

비디오 스트리밍 서버에서의 유전 알고리즘을 활용한 사용자 인기도 기반 데이터 배치 기법

Popularity-aware data placement techniques exploiting genetic algorithm in video streaming servers

이다영, 송민석¹⁾

Dayoung Lee, Minseok Song

(22212) 인천광역시 미추홀구 인하로 100 인하대학교 컴퓨터공학과
cecci08@naver.com, mssong@inha.ac.kr

요 약

동적 적응 스트리밍 (Dynamic Adaptive Streaming over HTTP (DASH)) 서버는 트랜스코딩된 다양한 비트율의 비디오 세그먼트를 저장해야 하기 때문에, 스토리지 요구량을 줄이는 것이 매우 중요하다. 본 논문에서는 비디오 스트리밍 서버에서의 유전 알고리즘을 활용한 사용자 인기도 기반 데이터 배치 기법을 제안한다. 우리는 트랜스코딩된 다양한 버전의 비디오 세그먼트들 중에서 어떤 버전을 저장하고, 어떻게 세그먼트를 디스크들에 분산시킬 지에 대한 문제를 정의하고, 유전 알고리즘을 이용한 시뮬레이션 모델을 개발하였다. 이를 위해서 유전 알고리즘의 개체군의 크기, 선택 비율, 교배 비율, 돌연변이 비율을 설정하기 위한 실험을 진행하였으며, 유전 알고리즘의 결과와 최적값을 비교하여 본 기법의 신뢰성을 확인했다.

Abstract

It is important to reduce the growing storage requirements of dynamic adaptive streaming over HTTP(DASH) servers that need to store transcoded video segments of various bit rates. In this paper, we propose a new genetic algorithm techniques for popularity-aware data placement in video streaming servers. We defined the popularity-aware data placement problem that chooses and places video segments on each disk. We designed and implemented a simulation model to solve this problem using a genetic algorithm. Experiments were carried out to set the population size, selection rate, crossover rate and mutation rate. We compared the results of the genetic algorithm with the optimum value to confirm the reliability of this technique.

※ 이 논문은 2018년도 과학기술정보통신부의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임 (NRF-2017M3C4A7080248).

1) 교신저자

키워드: 유전 알고리즘, 비디오 스트리밍 서비스, 사용자 인기도 기반 데이터 배치

Keyword: Genetic algorithm, Video streaming service, popularity-aware data placement

1. 서론

최근 인터넷 및 스마트 기기가 대중화되며 인터넷을 통해 콘텐츠를 제공하는 서비스가 늘어났다. 특히 기술의 발전과 함께 Youtube, Netflix, Twitch 등의 비디오 스트리밍 서비스는 우리 생활의 일부분으로 자리 잡았다.

Youtube, Netflix, Twitch 등과 같은 우수한 스트리밍 업체들은 동적 적응 스트리밍(DASH, Dynamic Adaptive Streaming over HTTP)을 일반적으로 채택한다[4,12]. 동적 적응 스트리밍은 비디오를 여러 개의 비디오 세그먼트들로 분할한 후, 해당 세그먼트를 여러 버전의 비트율로 트랜스코딩한다. 이후 사용자의 네트워크 환경에 맞추어 가장 적합한 비트율인 세그먼트를 제공하는 방법이다.

이를 위해서는 다양한 비트율의 세그먼트들이 필요하다. Netflix에 의하면 총 120번의 트랜스코딩 연산이 1개의 비디오에 대해서 필요한 것으로 알려져 있으며, 이를 저장하기 위한 스토리지 요구량도 급증하고 있다[14]. 그러나 각 콘텐츠의 인기도는 천차만별로 차이가 있기 때문에 모든 콘텐츠에 대하여 모든 비트율 버전들을 제공할 필요가 없으며, 이를 잘 활용할 필요가 있다.

