-	-	-	-	558(-1)
gap6 (8×32)	c832-1	761	761	761	-	-	-	-	-	-	-	-	-	759(-2)
	c832-2	759	759	758(-1)	-	-	-	-	-	-	-	-	-	754(-5)
	c832-3	758	758	758	-	-	-	-	-	-	-	-	-	749(-9)
	c832-4	752	752	752	-	-	-	-	-	-	-	-	-	746(-6)
	c832-5	747	747	747	-	-	-	-	-	-	-	-	-	741(-6)
gap7 (8×40)	c840-1	942	942	942	941(-1)	941(-1)	942	942	942	942	942	-	-	935(-7)
	c840-2	949	949	949	949	948(-1)	949	949	949	949	949	-	-	941(-8)
	c840-3	968	968	968	968	968	968	968	968	968	968	-	-	963(-5)
	c840-4	945	945	945	945	943(-2)	945	945	945	945	945	-	-	937(-8)
	c840-5	951	951	950(-1)	951	951	951	951	951	951	951	-	-	943(-8)
gap8 (8×48)	c848-1	1133	1133	1131(-2)	1129(-4)	1129(-4)	1131(-2)	1131(-2)	1132(-1)	1130(-3)	1129(-4)	-	-	1124(-9)
	c848-2	1134	1134	1131(-3)	1133(-1)	1132(-2)	1134	1133(-1)	1134	1133(-1)	1134	-	-	1126(-8)
	c848-3	1141	1141	1138(-3)	1138(-3)	1137(-4)	1139(-2)	1139(-2)	1140(-1)	1139(-2)	1139(-2)	-	-	1132(-9)
	c848-4	1117	1117	1113(-4)	1115(-2)	1113(-4)	1116(-1)	1115(-2)	1117	1116(-1)	1116(-1)	-	-	1111(-6)
	c848-5	1127	1127	1124(-3)	1124(-3)	1124(-3)	1125(-2)	1125(-2)	1125(-2)	1125(-2)	1125(-2)	-	-	1123(-4)
gap9 (10×30)	c1030-1	709	709	708(-1)	709	708(-1)	709	709	709	709	709	-	-	704(-5)
	c1030-2	717	717	715(-2)	715(-2)	715(-2)	717	716(-1)	717	716(-1)	716(-1)	-	-	711(-6)
	c1030-3	712	712	711(-1)	712	712	712	712	712	712	712	-	-	711(-1)
	c1030-4	723	723	723	722(-1)	722(-1)	723	723	723	723	723	-	-	717(-6)
	c1030-5	706	706	705(-1)	705(-1)	706	706	706	706	706	706	-	-	697(-9)
gap10 (10×40)	c1040-1	958	958	957(-1)	958	958	958	958	958	958	958	-	-	955(-3)
	c1040-2	963	963	961(-2)	962(-1)	961(-2)	963	963	963	962(-1)	963	-	-	957(-6)
	c1040-3	960	960	957(-3)	957(-3)	957(-3)	958(-2)	958(-2)	959(-1)	959(-1)	959(-1)	-	-	957(-3)
	c1040-4	947	947	944(-3)	946(-1)	945(-2)	947	947	947	947	946(-1)	-	-	944(-3)
	c1040-5	947	947	944(-3)	945(-2)	945(-2)	946(-1)	946(-1)	947	946(-1)	946(-1)	-	-	941(-6)
gap11 (10×50)	c1050-1	1139	1139	1139	1138(-1)	1138(-1)	1139	1139	1139	1139	1139	-	-	-
	c1050-2	1178	1178	1178	1178	1177(-1)	1178	1178	1178	1178	1178	-	-	-
	c1050-3	1195	1195	1195	1195	1195	1195	1195	1195	1195	1195	-	-	-
	c1050-4	1171	1171	1167(-4)	1170(-1)	1170(-1)	1171	1171	1171	1171	1171	-	-	-
	c1050-5	1171	1171	1167(-4)	1170(-1)	1169(-2)	1171	1171	1171	1171	1171	-	-	-
gap12 (10×60)	c1060-1	1451	1451	1449(-2)	1450(-1)	1449(-2)	1451	1450(-1)	1451	1450(-1)	1450(-1)	-	-	-
	c1060-2	1449	1449	1449	1448(-1)	1448(-1)	1449	1449	1449	1449	1449	-	-	-
	c1060-3	1433	1433	1433	1433	1432(-1)	1433	1433	1433	1433	1433	-	-	-
	c1060-4	1447	1447	1444(-3)	1446(-1)	1446(-1)	1447	1447	1447	1447	1447	-	-	-
	c1060-5	1446	1446	1444(-2)	1445(-1)	1445(-1)	1446	1446	1446	1446	1445(-1)	-	-	-
적중률 (%)		60/60 (100.0)	30/60 (50.00)	10/30 (33.33)	6/30 (20.00)	24/30 (80.00)	21/30 (70.00)	26/30 (86.67)	20/30 (66.67)	19/30 (63.33)	-	-	-	18/50 (36.00)
순위		1	7	9	10	3	4	2	5	6	-	-	-	8

