Journal of the Korea Industrial Information Systems Research Vol.27 No.2, Apr. 2022 :61-69
http://dx.doi.org/10.9723/jksiis.2022.27.2.061 ISSN:1229-3741

작업 완료 확률을 고려한 다수 에이전트-다수 작업 할당의 근사 알고리즘+

(Approximation Algorithm for Multi Agents-Multi Tasks Assignment with Completion Probability)

김 광[1)*]
(Gwang Kim)

요 약 다수 에이전트 시스템(Multi-agent system)은 에이전트 각자의 결정으로 최상의 조직화된 결정을 달성하는 것을 목표로 하는 시스템으로 본 논문에서는 다수 에이전트-다수 작업의 할당 문제를 제시한다. 본 문제는 각 에이전트가 하나의 작업에 할당이 되어 수행하고, 작업 수행에 대한 작업 완료 확률(completion probability)이 있으며 모든 작업의 수행 확률을 최대화하는 할당을 결정한다. 비선형(non-linearity)의 목적함수와 조합 최적화(combinatorial optimization)로 표현되는 본 문제는 NP-hard로, 효과적이면서 효율적인 문제 해결 방법론 제시가 필요하다. 본 연구에서는 한계이익(marginal gain)의 감소를 의미하는 하위모듈성(submodularity)을 활용한 근사 알고리즘(approximation algorithm)을 제안하고, 확장성(scalability)과 강건성(robustness) 측면에서 우수한 알고리즘임을 이론 및 실험적으로 제시한다.

핵심주제어: 다수 에이전트 시스템, 작업 할당 문제, 근사 알고리즘, 하위모듈성, 조합 최적화

Abstract A multi-agent system is a system that aims at achieving the best-coordinated decision based on each agent's local decision. In this paper, we consider a multi agent-multi task assignment problem. Each agent is assigned to only one task and there is a completion probability for performing. The objective is to determine an assignment that maximizes the sum of the completion probabilities for all tasks. The problem, expressed as a non-linear objective function and combinatorial optimization, is NP-hard. It is necessary to design an effective and efficient solution methodology. This paper presents an approximation algorithm using submodularity, which means a marginal gain diminishing, and demonstrates the scalability and robustness of the algorithm in theoretical and experimental ways.

Keywords: Multi-agent system, Task assignment problem, approximation algorithm, Submodularity, Combinatorial optimization

* Corresponding Author: gwangkim91@chosun.ac.kr
+ 이 논문은 2021학년도 조선대학교 학술연구비의 지원을 받아 연구되었음.

Manuscript received January 03, 2022 / revised February 18, 2022 / accepted February 28, 2022
1) 조선대학교 경영학부, 제1저자, 교신저자

1. 서 론

다수 에이전트 시스템(Multi-agent system)은 상호 작용하는 여러 에이전트가 존재하고, 이들을 활용해 최상의 시스템을 구성하는 목표를 가지고 있다. 각 에이전트는 자신의 국부적 결정(local decision)을 진행하지만, 이를 조합해 시스템 측면에서는 최상의 조직화된 결정(coordinated decision)을 달성하는 것을 목표로 한다(Terelius et al., 2011; Yuan et al., 2011). 최근에는 로봇, 무인항공기 등 자동화된(autonomous) 개체를 가지고 교통 관리, 에러 감지, 헬스 모니터링 등 다양한 분야에 다수 에이전트 시스템이 적용되고 있다(Jung et al., 2007; Pöllänen et al., 2009; Wu et al., 2010; Heilig et al., 2017; Jeong et al., 2020; Degas et al., 2021).

특히, 본 논문에서 다루는 다수 에이전트를 활용한 작업 할당 문제의 경우 각 에이전트는 작업 중 하나에 할당이 되어 임무를 수행하고, 각 작업은 할당된 에이전트 그룹에 따라 작업의 수행 결과가 달라진다. 에이전트의 기술개발을 통한 작업 수행 결과의 향상을 기대할 수도 있지만, 현재 가지고 있는 에이전트 능력을 바탕으로 최적의 운영 시스템을 구성하는 것 역시 중요하다. 에이전트와 작업의 수가 많아질수록 작업 할당은 복잡해지고, 이에 대해 효과적이면서 효율적인 할당을 위한 최적화, 시뮬레이션 및 알고리즘 등의 문제 해결 방법이 다양한 연구에서 제안되고 있다(Isler & Bajcsy, 2005; Terelius et al., 2011; Yuan et al., 2011; Douma et al., 2012).

