

Knapsack Problem과 Priority Scheduling을 활용한 지하아이돌 특전회 대기 전략 모델링

Shine Loi Lee

Abstract

본 연구는 지하아이돌 문화의 핵심 요소 중 하나인 **특전회**에서, 팬이 제한된 시간 내에 선호하는 여러 아이돌 멤버와 체크(기념 사진)를 최대한 효율적으로 촬영할 수 있도록 대기열 선택 전략을 최적화하는 문제를 다룬다. 이 문제는 각 멤버별 대기 시간, 체크 촬영 시간, 팬의 선호도, 희망 체크 수 등의 다양한 요소를 동시에 고려해야 하는 조합 최적화 문제로 정의될 수 있다. 이를 해결하기 위해, 본 논문에서는 **Knapsack Problem**의 자원 할당 개념과 **Priority Scheduling**의 우선순위 기반 정렬 방식을 결합한 휴리스틱 기반 알고리즘을 설계하였다. 제안한 알고리즘은 체크 수요와 대기 시간을 고려해 각 아이돌의 선택 우선순위를 점수화하고, 제한된 시간 내 최대 만족도를 달성할 수 있도록 체계적인 선택 전략을 구성한다. 또한 다양한 실험 시나리오에 대한 시뮬레이션을 통해 알고리즘의 효율성과 적용 가능성을 평가하였다.

This study addresses the problem of optimizing queue selection strategies in **tokutenkai**, a key component of indie idol culture, where fans aim to take as many commemorative photos (chekis) as possible with their favorite idols within a limited time frame. The problem is defined as a combinatorial optimization task that simultaneously considers multiple factors, including each member's queue length, cheki shooting time, fan preference, and desired number of chekis. To solve this problem, we propose a heuristic-based algorithm that integrates the resource allocation concept of the **Knapsack Problem** and the priority-based ordering method of **Priority Scheduling**. The proposed algorithm calculates a priority score for each idol based on their cheki demand and queue length, and constructs a systematic selection strategy to maximize fan satisfaction within the time constraint. We evaluate the effectiveness and applicability of the algorithm through simulations under various experimental scenarios, demonstrating its potential as a practical decision-making tool in real-world fan meeting environments.

Contents

1	Introduction	3
2	Related Work	3
2.1	Knapsack Problem	3
2.1.1	Knapsack Problem의 정의	3
2.1.2	Knapsack Problem의 주요 변형	3
2.1.3	본 연구에의 적용	3
2.2	Priority Scheduling	4
2.2.1	Aging	4
2.2.2	본 연구에의 적용	4
3	Methodology	4
3.1	알고리즘 설계	4
3.2	아이돌 정보 입력	5
3.3	Knapsack 기반 아이돌 선택	5
3.3.1	우선순위 가중치 파라미터 탐색	5
3.4	Priority Scheduling 기반 순서 결정	6
3.4.1	1단계: 스케줄 리스트 구성	7
3.4.2	2단계: 우선순위 정렬	7
3.4.3	3단계: 최종 출력 및 분석	7
4	Results	7
5	Discussion	8
6	Conclusion	8
7	References	8

1 Introduction

지하아이돌은 일본 아이돌 문화의 한 갈래로, 음악 방송 등 대중매체보다는 소규모 라이브 공연과 팬과의 대면 이벤트에 중심을 두는 활동 형태를 특징으로 한다. 이러한 지하아이돌 문화는 일본에서 처음 등장한 이후, 최근에는 한국에서도 벤치마킹되어 정착되고 있으며, 서울 마포구를 중심으로 다양한 그룹이 활발히 활동하고 있다. 일반적으로 지하아이돌 공연은 4~6개 그룹이 각 20~30분의 공연 시간을 갖고 출연하며, 공연 이후에는 약 2시간 동안 팬과 직접 소통할 수 있는 대면 이벤트인 **특전회**가 진행된다. 특전회에서는 팬이 굿즈를 구매하거나, ‘체키’라 불리는 즉석 사진을 촬영하고, 아이돌 멤버와 짧은 대화를 나눌 수 있는 기회가 제공된다. 이러한 구조 속에서, 여러 멤버를 응원하는 팬은 제한된 시간 안에 최대한 많은 체키를 효율적으로 촬영하고자 하는 욕구를 가지게 된다. 그러나 팬은 각기 다른 멤버의 줄에 순차적으로 대기해야 하며, 줄의 길이, 선호도, 희망 체키 개수 등 다양한 요인을 종합적으로 고려해야 한다.