최근에 인기도와 각 비트율 버전의 비디오 품질을 고려하여 선택적으로 트랜스코딩하여 저장하는 기법에 대해서 연구가 진행되었다[1,3,4]. 특정 디스크에 요청이 집중되면, 디스크 대역폭을 효과적으로 활용할 수 없으므로, 이용률이 저하되며, 사용자의 접근 패턴을 고려한 디스크 배치 방법이 필수적이다. 그러나 기존 연구에서는 비디오 품질을 최대화하는 데 초점을 두었으며, 데이터의 배치에 대한 고려가 미흡하다.

인기도를 고려한 데이터 배치는 다음의 2개의 사항을 고려해야 한다. 첫째, 가능한 요청 확률이 높은 비디오 세그먼트를 디스크에 저장해야 한다.

둘째, 요청 확률이 각 디스크에 균등하게 분배되어야 한다. 본 논문에서는 이러한 문제를 해결하기 위해 DASH 서버에서 유전 알고리즘 기법을 활용한 사용자 인기도를 고려한 데이터 배치 기법을 제안한다.

2장에서는 기존의 비디오 스트리밍 관련 연구 및 유전 알고리즘을 사용한 최적화 연구를 설명하고, 3장에서는 문제를 정의하며, 4장에서는 제안한 유전 알고리즘을 통한 최적화 기법에 대해 설명한다. 5장에서는 시뮬레이션 환경을 정의하고, 6장에서는 실험을 통해 최적의 매개변수를 찾아낸다. 7장에서는 6장에서 찾은 매개변수를 사용한 시뮬레이션을 통해 우리가 제안하는 데이터 배치 기법이 3장에서 정의한 문제를 해결하는지 확인하고, 사용한 유전 알고리즘과 최적값을 비교하여 설계한 적합도 함수의 신뢰성을 평가하며, 마지막 8장에서 결론을 제시하고 본 논문을 마친다.

2. 관련 연구

2.1. 기존의 비디오 스트리밍을 위한 관련 연구

Wang과 그 외 연구진은 측정 연구를 바탕으로 최적화 문제를 공식화하여 사용자에게 높은 스트리밍 품질을 제공하고, 시스템에게 적은 계산과 비용을 제공하는 트랜스코딩 기법을 제안했다[1]. Toni와 그 외 연구진은 사용자 기반 특성, 네트워크 용량 및 비디오 콘텐츠 유형을 기반으로 한 인코딩 매개 변수 선택을 위한 가이드라인을 세웠다[4]. Zhao와 그 외 연구진은 비디오 내부 인기도에 따라 비디오를 세그먼트로 분할하고, 여러 버전의 트랜스코딩 기반의 비디오를 제공하여 용량 및 CPU 자원의 비용을 최소화하는 VoD시스템을 제안했다[3]. Aparicio-Pardo와 그 외 연구진은 라이브 비디오 스트리밍 서비스를 위해 해상도 및 비트 전송률에 대한 분석을 바탕으로 QoE의 평균값을 극대

화하는 알고리즘을 제안했으며[5], Barlas는 Cluster 기반의 데이터 파티셔닝 및 스케줄링을 통한 병렬 비디오 트랜스코딩 최적화 기법을 통해 각 다른 기종의 플랫폼에서 CBR 및 VBR 미디어 모두에게 최적화된 트랜스코딩을 할 수 있도록 했다[2].

김형식과 그 외 연구진은 무분별한 트랜스코딩을 막기 위해, 명암도 기반의 비디오 화질 측정 기법을 [17], 백치선과 그 외 연구진은 트랜스코딩이 발생하는 서버에서의 부하량을 예측하여, 사용자가 인식할 수 있는 정보 손실을 최소화 하는 기법을 제안했다[18].

기존 연구에서는 디스크 용량에 대한 고려가 미흡하였다. 최근 Han과 그 외 연구진은 서버 및 QoE를 고려한 동적 적응 알고리즘을 제안했다. 해당 알고리즘은 휴리스틱으로, Hot, Cold 데이터와 디스크를 선정하고 각 데이터를 디스크에 분산시킨다[11]. 그러나, 제안한 기법은 비디오 세그먼트 선택 문제와 배치 문제를 별도로 다루었다.