본 논문에서 진행하는 문제 상황은 각 에이전트가 하나의 작업에 할당이 되어 수행할 경우, 작업을 완료할 수 있는 확률(completion probability)이 각각 정해져 있다고 가정한다. 각 에이전트는 국부적 결정인 하나의 작업을 선정하고 이를 조합해 모든 작업의 수행 확률을 최대화하는 조직화된 결정을 찾는 문제로 설명할 수 있다. 이러한 문제 상황은 센서가 부착된 무인항공기의 위치 할당 결정, 불확실한 명중률을 고려한 무장-표적 할당 결정, 전문성이 각각 다른 작업자의 작업 할당 결정, 사고/재난 등 불확실한 상황에서 최적의 감지를 목표로 하는 센서의 위치 혹은 이동 경로 결정 등 다양하게 적용할 수 있다(Le Thi et al., 2012; Lee & Shin, 2016; Sun et al., 2017; Qu et al., 2019). 특히, 작업의 수행 실패로 인한 피해가 클수록 최적의 조직화된 할당 결정을 찾는 것이 더욱더 중요하다.

작업 완료 확률을 고려한 다수 에이전트-다수 작업 할당 문제는 각 에이전트가 하나의 작업에 할당이 되어 작업을 수행하는 제약조건을 고려하고, 이는 조합 최적화(combinatorial optimization)로 표현할 수 있다. 모든 작업의 총 작업 수행 확률의 합을 최대화를 목적으로 하고, 각 작업의 수행 확률은 작업에 할당된 에이전트 그룹의 작업 완료 확률을 기반으로 결정된다. 확률을 고려한 목적함수는 비선형(non-linearity)을 나타내어, 본 논문에서 다루는 할당 문제는 NP-난해(NP-hard)임을 알 수 있다(Nemhauser et al., 1978; Fisher et al., 1978).

본 논문에서 할당 문제에 대한 수리적 모형(mathematical formulation)을 제시하나, 이 모형을 활용해 최적해(optimal solution)를 구하는 것은 에이전트와 작업의 수의 규모가 커질수록 계산 복잡도(time complexity)가 기하급수적으로 커져 불필요하게 많은 계산 시간을 요구하게 된다. 계산 복잡도를 해결하기 위해 유전알고리즘, ant colony 알고리즘(개미집단 알고리즘), cross-entropy 알고리즘(교차 엔트로피 알고리즘) 등의 방법론을 활용한 기존 연구들이 제안되었다(Le Thi et al., 2012; Li et al., 2015; Bai et al., 2018, Yun et al., 2019).

기존 연구에서 제안한 방법론은 변수의 개수가 증가하여도 적절한 계산 시간 내에 가능해를 찾는 확장성(scalability)은 해결할 수 있으나, 방법론을 통해 찾은 가능해의 목적함수 값이 최적해의 목적함수 값의 얼마만큼을 보장해주는지에 대한 강건성(robustness) 측면에서 다루는 연구는 거의 없었다. 따라서, 본 논문에서는 할당 문제에 대한 확장성과 강건성을 모두 만족하는 문제 해결 방법론을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본

논문에서 다루는 작업 완료 확률을 고려한 다수 에이전트-다수 작업 할당 문제의 수리적 모형을 제시한다. 3장에서는 하위모듈성(submodularity)에 대한 설명 및 본 문제의 목적함수가 하위모듈성을 만족하는지 제시한다. 더불어, 하위모듈성을 활용해 확장성과 강건성을 보장하는 근사 알고리즘(approximation algorithm)을 제안한다. 4장에서는 다양한 수치 실험을 통해 근사 알고리즘의 확장성과 강건성을 실험적으로 보이고자 한다. 마지막 5장에서는 본 연구의 결론에 대해 논한다.

2. 할당 문제의 수리적 모형

작업 완료 확률(completion probability)을 고려한 다수 에이전트-다수 작업 할당 문제는 일반적으로 이분 그래프(bipartite graph)를 활용해 표현할 수 있다. Fig. 1은 이분 그래프 $G=(V_1, V_2, E)$로 표현한 문제 상황을 보여주고 있다.