이러한 상황은 제한된 자원(시간) 하에서 목표(희망 체키 횟수)를 극대화하려는 **조합 최적화 문제**로 해석할 수 있다. 본 논문에서는 이 문제를 해결하기 위해 두 가지 알고리즘적 모델을 결합한 **휴리스틱 기반 접근법**을 제안한다. 첫 번째는 **Knapsack Problem**(배낭 문제)으로, 제한된 시간이라는 자원 하에 팬이 어떤 멤버를 몇 번 선택할지 결정하는 데 활용된다. Knapsack Problem은 각 항목에 무게와 가치가 부여되었을 때, 전체 무게가 제약 조건을 초과하지 않는 범위 내에서 가치의 합이 최대가 되도록 하는 부분집합을 선택하는 문제이다. 본 연구에서는 이를 기반으로 각 멤버의 대기 시간을 ‘무게’로, 팬의 선호도를 ‘가치’로 대응시켜 모델링하였다. 두 번째는 **Priority Scheduling**으로, Knapsack Problem을 통해 선택된 멤버들 사이에서 어떤 순서로 줄을 설 것인지 결정하기 위해 사용된다. 이 알고리즘은 선호도 기반 우선순위를 고려하면서도 전체 대기 시간을 최소화하는 순서를 구성하는 데 활용된다. 이러한 두 가지 접근을 결합함으로써, 본 연구는 팬이 제한된 특전회 시간 내에서 체계적이고 효율적인 전략을 통해 최대한 많은 체키를 촬영할 수 있도록 지원하는 새로운 대기열 선택 모델을 제시하고자 한다.

2 Related Work

2.1 Knapsack Problem

2.1.1 Knapsack Problem의 정의

Knapsack Problem은 제한된 용량을 가진 배낭에 여러 개의 물건 중 일부를 선택하여, 전체 무게가 용량을 초과하지 않으면서 총 가치를 최대화하는 조합 최적화 문제이다. 각 물건은 고유의 가치와 무게를 가지며, 이들 중 어떤 물건을 선택하느냐에 따라 전체 가치가 달라진다.

2.1.2 Knapsack Problem의 주요 변형

Knapsack Problem은 다양한 현실 문제에 적용될 수 있도록 다음과 같은 변형 형태로도 활발 연구되어 왔다.

- **0-1 Knapsack Problem**: 각 물건을 최대 한 번만 선택할 수 있는 문제이다.
- **Fractional Knapsack Problem**: 각 물건을 분할하여 선택할 수 있으며, 무게 단위당 가치를 기준으로 계산한다.
- **Multi-dimensional Knapsack Problem**: 물건의 무게 외에 부피, 비 등 복수의 제약 조건이 존재하는 문제이다.
- **Bounded Knapsack Problem**: 각 물건을 정해진 최대 개수만큼 선택할 수 있는 문제이다.

2.1.3 본 연구에의 적용

본 연구에서는 팬이 특정 멤버와 여러 번 체키를 찍을 수 있는 상황을 반영하기 위해, **Bounded Knapsack Problem**을 기반으로 문제를 모델링하였다. 해당 문제의 구성 요소는 물건, 무게, 가치, 제한 수량, 배낭 용량으로 정의되며, 이를 본 연구의 상황에 대응시킨 결과는 Table 1과 같다.

체키를 촬영하고자 하는 멤버의 총 수를 n 이라고 하고, 멤버 i 에 대해 체키 한 장당 대기 시간을 w_i , 선호도를 v_i , 최대 체키 수를 b_i 라고 할 때, 우리는 각 멤버와 촬영할 체키의 수 $x_i \in \{0, 1, 2, \dots, b_i\}$ 를 결정해야 한다. 이를 통해 다음과 같은 수학적 최적화 문제로 정식화할 수 있다.