2.2 기존의 유전 알고리즘을 이용한 최적화 연구

Kaur와 그 외 연구진은 클라우드 서비스의 작업 예약 시, 유전 알고리즘을 이용하여 실행 시간과 실행 비용을 최소화하는 스케줄링을 제안했다[15].

류기동과 그 외 연구진은 유전 알고리즘을 이용하여 담당자 업무에 대한 부하 균등, 할당된 사이트의 이동거리 균등 및 기술스펙 일치성 등을 확인하여 IT 서비스 담당자의 권역을 할당 최적화 모델을 제안했다[6].

Qiu와 그 외 연구진은 그린 클라우드의 PCM 메모리를 탑재한 멀티프로세서를 위한 최적화 알고리즘을 제안하고[7], Dasgupta와 그 외 연구진은 클라우딩 컴퓨팅을 위해 유전 알고리즘을 사용하여 변화하는 환경과 작업 유형에 맞게 동적으로 로드 밸런싱하고 시뮬레이션 하였으며[8], Portaluri와 그 외 연구진은 유전 알고리즘을 사용해 클라우드 컴퓨팅 데이터 센터의 연산 및 네트워킹 요구 사항을 고려하여 자원을 최적화했다[9].

윤경목은 유전 알고리즘을 통해 얻은 특징

(feature)를 신경망을 이용한 영화 추천 시스템의 기본 데이터로 사용하였다[16].

상기의 연구들은 여러 방법으로 유전 알고리즘을 통해 자원을 최적화 한다. 그러나 비디오 스트리밍 서버에서의 상황이 고려되지 않았다. Cheng과 그 외 연구진은 유전 알고리즘을 통해 End to End 지연 시간 범위 내에 LTE 환경에서 스트리밍 되는 비디오의 다운로드 스케줄링을 시행하는 기법을 제안했다[10]. 그러나, 동적 적응 스트리밍 서버에서의 파일 배치 문제에 대한 연구는 없는 것으로 파악된다.

3. 문제 정의

본 연구진은 동적 적응 스트리밍을 지원하는 디스크 배열 기반 스트리밍 서버를 위한 데이터 배치 기법을 제안한다.

동적 적응 스트리밍에서는 최상위 비트율 비디오를 다양한 비트율의 여러 버전으로 트랜스코딩하고, 고객의 비트율 요청에 따라서 서비스를 제공한다. 이 때 각 비디오 세그먼트의 모든 트랜스코딩 버전을 저장한다고 가정했을 때, 많은 용량의 스토리지를 요구한다.

그러나 비디오의 인기도는 긴꼬리 분포 (long-tail distribution)을 따르므로, 인기 없는 세그먼트의 여러 버전을 동시에 저장하는 것은 스토리지 용량을 크게 증가시킬 수 있다. 또한 특정 디스크에만 사용자의 접근이 편중되는 것은 디스크 대역폭을 비효율적으로 관리하게 되며, 모든 디스크에 균등하게 사용자 요청을 분배할 필요가 있다.

위의 2가지 요구조건을 정리하면 다음과 같다.

- ① 스토리지 공간이 제한적일 때 가능한 요청 확률이 높은 비디오 세그먼트를 디스크에 저장한다.
- ② 요청 확률이 각 디스크에 균등하게 분배되어야 한다.

본 장에서는 상기 2가지 요구조건을 동시에 만족하기 위해, 각 비디오의 트랜스코딩 세그먼트들 중 무엇을 저장할 지 선택하는 문제를 정의한다.

제안하는 문제는 비디오 세그먼트 $V_{i,j} (i = 1, \dots, N^{seg}, j = 1, \dots, N^{ver})$ 에 대한 접근률을 고려하여 저장할 세그먼트를 결정하고, 저장될 디스크의 위치를 결정한다. 여기서 N^{seg} 은 비디오 세그먼트의 총 개수, N^{ver} 을 각 비트율 버전의 개수이다.