Opt	Profits(p _{ij})																								Sum	
564	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	Sum(f)	b-f
M1	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	74	0
M2	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	24	0	0	0	0	0	0	0	0	48	0
M3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	23	0	0	0	46	0
M4	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	41	1
M5	0	21	25	22	0	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	116	0
M6	0	0	0	0	0	0	0	0	23	0	0	0	0	0	25	0	0	25	0	0	0	0	0	25	98	0
M7	0	0	0	0	0	24	0	24	0	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	0
M8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	24	0	0	0	0	25	49	0

(m) c824-3

Opt	Profits(p _{ij})																								Sum	
568	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	Sum(f)	b-f
M1	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	24	0	0	0	0	0	25	0	23	96	0
M2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	20	0	0	0	68	0
M3	0	0	25	24	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73	0
M4	0	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	0
M5	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	25	0	0	0	69	0
M6	0	0	0	0	0	0	0	0	0	0	25	0	0	0	0	0	0	0	0	0	0	0	25	0	75	0
M7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	25	0	70	0
M8	25	0	0	0	0	0	0	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	46	0

(n) c824-4

Opt	Profits(p _{ij})																								Sum	
570	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	Sum(f)	b-f
M1	0	0	0	0	0	0	0	0	0	20	23	0	0	0	0	0	0	0	0	0	25	0	0	0	117	0
M2	23	0	0	0	0	0	0	25	0	25	0	0	0	0	0	0	0	0	0	0	0	0	0	18	91	0
M3	0	20	0	0	24	24	0	0	0	0	0	0	0	0	0	21	0	0	0	0	0	0	0	20	133	0
M4	0	0	0	24	0	0	0	0	20	0	0	0	0	24	0	0	0	0	0	21	23	0	24	0	156	0
M5	0	0	24	0	0	0	0	0	24	0	0	0	0	0	0	0	23	22	0	0	0	0	0	0	93	0

(o) c525-4

Opt	Profits(p _{ij})																								Sum	
564	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	Sum(f)	b-f
M1	24	0	0	0	0	0	0	0	0	0	0	25	0	0	0	20	0	0	0	24	0	0	0	0	93	0
M2	0	0	23	0	0	0	0	24	0	22	0	20	0	0	0	0	0	0	22	0	0	18	0	0	129	0
M3	0	0	0	0	0	0	0	0	23	0	0	0	21	0	0	24	0	0	0	0	0	0	0	23	116	0
M4	0	22	0	0	0	25	0	0	0	0	0	0	22	0	0	0	0	0	0	0	0	0	19	0	88	0
M5	0	0	0	25	24	0	23	0	0	0	0	0	0	0	0	0	25	0	0	21	0	0	0	0	138	0

(p) c525-5

Opt	Profits(p _{ij})																								Sum	
656	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	Sum(f)	b-f
M1	0	0	0	0	0	19	0	0	0	0	0	20	0	0	0	22	0	0	0	0	0	0	0	25	105	0
M2	0	20	0	0	22	0	0	0	24	0	23	0	22	22	0	0	0	0	0	0	0	0	0	24	157	0
M3	19	0	0	0	0	0	22	0	0	0	0	0	0	0	0	22	21	0	0	0	0	25	0	0	130	0
M4	0	0	0	0	23	0	0	0	20	0	0	0	0	0	0	0	0	0	0	24	24	0	25	0	138	0
M5	0	0	16	23	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	21	0	0	0	24	126	0

(q) c530-1

Opt	Profits(p _{ij})																								Sum	
647	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	Sum(f)	b-f
M1	0	0	0	0	0	0	0	0	0	22	0	0	21	0	0	24	0	23	0	0	23	0	0	0	22	135
M2	0	18	23	0	20	0	0	22	0	0	0	0	18	0	0	0	0	0	0	0	0	23	0	0	124	0
M3	0	0	0	24	0	21	0	24	0	23	0	0	0	0	0	0	0	0	0	0	0	0	20	0	112	0
M4	0	0	0	16	0	0	23	0	0	0	0	0	19	0	0	0	0	0	0	0	22	20	0	19	119	0
M5	19	0	0	0	0	0	0	0	0	23	0	0	21	0	0	23	0	22	0	25	24	0	0	0	157	0

(r) c530-4

그림 4. GAP 실험데이터의 k-opt SOA
 Fig. 4. k-opt SOA for GAP experimental data

제안된 방법은 이득 극대화 와 가중치 충족의 2가지 목표 모두 만족하는 대상 선정 기준을 적용하여 데이터를 축소시킨 후 이들 중 각 열에서 최대 이득을 가진 데이터를 선택하여 상자를 채웠다. 용량을 초과한 상자의 물품을 여유가 있는 상자로 옮겨 모든 상자의 용량 제약 조건을 충족시켰다. 마지막으로 GAP에 특화된 이득을 극대화시키는 k-opt 교환 최적화 기법을 제안하여 적용하였다.