Table 1: Knapsack 문제 요소와 본 연구의 대응 관계

Knapsack 요소	본 연구에서의 해석
물건	체키를 촬영하는 행위
무게	체키 촬영 대기 시간
가치	해당 멤버에 대한 선호도
제한 수량	각 멤버별 희망 체키 수
배낭 용량	특전회 전체 시간

- 제약식:

$$\sum_{i=1}^n w_i \cdot x_i \leq W$$

- 목표 함수:

$$\max \sum_{i=1}^n v_i \cdot x_i$$

위와 같은 모델링을 통해, 팬이 제한된 시간 내에서 선호도와 대기 시간의 균형을 고려하여 최대의 만족을 얻을 수 있는 멤버 선택 조합을 구성할 수 있다.

2.2 Priority Scheduling

Priority Scheduling은 각 프로세스에 우선순위를 부여하고, 우선순위가 높은 프로세스부터 실행하는 스케줄링 알고리즘이다. 동일한 우선순위를 가진 프로세스가 여러 개 존재할 경우에는 일반적으로 도착 순서(FIFO: First-In, First-Out)에 따라 처리된다. 이 알고리즘은 운영체제의 프로세스 스케줄링뿐만 아니라, 병렬 처리 환경이나 자원 배분과 같이 순서가 중요한 다양한 상황에서 널리 활용되고 있다.

Priority Scheduling의 가장 큰 장점은 제한된 자원을 우선순위에 따라 합리적으로 배분할 수 있다는 점이다. 이를 통해 중요한 작업이 빠르게 처리될 수 있도록 보장할 수 있으며, 시스템 전반의 응답성과 효율성을 향상시킬 수 있다.

2.2.1 Aging

Priority Scheduling은 강력한 선별 기능을 제공하지만, 그로 인해 발생할 수 있는 문제점도 존재한다. 그 중 대표적인 문제가 바로 **Starvation**(기아 상태)이다. 이는 낮은 우선순위를 가진 프로세스가 높은 우선순위 프로세스들에게

지속적으로 밀려, 장시간 동안 처리되지 못하는 현상이다. 시스템 내에서 새로운 높은 우선순위 프로세스가 계속 유입되는 경우, 일부 프로세스는 무기한 대기하게 될 수 있다. 이러한 문제를 완화하기 위한 대표적인 해결책이 **Aging** 기법이다. Aging은 시스템에서 오랫동안 대기 중인 프로세스의 우선순위를 일정 시간 간격으로 점진적으로 증가시키는 방식으로, 일정 시간이 지나면 결국 해당 프로세스가 실행될 수 있도록 보장한다. 이 방식은 Starvation을 방지하고, 전체 프로세스 흐름의 공정성과 형평성을 확보하는데 효과적이다.

2.2.2 본 연구에의 적용

본 연구에서는 Priority Scheduling을 활용하여, Knapsack 알고리즘을 통해 선택된 멤버들 사이의 **체키 촬영 순서**를 결정한다. 팬의 선호도, 각 멤버 앞의 대기열 길이, 체키 수요 등을 종합적으로 고려하여 점수를 계산하고, 이 점수를 기반으로 한 우선순위 정렬을 수행함으로써 평균 대기 시간을 최소화하는 순서를 구성한다. 또한 Aging 개념을 응용한 점수 갱신 방식을 통해 우선순위 점수에 시간 경과 요소를 반영하였다. 이를 통해 현실적인 현장 상황에서도 유연한 촬영 순서를 도출할 수 있도록 설계하였다.

3 Methodology

3.1 알고리즘 설계

본 연구에서는 계산 효율성과 구현 편의성을 고려하여 알고리즘을 C 언어로 구현하였다. 전체 알고리즘의 흐름은 크게 세 단계로 구성된다. 첫째, 사용자로부터 아이돌 관련 데이터를 입력받아 초기 상태를 설정한다. 둘째, Knapsack Problem 기반의 점수 평가 방식으로 각 아이돌과 체키 수에 대한 우선순위를 계산한다. 셋째, 반복적인 선택 및 점수 갱신 과정을 통해 제약 조건을 만족하면서도 최대 만족을 달성하는 체키 조합을 도출한다.

본 연구에서 제안하는 알고리즘은 **Static Environment**를 가정한다. 즉, 각 아이돌의 대기 인원 수는 알고리즘 실행 시점에서 고정되어 있으며, 시간 경과에 따른 줄 길이의 변동이나, 팬의 행동 변화, 외부 이벤트와 같은 요소는 고려 대상에서 제외하였다. 이러한 가정은 시뮬레이

선의 복잡성을 줄이고, 특정 조건 하에서 알고리즘의 성능을 정량적으로 비교할 수 있게 해준다.