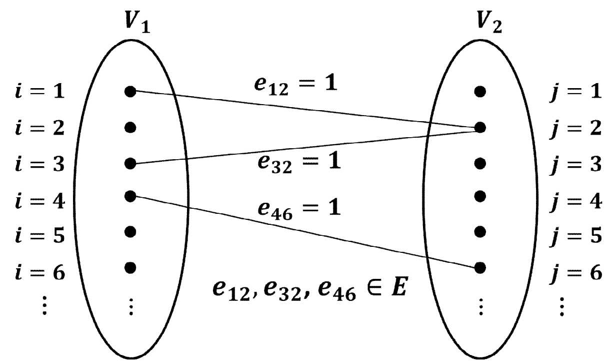

Fig. 1 Example of Bipartite Graph

V_1은 에이전트로 표현되는 노드(node)의 집합을 V_2는 작업으로 표현되는 노드의 집합을 의미하고, 이분 그래프 특성상 E는 V_1과 V_2의 각각 하나의 노드와 연결되는 변(edge)의 집합을 의미한다. 하나의 변은 하나의 결정 변수(decision variable)로 표현할 수 있는데, 식 (1)과 같이 이진 변수(binary variable)로 나타낸다. $v_i \in V_1$ 이고 $v_j \in V_2$ 일 때 노드 v_i와 노드 v_j를 연결하게 되면 '1'의 값을, 그렇지 않으면 '0'의 값을 나타낸다.

$$e_{ij} := \begin{cases} 1, \text{ if } v_i \text{와 } v_j \text{가 연결} \\ 0, \ otherwise \end{cases} \tag{1}$$

e_{ij}가 '1'의 값을 갖는 것은 에이전트 $i(v_i)$가 작업 $j(v_j)$에 할당이 되어 업무를 수행한다는 의미이다. 본 논문에서 진행하는 할당 문제는 각 에이전트가 작업 중 하나에 할당이 되어 임무를 수행하기 때문에 식 (2)와 같이 제약조건을 설정할 수 있다.

$$\sum_{j=1}^{N_2} e_{ij} = 1, \ \forall i \in V_1 \tag{2}$$

N_1과 N_2 는 V_1의 에이전트 수와 V_2의 작업 수를 의미한다. 위의 제약조건을 통해 알 수 있는 가능해(feasible solution)의 조건을 살펴보면, 각 에이전트는 하나의 변으로 임의의 작업과 연결이 이루어져야 하는 것이다. 각 작업의 관점에서 보면, 하나의 작업을 수행하기 위해 할당된 에이전트의 수는 0부터 N_1개까지 가능하다. 에이전트 i가 작업 j에 할당이 되면서 작업을 완료할 수 있는 확률을 p_{ij}라고 할 때, 작업 j에 할당된 에이전트 집합을 통해 작업 j의 수행 확률 R_j로 정의하고 이를 구하면 식 (3)과 같다.

$$R_j := 1 - \prod_{i=1 \ | \ e_{ij}=1}^{N_1} (1 - p_{ij}), \ \forall j \in V_2 \tag{3}$$

각 작업의 수행 확률을 모두 더한 $\sum_{j=1}^{N_2} R_j$ 값이 본 논문에서 다루는 할당 문제의 목적함수 값이 되고 최대화(maximization)를 목표로 한다. 이는 비선형(non-linearity)을 나타내고, 제약조건을 포함한 최종 수리적 모형은 NP-난해(NP-hard)이다. 본 논문에서는 할당 문제를 해결하기 위해 확장성과 강건성을 보장하는 근사 알고리즘(approximation algorithm)을 제안한다. 이는 목적함수의 하위모듈성(submodularity) 특성을 이용해 알고리즘을 고안하므로 본 논문 3

장에서 하위모듈성에 대한 설명과 본 할당 문제의 목적함수가 하위모듈성을 만족하는지에 대한 증명을 진행한다.

3. 하위모듈성 기반 근사 알고리즘

3.1 하위모듈성 (submodularity)

하위모듈성(submodularity)은 집합 함수(set function)에서 정의된다. 집합 함수는 정의역이 주어진 집합의 부분 집합을 의미하고 공역이 실수를 나타내는 함수를 의미한다. 본 할당 문제에서 이를 설명하자면, 에이전트 집합 $V_1 = \{1, 2, ..., N_1 RIGHT$의 부분 집합을 정의역으로 갖는 작업 j의 수행 확률은 집합 함수 '$\widetilde{R}_j(A)$'로 식 (4)와 같이 표현할 수 있다.

$$\widetilde{R}_j(A) := 1 - \prod_{i \in A \mid e_{ij} = 1} (1 - p_{ij}), \ \forall A \subseteq V_1, j \in V_2 \tag{4}$$

만약 작업 j에 할당된 에이전트가 1과 2라고 하면, 부분 집합 $\{1, 2\}$가 함수의 입력값이 되고 함수의 출력값은 $1 - (1 - p_{1j}) * (1 - p_{2j})$가 된다.