이 때 원래 버전인 최대 비트율 버전 $V_{i,1}$ 은 다른 비트율로 트랜스코딩하기 위해서 저장되어야 하며, 최소 비트율 버전 $V_{i,N^{ver}}$ 은 요청한 비트율 버전이 없을 때 낮은 품질의 비디오로 서비스하기 위해서 인기도와 상관없이 반드시 저장되어야 한다.

우리는 문제의 해결을 위해 각 비디오 세그먼트 $V_{i,j} (i = 1, \dots, N^{seg}$ 및 $j = 2, \dots, N^{ver} - 1)$ 에 대한 $X_{i,j}$ 와 $W_{i,j}$ 변수를 다음과 같이 정의한다.

첫째, $X_{i,j} (i = 1, \dots, N^{seg}$ 및 $j = 1, \dots, N^{ver}, X_{i,j} \in \{0,1\})$ 는 $V_{i,j}$ 의 저장 여부를 의미한다. 만약 $X_{i,j}$ 가 1이면, 비디오 세그먼트 $V_{i,j}$ 는 디스크에 저장됨을 의미한다. 반면에 $X_{i,j}$ 가 0이면 비디오 세그먼트 $V_{i,j}$ 는 트랜스코딩 되지 않으며, 다시 말해서 디스크에 저장되지 않는다. 그러나 상기에서 언급했 듯 최소 비트율과 최대 비트율 버전은 반드시 저장되어야 하므로 다음의 등식이 성립한다. 이 때 N^{disk} 을 서버를 이루는 디스크의 개수이다.

$$X_{i,1} = 1 \text{ and } X_{i,N^{ver}} = 1 \quad (1)$$

$W_{i,j}$ 는 $X_{i,j}$ 가 1일 경우, 비디오 세그먼트 $V_{i,j}$ 가 저장된 디스크를 의미한다. 만약 $X_{i,j}$ 가 0일 경우, $W_{i,j}$ 도 0이며 이는 어떤 디스크에도 해당 비디오 세그먼트가 저장되지 않는 것을 의미한다.

$$W_{i,j} \in \{0, \dots, N^{disk}\} \quad (2)$$

$p_{i,j}$ 는 각 비디오 세그먼트 $V_{i,j}$ 에 대한 접근률이다. $p_k^{disk} (k = 1, \dots, N^{disk})$ 는 각 디스크에 저장된 세그먼트들의 접근률에 대한 총합이고, \bar{p} 는

각 디스크 마다 저장된 세그먼트들의 접근률의 평균값이다. 각각은 다음과 같이 계산된다.

$$p_k^{disk} = \sum_{\forall V_{i,j} \text{ where } W_{i,j} = k} p_{i,j} \quad (3)$$

$$\bar{p} = \frac{\sum_{k=1}^{N^{disk}} p_k^{disk}}{N^{disk}} \quad (4)$$

$S_{i,j}$ 는 $V_{i,j}$ 의 파일 크기이고, C_k^{disk} 는 각 디스크의 사용 가능한 공간의 용량이다. 제안하는 데이터 배치 문제의 목적은 다음과 같이 요약된다.

① 서버에 저장될 세그먼트의 접근률 합

$$\left(\sum_{i=1}^{N^{seg}} \sum_{j=2}^{N^{ver}-1} X_{i,j} p_{i,j} \right) \text{을 최소화한다.}$$

② 위의 식을 만족시키면서, 요청 확률이 각 디스크에 균등하게 분배되도록 하기 위해, 각 디스크

크 접근률의 표준편차 $\left(\sqrt{\frac{\sum_{k=1}^{N^{disk}} (p_k^{disk} - \bar{p})^2}{N^{disk}}} \right)$ 를 최소화한다.