알고리즘은 다음과 같은 절차에 따라 실행된다:

1. 선호 아이돌 수 n 과 각 아이돌의 정보를 입력받는다.
2. 각 아이돌에 대해 Knapsack 기반의 점수 함수

$$\text{Score} = \alpha \cdot \text{Preference} + \beta \cdot \text{The Number of Chekis}$$

를 사용하여 우선순위를 계산한다.

3. 총 대기 시간이 2시간(7,200초)을 초과하지 않도록 제약 조건을 만족하는 범위 내에서 체크기를 선택한다. 이때 선호도 값이 1인 아이돌은 반드시 최소 1회 이상 포함되어야 한다.
4. 선택된 아이돌 목록과 각 아이돌에 대해 촬영할 체크기 수를 출력한다.

위 구조를 통해 팬의 선호도와 현실적 제약을 동시에 고려한 체계적인 체크기 촬영 전략이 가능해진다.

3.2 아이돌 정보 입력

프로그램 실행 시, 사용자는 먼저 체크기 촬영을 희망하는 아이돌의 수 n 을 입력한다. 이후, 각 아이돌에 대해 다음의 네 가지 정보를 순차적으로 입력한다.

- 이름: 최대 20자 이내 문자열
- 선호도: 0 또는 1로 입력되며, 1은 반드시 체크기를 촬영하고자 하는 멤버를 의미한다
- 대기 인원 수: 해당 멤버 앞에 줄 서 있는 사람의 수
- 희망 체크기 수: 해당 멤버와 촬영하고자 하는 체크기의 총 수

특히, 선호도 값이 1로 설정된 멤버는 필수 선택 대상으로 간주된다. 만약 선호도 1인 멤버가 많아 이들만으로도 전체 특전회 시간(2시간)을 초과할 경우, 입력 순서에 따라 우선순위를 부여하여 일부 멤버만 선택되도록 한다. 이 조건은 현실적인 시간 제약과 팬의 우선 관심도를 동시에 반영하기 위한 규칙이다.

3.3 Knapsack 기반 아이돌 선택

본 연구에서는 Bounded Knapsack Problem의 개념을 응용하여, 제한된 시간 내에 팬이 선호하는 멤버들과의 체크기 촬영 횟수를 극대화하는 선택 전략을 수립한다. 각 멤버는 하나의 항목으로 간주되며, 체크기 촬영을 위한 대기 시간이 무게, 팬의 선호도는 가치로 대응된다.

각 아이돌 i 에 대해, 다음과 같은 점수 함수를 정의한다:

$$\text{Score}_i = \alpha \cdot \text{Preference}_i + \beta \cdot \text{Cheki Count}_i$$

여기서 Preference_i 는 아이돌 i 에 대한 선호도(0 또는 1), Cheki Count_i 는 해당 아이돌과 촬영하고자 하는 체크기 수, α, β 는 점수에 영향을 주는 가중치 파라미터이다.

본 연구에서는 시뮬레이션 기반 파라미터 탐색을 통해 $\alpha = 0.5, \beta = 0.1$ 로 설정하였다. 이 점수 함수는 선호도 기반 선택의 직관성과, 체크기 수에 대한 우선순위를 동시에 반영한다. 점수 계산 후, 가장 높은 점수를 가진 아이돌을 선택하고 해당 아이돌의 체크기 수를 1 감소시킨다. 이 과정을 반복하면서 전체 대기 시간이 2시간(7,200초)을 초과하지 않는 범위 내에서 가능한 많은 체크기를 선택하는 것이 목표이다.

본 단계에서는 계산 효율성을 높이기 위해 **Greedy 방식의 휴리스틱 알고리즘**을 사용하였다. 이 방식은 매 단계마다 국소적으로 최적의 선택을 반복하여 전체 해를 구성하며, Bounded Knapsack과 같이 탐색 공간이 크고 실시간성이 요구되는 문제에서 실용적인 접근으로 널리 활용된다. 본 알고리즘의 의사코드는 Algorithm 1에 제시되어 있다.