식 (4)는 하위모듈성을 갖는데, 하위모듈성은 Def. 1과 같이 정의한다.

Def. 1 집합 S의 임의의 두 부분 집합 S_1, S_2가 있고, $S_1 \subset S_2 \subseteq S$라 하자. S_2에 포함되지 않은 임의의 원소 s가 있다고 할 때, 집합 함수 f가 $f(\{s RIGHT \cup S_2) - f(S_2) \le f(\{s RIGHT$를 항상 만족하는 경우 집합 함수 f는 하위모듈성을 갖는다.

즉, 하위모듈성은 하나의 원소가 집합 함수의 입력값에 추가되면서 얻게 되는 한계 이익(marginal gain)이 부분 집합의 크기가 커질수록 감소하는 특성을 의미한다. 본 할당 문제에서 예시를 들면, 작업 j에 $\{1, 2\}$와 $\{1, 2, 3\}$이 할당되는 두 가지 경우가 있다고 하자. 이때, 새로운 에이전트 '4'가 작업 j에 할당이 되어 추가로 얻게 되는 작업 수행 확률의 증가량은 $\{1, 2, 3\}$보다 $\{1, 2\}$에서 더 큼을 의미한다. 본 할당 문제의 목적함수를 집합 함수로 표현했을 때, 부분 집합의 크기가 커질수록 출력값이 증가하는 단조 증가(monotone increasing)와 하위모듈성을 만족함을 Thm. 1에서 보인다.

Thm. 1 임의의 부분 집합 $A \subseteq V_1$에서, $\sum_{j=1}^{N_2} \widetilde{R}_j(A)$는 단조 증가와 하위모듈성을 만족한다.

Proof. 함수의 합은 단조 증가와 하위모듈성을 모두 유지하므로, 집합 함수 '$\widetilde{R}_j(A)$'에서 두 가지의 특징을 만족하는지 살펴본다. 두 부분 집합 $A_1 \subset A_2 \subseteq V_1$이 있다고 가정한다.

$$1 - \widetilde{R}_j(A_1) = \prod_{i \in A_1 \mid e_{ij} = 1} (1 - p_{ij})$$

$$1 - \widetilde{R}_j(A_2) = \prod_{i \in A_2 \mid e_{ij} = 1} (1 - p_{ij})$$

위의 두 가지 식을 구할 수 있다. A_2가 더 큰 집합이므로 $1 - \widetilde{R}_j(A_2) \le 1 - \widetilde{R}_j(A_1)$을 만족한다. 즉, $\widetilde{R}_j(A_1) \le \widetilde{R}_j(A_2)$이므로 단조 증가 함수를 만족한다.

A_2에 포함되지 않은 원소 'a'가 있다고 가정한다.

$$\begin{aligned}
&\widetilde{R}_j(\{a\} \cup A_2) = 1 - \prod_{i \in \{a\} \cup A_2 \mid e_{ij} = 1} (1 - p_{ij}) \\
&= 1 - (1 - p_{aj}) \prod_{i \in A_2 \mid e_{ij} = 1} (1 - p_{ij}) \\
&= 1 - \prod_{i \in A_2 \mid e_{ij} = 1} (1 - p_{ij}) + p_{aj} \prod_{i \in A_2 \mid e_{ij} = 1} (1 - p_{ij}) \\
&= \widetilde{R}_j(A_2) + p_{aj} \prod_{i \in A_2 \mid e_{ij} = 1} (1 - p_{ij})
\end{aligned}$$

위의 식으로 계산과정을 나타낼 수 있다.

정리해보면, 아래의 식을 만족한다.

$$\widetilde{R}_j(\{a\} \cup A_2) - \widetilde{R}_j(A_2) = p_{aj} \prod_{i \in A_2 \mid e_{ij}=1} (1-p_{ij})$$

같은 원리로 A_1에서 구하면, 아래의 식을 만족한다.

$$\widetilde{R}_j(\{a\} \cup A_1) - \widetilde{R}_j(A_1) = p_{aj} \prod_{i \in A_1 \mid e_{ij}=1} (1-p_{ij})$$

앞서 단조 증가 함수 증명을 통해 $\prod_{i \in A_1 \mid e_{ij}=1} (1-p_{ij}) \geq \prod_{i \in A_2 \mid e_{ij}=1} (1-p_{ij})$임을 알고 있으므로, $\widetilde{R}_j(\{a\} \cup A_2) - \widetilde{R}_j(A_2)$보다 $\widetilde{R}_j(\{a\} \cup A_1) - \widetilde{R}_j(A_1)$값이 항상 크거나 같다. 이는 하위모듈성을 만족한다. ■