제안된 알고리즘을 GAP 벤치마킹 데이터들에 적용한 결과 1/3 데이터는 기준에 알려진 최적 해를 찾는데 성공하였으며, 나머지 2/3 데이터는 근사 해를 구하였다. 이는 GAP에 적용될 수 있는 다항시간 탐욕 알고리즘인 k-opt를 가시적, 수기식으로 적용한 결과 k-opt(k ≥ 3)의 판단 어려움으로 인해 발생된 현상으로, 만약 이를 프로그래밍으로 구현한다면 보다 좋은 결과를 얻을 수 있으리라 기대된다.

따라서 본 연구는 GAP은 다항시간으로 최적 해를 찾아가는 규칙이 존재하지 않는다는 전제하에 컴퓨터의 도움을 받아 메타휴리스틱 기법들로 다항시간으로 근사 해를 찾아가는 방식에서 탈피하여 역발상으로 다항시간 정확한 알고리즘의 존재 가능성에 대한 연구 방향을 전환시킬 수 있는 토대를 제시하였다는 점에서 연구 의의를 찾을 수 있다.

본 논문에서 제안된 k-opt 알고리즘을 보다 향상시킨 진화된 k-opt(advanced k-opt)를 적용한다면 모든 데이터들에 대해 최적 해를 찾을 수 있을 것으로 판단된다. 따라서 추후 이 분야의 연구를 지속적으로 수행할 예정이다.

References

- [1] P. C. Chu and J. E. Beasley, "A Genetic Algorithm for the Generalized Assignment Problem," *Computers & Operations Research*, Vol. 24, No. 1, pp. 17-23, Jan. 1997, [https://doi.org/10.1016/S0305-0548\(96\)00032-9](https://doi.org/10.1016/S0305-0548(96)00032-9)
- [2] L. Özbakir, A. Baykasoğlu, and P. Tapkan, Pınar, "Bees Algorithm for Generalized Assignment Problem," *Applied Mathematics and Computation*, Vol. 215, No. 11, pp. 3782-3795, Feb. 2010, <https://doi.org/10.1016/j.amc.2009.11.018>
- [3] R. M. Naus, "Solving the Generalized Assignment Problem: An Optimizing and Heuristic Approach," *Informatics Journal on Computing*, Vol. 15, No. 3, pp. 249-266, Aug. 2003,

[https://doi.org/ 10.1287/ijoc.15.3.249.16075](https://doi.org/10.1287/ijoc.15.3.249.16075)

- [4] P. Avella, M. Boccia, and I. Vasilyev, "A Computational Study of Exact Knapsack Separation for the Generalized Assignment Problem," Computational Optimization and Applications, Vol. 45, No. 3, pp. 543-555, Apr. 2010, <https://doi.org/10.1007/s10589-008-9183-8>
- [5] I. H. Osman, "Heuristics for the Generalised Assignment Problem : Simulated Annealing and Tabu Search Approaches," Operations Research Spektrum, Vol. 17, pp. 211-225, Dec. 1995.
- [6] S. U. Lee, "Time-Cost Optimization Algorithm for Generalized Assignment Problem," Journal of KIIT, Vol. 14, No. 5, pp. 147-153, May 2016, <http://doi.org/10.14801/jkiit.2016.14.5.147>
- [7] H. R. Lourenço and D. Serra, "Adaptive Approach Heuristics for the Generalized Assignment Problem," Economics Working Papers 288, pp. 1-28, Department of Economics and Business, Universitat Pompeu Fabra. May 1998, <http://hdl.handle.net/10230/292>
- [8] K. Supriya, B. Kajla, and M. Sathi, "Solution of a Class of Generalized Assignment Problem," Journal of Intelligent & Fuzzy Systems, Vol. 33, No. 3, pp. 1687-1697, Aug. 2017, <https://doi.org/10.3233/JIFS-17086>
- [9] M. B. Akbulut and E. Yilmaz "A Modified Genetic Algorithm for the Generalized Assignment Problem," Istanbul University - Journal of Electrical and Electronics Engineering, Vol. 9, No. 2, pp. 951-958, Jul. 2009.
- [10] J. E. Beasley, "OR-Library : Generalized Assignment Problem," <http://people.brunel.ac.uk/~astjjb/eb/orlib/gapinfo.html>, Retrieved Jul. 2021.
- [11] T. H. Man, "An Algorithm for Multi-Objective Assignment Problem," Master Thesis, Chinese University of Hong Kong, Jun. 2005.
- [12] SAS, "SAS Optimization 8.3 Mathematical Optimization Procedures: Chapter 18. The Decomposition Algorithm, Example 18.2. Generalized Assignment Problem," pp. 902-908, SAS, Jul. 2018.

저 자 소 개

이 상 윤(정회원)



- 1987년 : 한국항공대학교 항공전자공학과 (학사)
- 1997년 : 경상대학교 컴퓨터과학과 (석사)
- 2001년 : 경상대학교 컴퓨터과학과 (박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사
- 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3 ~ 2015.3 : 강릉원주대학교 멀티미디어공학과 부교수
- 2015.4 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 인공지능과 빅데이터분석, 최적화 알고리즘
- e-mail : sulee@gwnu.ac.kr