3.3.1 우선순위 가중치 파라미터 탐색

본 연구에서는 우선순위 점수 함수의 가중치 파라미터 α, β 값을 최적화하기 위해 MATLAB을 활용한 시뮬레이션을 수행하였다. 시뮬레이션은 다음과 같은 절차로 진행되었다.

1. 선호도 w , 체크기 개수 n 에 대해 $w \in \{0, 1\}$, $n \in \{1, 2, \dots, 5\}$ 를 만족하는 순서쌍 (w, n) 을 생성한다.

Algorithm 1 Greedy Knapsack-based Cheki Selection

```

1: Initialize remaining_time  $\leftarrow$  7200 seconds
2: for each idol  $i$  do
3:   if  $preference_i = 1$  and  $cheki\_count_i > 0$  then
4:      $queue\_time \leftarrow (queue\_length_i + 1) \times 90$ 

5:   if  $remaining\_time \geq queue\_time$  then
6:     Select one cheki for idol  $i$ 
7:      $remaining\_time \leftarrow remaining\_time - queue\_time$ 

8:      $cheki\_count_i \leftarrow cheki\_count_i - 1$ 
9:   while true do
10:     $best\_score \leftarrow 0, best\_index \leftarrow -1$ 
11:    for each idol  $i$  do
12:       $queue\_time \leftarrow (queue\_length_i + 1) \times 90$ 

13:      if  $cheki\_count_i > 0$  and  $remaining\_time \geq queue\_time$  then
14:         $score \leftarrow 0.5 \times preference_i + 0.1 \times cheki\_count_i$ 
15:        if  $score > best\_score$  then
16:           $best\_score \leftarrow score$ 
17:           $best\_index \leftarrow i$ 
18:      if  $best\_index = -1$  then
19:        break
20:      Select one more cheki for idol  $best\_index$ 

21:     $remaining\_time \leftarrow remaining\_time - queue\_time$ 
22:     $cheki\_count_{best\_index} \leftarrow cheki\_count_{best\_index} - 1$ 

```

2. α β 는 각각 0부터 10까지 0.1 단위로 증가시키며, 각 조합마다 점수

$$Score = \alpha \cdot w + \beta \cdot n$$

를 계산한 뒤, 해당 점수에 따라 순서쌍을 정렬한다.

3. 정렬 결과에서 선호도(w)가 1인 순서쌍과 체키 개수(n)가 큰 순서쌍이 상위에 위치할수록 높은 점수를 부여한다.
4. 모든 조합에 대한 평균 순위 점수를 계산하고, 가장 높은 평균 점수를 기록한 α, β 조합을 후보군으로 선정한다.

위와 같은 방식으로 선호도 중심성과 체키 개수 반영 정도 사이의 균형을 평가할 수 있으며, 이를 통해 실용적이고 일관성 있는 우선순위 점수를 부여하는 파라미터 조합을 찾을 수 있다.

Figure 1는 α 와 β 값을 변화시켰을 때의 평균 점수 분포를 시각화한 결과를 나타낸다. 시뮬레이션 결과, 여러 조합이 유사한 성능을 보였으나, 그 중에서도 $\alpha + \beta$ 의 합이 가장 작은 조합을 최종 파라미터로 채택하였다. 이는 계산 복잡도를 낮추고, 점수 계산식의 단순성과 직관성을 유지하기 위함이다.

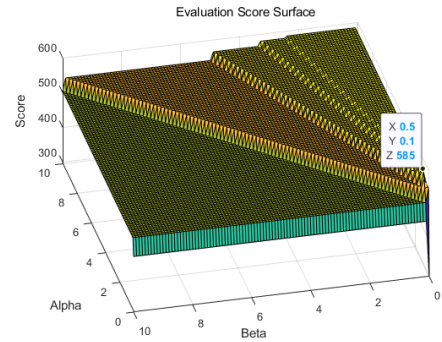


Figure 1: α, β 값 변화에 따른 평균 점수 분포 그래프.

3.4 Priority Scheduling 기반 순서 결정

최종 체키 촬영 순서를 결정하기 위해, 본 연구에서 Priority Scheduling 알고리즘을 응용하였다. 제안된 알고리즘은 세 단계로 구성되며, 각 체키 항목의 우선순위를 계산하고 이를 기반으로 정렬하여 효율적인 순서를 도출한다.