본 논문의 수리적 모형은 결과적으로 하위모듈성을 갖는 목적함수를 최대화하는 문제로 표현할 수 있고, 식 (2)와 같이 할당 제약조건을 갖는 경우 NP-난해(NP-hard) 문제로 알려져 있다(Nemhauser et al., 1978; Fisher et al., 1978). 이러한 이유로 확장성을 보장하는, 즉 빠른 시간 내에 효과적인 가능해를 찾는 알고리즘 기반의 문제 해결 방법론이 요구된다. 본 논문에서는 탐욕 기반(greedy-type) 알고리즘을 3.2절에서 제안한다. 그리고, 제안한 알고리즘이 앞서 언급한 확장성뿐만 아니라 근사 최적해(near-optimal solution)를 보장하는 강건성을 가지고 있음을 하위모듈성의 특성을 활용해 보이고자 한다.

3.2 탐욕 기반 근사 알고리즘

본 논문의 할당 문제의 최적해를 구하는 경우, 입력값의 규모가 커질수록 계산 복잡도가 기하급수적으로 늘어나 많은 계산 시간을 요구한다. 본 논문에서는 목적함수의 하위모듈성을 이용해 단순하지만, 효과적인 가능해를 빠른 시간에 찾는 탐욕 기반(greedy-type)의 '순서 탐욕 알고리즘(Alg. 1)'을 제안한다. 이는 Conforti and Cornuéjols (1984)와 Qu et al. (2019) 논문에서 제시한 알고리즘을 기반으로 고안하였다. Alg. 1의 자세한 절차는 다음과 같다. $\widetilde{e}_{ij}$는 결정 변수 e_{ij}의 의미에 해당하는 집합 내 원소 표현이라고 하자.

Step 1. (초기화)
 $e_{ij} \leftarrow 0 \ \forall i \in V_1, j \in V_2$; 초기해
 $A \leftarrow \varnothing$; 초기 집합
Step 2. (에이전트 순서 결정)
 N_1개의 에이전트를 임의로 1번부터 N_1번까지 순서를 결정 $(i = 1, 2, \dots, N_1)$;
 $i \leftarrow 1$; 1번 에이전트부터 할당
Step 3. (순서에 따른 탐욕 알고리즘 진행)
 $j^* \leftarrow \underset{j}{\arg max}[\sum_{j=1}^{N_2} \widetilde{R}_j(A \cup \{\widetilde{e}_{ij}\}) - \sum_{j=1}^{N_2} \widetilde{R}_j(A)]$;
 $e_{ij^*} \leftarrow 1$; 작업 j^*에 에이전트 i 할당
 $A \leftarrow A \cup \{\widetilde{e_{ij^*}} RIGHT$; 현재 집합 업데이트
 $i = N_1$이면 Step 3 종료;
 아니면, $i \leftarrow i+1$ 그리고 Step 3 반복; 다음 에이전트 할당 진행
Step 4. (해와 목적함수 값 도출)
 Step 3에서 구한 e_{ij}=1이 할당 문제의 가능해,
 대응하는 $\sum_{j=1}^{N_2} R_j$ 값이 목적함수 값;

일반 탐욕 알고리즘과 달리 Alg. 1은 Step 2에서 에이전트의 순서를 하나 먼저 결정한다. 그리고 순서에 따라 각 에이전트는 현재 상황에서 가장 큰 한계 이익을 보이는 작업에 할당하는 규칙을 Step 3에서 나타낸다. 순서가 미리 결정되어 알고리즘을 진행하게 되면 알고리즘의 iteration 수가 감소해 일반 탐욕 알고리즘보다 더 빠른 가능해를 도출할 수 있다. 실제 Alg. 1의 최대 계산 횟수는 $N_1 \cdot N_2$로 다항시간(polynomial time) 내에 가능해를 도출할 수 있어 확장성을 보장한다.

다음으로 Alg. 1이 근사 알고리즘임을 제시한다. 어떠한 에이전트-작업 상황에도 알고리즘으로 구한 가능해의 목적함수 값이 최적해의 목적함수 값을 얼마만큼 보장해주는지를 의미하는 근사 비율(approximation ratio)을 Thm. 2에서 제시한다. 제시한 근사 비율은 하위모듈성 특성

을 활용해 얻을 수 있으며, 자세한 증명과정은 Conforti and Cornuéjols (1984)와 Qu et al. (2019) 논문을 참고하기 바란다.

Thm. 2 Alg. 1을 통해 구한 해의 목적함수 값을 R_{ALG}이라 하고, 최적 목적함수 값을 R_{OPT}라고 하자. 그리고, 집합 함수 $\widetilde{R}_j(A)$를 활용해 아래의 값을 구한다고 하자.

$c_j = \max_{i \in V_1}[1 - \dfrac{\widetilde{R}_j(N_1) - \widetilde{R}_j(N_1 - \{i\})}{\widetilde{R}_j(\{i\})}]$ 이고, $c = \max_{j \in V_2} c_j$ $(0 \le c \le 1)$라고 할 때, 근사 비율은 $\dfrac{1}{1+c}$이다. 즉, $\dfrac{R_{ALG}}{R_{OPT}} \ge \dfrac{1}{1+c}$를 항상 보장한다.

Thm. 2에서 제시한 근사 비율을 살펴보면, 에이전트-작업의 상황과 작업 완료 확률에 따라 c값은 달라진다. 만약 하나의 case에서 c값이 0.1이 나온 경우, Alg. 1을 통해 구한 목적함수 값과 최적값(optimal value)의 백분율 상대오차는 최악의 경우라도 최대 {1-(1/1.1)}*100(%) ≒ 9%를 넘지 않는다는 것을 의미한다.

4. 수치실험

4.1 실험 설정

작업 완료 확률을 고려한 다수 에이전트-다수 작업 할당 문제를 다루기 위해 본 연구에서는 확장성과 강건성을 지닌 '순서 탐욕 알고리즘(Alg. 1)'을 제안하였다. 4장에서는 여러 dataset을 이용하여 Alg. 1의 확장성과 강건성을 수치실험을 통해 보이고자 한다. Alg. 1에서 구한 해의 성능을 비교하기 위해, 브루트-포스(brute-force) 알고리즘을 이용해 분석한다. 브루트-포스 알고리즘의 경우 '완전탐색 알고리즘'으로 알려져 있다, 이는 가능한 모든 경우의 수를 다 확인하는 방법으로 할당 문제의 최적해를 구할 수 있다. 정수 최적화(Integer Programming)에서 다루는 분기한정법(branch-and-bound)이나 분기절단법(branch-and-cut) 등을 통한 최적해를 구하는 방법이 있다. 하지만 목적함수가 비선형인 경우, 특정 문제를 제외 항상 최적해를 보장하지 않아 본 실험에서는 브루트-포스 알고리즘을 가지고 비교한다.

브루트-포스 알고리즘의 경우 계산 횟수가 $N_2^{N_1}$으로, 계산 시간이 3,600초 이내로 가능한 dataset에서 진행한다. 크게 다음과 같이 네 가지의 Case에서 진행하고, 각 Case당 무작위로 생성된 열 개의 data를 이용한다. 각 Case의 실험 설정은 Table 1과 같다. 하나의 작업에 여러 에이전트가 할당될 수 있도록 $N_1 > N_2$로 설정하였다. U[a, b]는 폐구간 [a, b]에서 정의된 균일분포(uniform distribution)를 의미한다.

Table 1 Parameter setting

Parameter	Setting Value
(N_1, N_2)	Case 1: (5, 3), Case 2: (8, 3), Case 3: (10, 5), Case 4: (11, 5)
p_{ij}	U[0.3, 0.8]

4.2 실험 결과분석

네 가지 Case에 대한 브루트-포스 알고리즘과 Alg. 1의 실험 결과는 Table 2에 제시된다. 실험은 모두 동일한 컴퓨터 환경(CPU: Intel(R) Core(TM) i5-10500, RAM: 16GB, OS: Windows 10)에서 진행되었고, 알고리즘 실행을 위한 프로그래밍 언어로는 'Python 3'을 사용하였다. Table 2에 제시된 결과값은 무작위로 생성된 data를 이용해 나온 결과의 평균값을 의미한다.

Table 2의 결과를 살펴보면, 네 가지 Case 모두에서 R_{ALG} 값은 실제 최적해와의 차이가 2% 이내로 나타난 것을 확인할 수 있다. Case 1의 경우는 R_{ALG} / R_{OPT} 값이 평균적으로 0.99 이상으로 1% 미만의 gap을 보였고, 나머지 Case

에서도 2% 이내임을 나타냈다. 근사 비율은 0.5 ~ 0.6으로 실제 gap과 다소 차이가 나는 것으로 보이나, 이는 최악의 상황에서도 Alg. 1에서 구한 해가 예외 없이 최적해의 일정 수준을 보장한다는 강건성을 의미한다. **Thm.** 2에서 구한 c 값의 공식을 살펴보면, 모든 i와 j에서 p_{ij}값이 낮아질수록 그리고 N_1값보다 N_2값이 커질수록 근사 비율은 1에 더 가까워 Alg. 1만으로도 근사 최적해를 구할 수 있는 확률이 높아진다. 실제 $p_{ij} \in$ U[0.01, 0.03]으로 실험한 결과, **Thm.** 2의 근사 비율은 평균적으로 0.93 이상 나와 높은 수준의 근사 최적해를 확인할 수 있다.