3.4.1 1단계: 스케줄 리스트 구성

먼저 각 아이돌 i 에 대해, 촬영하고자 하는 체크 수만큼 반복하여 스케줄 항목을 생성한다. 각 항목에는 아이돌의 인덱스와 함께 아래의 점수 함수를 기반으로 우선순위 값이 부여된다.

$$\text{Priority} = (\text{Preference} + 1) \times 50 + \text{Cheki Count} \times 1.5 - \frac{\text{Queue Length}}{5}$$

이 점수 함수는 다음 세 가지 요소를 통합하여 우선순위를 산정한다:

- **Preference:** 팬의 선호도를 반영하며, 우선순위에 가장 높은 가중치를 가진다.
- **Cheki Count:** 해당 멤버와 촬영하고자 하는 체크 수가 많을수록 우선순위가 상승한다.
- **Queue Length:** 대기열에 있는 인원이 많을수록 대기 시간이 길어지므로, 감점 요인으로 작용한다.

3.4.2 2단계: 우선순위 정렬

생성된 스케줄 리스트를 우선순위 값에 따라 내림차순으로 정렬한다. 정렬 알고리즘으로는 구현의 단순성을 고려하여 Bubble Sort를 사용하였다. 이 정렬을 통해 높은 우선순위를 가진 체크 항목이 리스트의 앞쪽에 배치되며, 이후 단계에서 촬영 순서에 직접적으로 반영된다.

3.4.3 3단계: 최종 출력 및 분석

정렬이 완료된 후, 리스트의 각 항목을 순서대로 출력한다. 출력 항목에는 체크 순번, 아이돌 이름, 우선순위 점수, 예상 대기 시간이 포함된다. 이를 통해 사용자는 제한된 시간 내에 어떤 아이돌과 어떤 순서로 체크를 촬영하는 것이 가장 효율적인지를 직관적으로 파악할 수 있다.

위 알고리즘은 선호도, 체크 수, 대기 시간이라는 세 가지 요소를 동시에 고려하면서도 계산량이 적어 실시간 의사결정 도구로 활용 가능하다. 계산된 우선순위에 따라 정렬된 체크 순서를 생성하는 과정은 Algorithm 2에 제시되어 있다.

Algorithm 2 Priority-based Cheki Scheduling

- 1: Initialize schedule list as empty
- 2: **for** each idol i **do**
- 3: **for** each selected cheki for idol i **do**
- 4: $priority \leftarrow (preference_i + 1) \times 50 + cheki_count_i \times 1.5 - \frac{queue_length_i}{5}$
- 5: Add $(i, priority)$ to schedule list
- 6: $cheki_count_i \leftarrow cheki_count_i - 1$
- 7: Sort schedule list in descending order of priority
- 8: **for** each entry $(i, priority)$ in sorted schedule **do**
- 9: Print idol i with priority and queue time

4 Results

본 연구에서는 지하아이돌 특전에서 팬이 제한된 시간 안에 최대한 많은 체크를 촬영할 수 있도록, 멤버 선택과 대기 순서를 최적화하는 알고리즘을 제안하였다. 전체 알고리즘은 두 단계로 구성된다. 첫째는 Knapsack Problem을 응용한 멤버 선택, 둘째는 Priority Scheduling 기반의 촬영 순서 결정이다. 알고리즘은 C 언어로 구현되었으며, 각 단계에서 사용되는 가중치 파라미터의 최적값은 MATLAB 기반 시뮬레이션을 통해 사전에 탐색되었다.

Knapsack 단계에서는 체크 촬영에 소요되는 총 시간(대기열 길이 포함)과 선호도를 기반으로 멤버를 선택한다. 이때 각 멤버의 점수는 다음의 식을 통해 계산된다:

$$\text{Score}_i = \alpha \cdot \text{Preference}_i + \beta \cdot \text{Remaining Cheki}_i$$

여기서 Preference_i 는 아이돌 i 에 대한 선호도(0 또는 1), Remaining Cheki_i 는 촬영하고자 하는 체크 수를 의미한다. 점수가 높은 멤버부터 순차적으로 선택되며, 시간 제한(총 7,200초, 2시간)을 넘지 않는 범위 내에서 가능한 많은 체크가 선택되도록 **Greedy 방식**으로 반복 수행된다.