Case 1에서 Case 4로 갈수록 N_1과 N_2값이 커지고, 이는 두 가지 실험방법 사이 계산 횟수가 기하급수적으로 차이가 난다는 것을 실험적으로 보일 수 있다. Case 1에서는 브루트-포스 방법과 Alg. 1 사이의 계산 시간이 평균 10배 정도 차이가 난 것과 달리, Case 4에서는 평균 1×10^6배 차이가 난 것을 확인할 수 있다. 그에 비해 R_{ALG} / R_{OPT} 값은 Case 별로 크게 차이가 나지 않아, Alg. 1은 상대적으로 빠른 시간 내 효과적인 해를 구할 수 있음을 확인하였다. 즉, 이론적으로 살펴보았던 확장성과 강건성을 실험적으로도 확인할 수 있다.

Table 2 Experiment Results

	Case 1	Case 2	Case 3	Case 4
R_{ALG}	2.301	2.755	4.461	4.502
R_{OPT}	2.318	2.791	4.547	4.593
R_{ALG} / R_{OPT}	0.992	0.987	0.981	0.980
Approximation ratio in **Thm.** 2	0.601	0.550	0.542	0.525
Avg. comput. time in brute-force (sec.)	0.003	1.03	202.4	1042.35
Avg. comput. time in Alg. 1 (sec.)	0.0003	0.0004	0.001	0.001

Case 4보다 크기가 더 큰 dataset의 경우, 브루트-포스 방법에서는 계산 횟수가 1×10^8회가 넘고 3,600초 이내에 최적해를 찾기가 어렵다. 좀 더 큰 사이즈의 dataset을 이용해 Alg. 1의 계산 시간을 제시한 표는 Table 3과 같다. 각 Case당 무작위로 생성된 열 개의 data를 이용해 평균을 낸 값을 제시한다. N_1과 N_2값이 증가할수록 계산 시간은 증가하지만, N_1과 N_2값이 각각 1,000개, 30개 이내에서는 40초 내외로 Alg. 1을 이용한 근사 최적해를 구할 수 있다.

Table 3 Computation times in Alg. 1

(N_1, N_2)	Avg. comput. time (sec)
(100, 10)	0.09
(250, 15)	1.89
(500, 20)	6.07
(750, 25)	19.9
(1000, 30)	43.5

5. 결 론

본 논문에서는 각 에이전트가 하나의 작업에 할당이 되어 수행하고, 작업 수행에 대한 작업 완료 확률(completion probability)이 존재하는 다수 에이전트-다수 작업 할당 문제를 제시하였다. 작업 할당 문제는 일반적으로 이분 그래프(bipartite graph)를 활용한 조합 최적화(combinatorial optimization)문제로 표현할 수 있었다.

목적함수는 비선형(non-linearity)이고, 제약조건을 포함한 수리적 모형은 NP-난해(NP-hard)인 작업 할당 문제를 효과적으로 풀기 위해 근사 알고리즘(approximation algorithm)인 ‘순서

탐욕 알고리즘'을 제안하였다. '순서 탐욕 알고리즘'은 할당 문제의 목적함수가 가지는 하위모듈성(submodularity)을 활용하였다. 기존에 제안되었던 메타휴리스틱(metaheuristic) 방법론과 달리 본 연구에서 제안하는 근사 알고리즘은 문제의 크기가 커져도 적절한 계산 시간 내에 가능해를 찾는 확장성(scalability)과 해의 품질을 보장해주는 강건성(robustness)을 모두 만족하는 알고리즘임을 이론적으로 제시하였다. 또한, 다양한 수치 실험을 통해 실험적으로도 제안한 알고리즘의 효과성과 효율성을 확인하였다.

본 알고리즘은 작업 할당 문제뿐만 아니라 하위모듈성을 갖는 최대화 문제에서 활용해 근사 최적해를 구하는 방법으로 기대해 볼 수 있다. 본 연구에서는 작업 완료 확률이 에이전트-작업별 고정이었으나, 좀 더 불확실성을 반영한 추계적(stochastic) 확률을 고려한 작업 할당 문제는 추후 개선 연구 방향으로 진행할 수 있을 것이다.