이 알고리즘에서 사용하는 가중치 파라미터 α 와 β 는 결과에 큰 영향을 미치므로, 다양한 조합에 대한 MATLAB 시뮬레이션을 수행하였다. 그 결과 $\alpha = 0.5$, $\beta = 0.1$ 일 때 전체 촬영 가능한 체크 수와 누적 선호도 점수가 모두 최대가 되는 조합임을 확인하였다. 해당 조합은 선택의 직관성과 성능 간의 균형이 가장 우수한 것으로

나타났다.

멤버가 선택된 이후에는 Priority Scheduling 알고리즘을 적용하여 각 체크의 촬영 순서를 결정한다. 각 체크 항목의 우선순위는 다음과 같이 정의된다:

$$\text{Priority} = (\text{Preference} + 1) \times 50 \\ + \text{Cheki Count} \times 1.5 - \frac{\text{Queue Length}}{5}$$

이 점수 함수는 선호도가 높고 촬영해야 할 체크가 많은 멤버를 우선시하는 동시에, 대기열이 긴 멤버는 후순위로 배치되도록 설계되었다. 결과적으로, 각 체크 항목은 계산된 우선순위에 따라 내림차순으로 정렬되며, 전체 평균 대기 시간이 최소화되는 순서를 도출하게 된다. 이러한 방식은 팬의 선호도, 체크 수요, 대기 시간이라는 세 가지 핵심 요소를 통합적으로 고려하여, 현실적인 특전회 현장에서 실질적인 의사결정 지원 도구로 활용될 수 있는 가능성을 보여준다.

5 Discussion

실험 결과는 제안한 알고리즘이 지하아이돌 특전회와 같은 제한된 시간 내 의사결정이 요구되는 환경에서 팬의 인터랙션 기회를 효과적으로 극대화할 수 있음을 보여주었다. 특히 점수 함수 내의 가중치 파라미터 α , β 를 조절함으로써, 팬의 선호도와 대기열 부담 간의 균형을 유연하게 설계할 수 있는 가능성이 확인되었다.

시뮬레이션 결과, α 값이 지나치게 클 경우, 알고리즘은 선호도가 높은 특정 멤버에만 체크 선택을 집중하는 경향을 보였으며, 이로 인해 전체 체크 수가 감소하는 현상이 나타났다. 반면 β 값이 과도하게 커질 경우, 알고리즘은 대기열이 짧은 멤버 위주로 선택을 변경하게 되어, 결과적으로 선호도가 높은 멤버의 우선순위가 낮아지고 팬 만족도가 저하되는 양상이 관찰되었다. 이러한 결과는 알고리즘이 단순히 선호도 기반 선택에만 의존할 것이 아니라, 대기열 정보와의 상호작용을 함께 고려하여 통합적인 판단을 내릴 수 있어야 한다는 점을 시사한다. 즉, 함수의 설계에 있어 α 와 β 의 적절한 조합은 팬 만족도와 체크 효율 간 균형을 달성하기 위한 핵심 요소이며, 이를 통해 실시간 전략 설계에서도 유연하게 적용 가능한 틀을 제공할 수 있다.

6 Conclusion

본 연구는 지하아이돌 특전회라는 시간 제약적이고 복잡한 의사결정 환경을 수학적으로 모델링하고, 그에 적합한 대기 전략 알고리즘을 설계하였다는 점에서 의의를 갖는다. Knapsack Problem과 Priority Scheduling 개념을 결합한 휴리스틱 기반 알고리즘은 팬의 선호도, 대기열 길이, 체크 개수, 시간 제한 등 현실적인 변수들을 통합적으로 고려하여 팬의 만족도를 극대화할 수 있는 전략을 제시하였다.

제안된 알고리즘은 각 아이돌 앞의 대기 인원을 사전에 정확히 알고 있다는 정적 가정 하에 작동하며, 시간 경과에 따른 줄 길이의 동적 변화나 외부 변수의 영향을 반영하지 않았다. 이로 인해 실제 운영 환경과의 괴리가 발생할 가능성도 존재한다. 따라서 향후 연구에서는 이와 같은 보다 현실적인 조건을 반영한 동적 모델링 및 알고리즘 확장이 요구된다.

7 References

1. Abraham Silberschatz, *Operating System Concepts 10th Edition*, Wiley, 2018.