References

Bai, X., Yan, W., Ge, S. S., & Cao, M. (2018). An integrated multi-population genetic algorithm for multi-vehicle task assignment in a drift field. *Information Sciences*, 453, 227-238.

Conforti, M., & Cornuéjols, G. (1984). Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete applied mathematics*, 7(3), 251-274.

Degas, A., Kaddoum, E., Gleizes, M. P., Adreit, F., & Rantrua, A. (2021). Cooperative multi-agent model for collision avoidance applied to air traffic management. *Engineering Applications of Artificial Intelligence*, 102, 104286.

Douma, A. M., van Hillegersberg, J., & Schuur, P. C. (2012). Design and evaluation of a simulation game to introduce a multi-agent system for barge handling in a seaport. *Decision support systems*, 53(3), 465-472.

Fisher, M. L., Nemhauser, G. L., & Wolsey, L. A. (1978). An analysis of approximations for maximizing submodular set functions—II. Berlin, Heidelberg. *Polyhedral combinatorics,* pp. 73-87.

Jung, J.J. (2007). Applying CSP techniques to automated scheduling with agents in distributed environment. *Journal of the Korea Industrial Information Systems Research*, 12(1), 87-94.

Heilig, L., Lalla-Ruiz, E., & Voß, S. (2017). port-IO: an integrative mobile cloud platform for real-time inter-terminal truck routing optimization. *Flexible Services and Manufacturing Journal*, 29(3), 504-534.

Isler, V., & Bajcsy, R. (2005, April). The sensor selection problem for bounded uncertainty sensing models. In IPSN 2005. *Fourth International Symposium on Information Processing in Sensor Networks*, 2005. pp. 151-158.

Jeong, H. Y., David, J. Y., Min, B. C., & Lee, S. (2020). The humanitarian flying warehouse. T*ransportation research part E: logistics and transportation review*, 136, 101901.

Le Thi, H. A., Nguyen, D. M., & Dinh, T. P. (2012). Globally solving a nonlinear UAV task assignment problem by stochastic and deterministic optimization approaches. Optimization Letters, 6(2), 315-329.

Lee, J.H. & Shin M.I (2016), Stochastic Weapon Target Assignment Problem under Uncertainty in Targeting Accuracy, *The Korean Operations Research and Management Science Society*, 41(3), 23-36.

Li, J. J., Zhang, R. B., & Yang, Y. (2015). Meta-heuristic ant colony algorithm for multi-tasking assignment on collaborative AUVs. *International Journal of Grid and Distributed Computing*, 8(3), 135-144.

Nemhauser, G. L., Wolsey, L. A., & Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical programming*, 14(1), 265-294.

Pöllänen, R., Toivonen, H., Peräjärvi, K., Karhunen, T., Ilander, T., Lehtinen, J., ... & Juusela, M. (2009). Radiation surveillance using an unmanned aerial vehicle. *Applied radiation and isotopes*, 67(2), 340-344.

Qu, G., Brown, D., & Li, N. (2019). Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions. *Automatica*, 105, 206-215.

Sun, X., Cassandras, C. G., & Meng, X. (2017, December). A submodularity-based approach for multi-agent optimal coverage problems. *2017 IEEE 56th Annual Conference on Decision and Control* (CDC), pp. 4082-4087.

Terelius, H., Topcu, U., & Murray, R. M. (2011). Decentralized multi-agent optimization via dual decomposition. *IFAC proceedings volumes*, 44(1), 11245-11251.

Wu, J., Yuan, S., Ji, S., Zhou, G., Wang, Y., & Wang, Z. (2010). Multi-agent system design and evaluation for collaborative wireless sensor network in large structure health monitoring. *Expert Systems with Applications*, 37(3), 2028-2036.

Yuan, D., Xu, S., & Zhao, H. (2011). Distributed primal‐dual subgradient method for multiagent optimization via consensus algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(6), 1715-1724.

Yun, Y.S. & Chuluunsukh, A. (2019). Green Supply Chain Network Model: Genetic Algorithm Approach. *Journal of the Korea Industrial Information Systems Research*, 24(3), 31-38.

김 광 (Gwang Kim)

• 서울대학교 산업공학과 공학사
• 서울대학교 산업공학과 공학박사
• (현재) 조선대학교 경상대학 경영학부 조교수
• 관심분야: 생산운영관리, 최적화, 문제 해결 알고리즘 개발 등