③ 서버의 각 디스크에 저장된 세그먼트들의 파일 크기 합은 해당 세그먼트가 저장된 각 디스크의 사용 가능한 용량을 초과 할 수 없다.

$$\left(\text{if } W_{i,j} = k, \sum_{i=1}^{N^{seg}} \sum_{j=1}^{N^{ver}} X_{i,j} S_{i,j} \leq C_k^{disk} \right)$$

상기 조건을 만족하는 $X_{i,j}$ 와 $W_{i,j}$ 를 구하는 최적화 문제를 식으로 표현하면 아래와 같다.

$$\text{Maximize } \sum_{i=1}^{N^{seg}} \sum_{j=2}^{N^{ver}-1} X_{i,j} p_{i,j} \quad (5)$$

$$\text{Minimize } \sqrt{\frac{\sum_{k=1}^{N^{disk}} (p_k^{disk} - \bar{p})^2}{N^{disk}}} \quad (6)$$

$$\text{Subject to if } W_{i,j} = k, \sum_{i=1}^{N^{seg}} \sum_{j=1}^{N^{ver}} X_{i,j} S_{i,j} \leq C_k^{disk} \quad (7)$$

상기 문제는 (식 6)을 제외 시 다중 배낭 문제(Multiple knapsack problem)에 해당된다[13]. 다중 배낭 문제는 여러 개의 배낭의 무게 한도 내에서, 배낭에 담을 물건의 이득을 최대화하는 것이다. 제안하는 데이터 배치 문제에서는 서버를 이루는 여러 개의 디스크 용량 한도 내에서, 접근률의 총합을 최대화하는 것이므로, 조건을 하나 더 추가한 문제이다. 모든 knapsack 문제는 NP-hard이므로 [13], 해당 문제는 NP-hard이다.

4. 유전 알고리즘의 설계

4.1. 기본 개념

유전 알고리즘이란 진화를 일으키는 유전 과정을 본 뜬 알고리즘으로, 두 개의 부모해의 형질을 조합하여 하나의 자손을 만든다. 첫 번째 세대는 개체군(Population)의 개수만큼 랜덤으로 염색체를 생성하여 채우는 방식으로 생성한다. 이렇게 만들어진 개체군을 교배하고, 돌연변이를 생성시켜 우수한 유전자를 선택하여 다음 세대를 만든다[6,15].

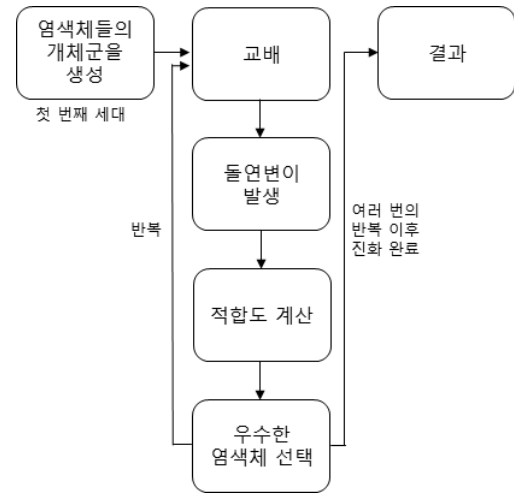
그러나 일반적인 유전 알고리즘은 초기 해가 생성된 후, 그 중 우수한 해를 선정하여 교배 및 돌연변이 생성을 진행하여 다음 세대를 만들었으나, 이렇게 하면 서버의 각 디스크의 용량을 넘지 않는 해를 생성하기 쉽지 않아 적합도가 0이 되어버리는 경우가 많았다. 이를 해결하기 위해 초기해 중 일정 비율에 한하여, 최대, 최저 비트율인 버전만 저장하도록 생성하였다. 이는 이후 4.4절에서 자세하게 후술할 것이다.

이렇게 생성한 초기해를 교배 및 돌연변이 생성 후 우수한 해를 선정하여 다음 세대를 제작하도록 설계했다.

유전 알고리즘은 이렇게 세대를 거칠수록 자손들은 진화하여 적합도는 더욱 높아진다는 전제를 두고 있다. 본 연구에서는 3장의 문제를 해결하기 위

한 유전 알고리즘을 설계하였다. 이는 Java기반의 JGAP 라이브러리를 사용하여 구현되었다.

우리는 유전 알고리즘의 염색체(chromosome)를 저장할 비디오 세그먼트의 집합에, 염색체를 이루는(gene)을 해당 세그먼트의 저장여부 및 저장될 디스크에 대응시킴으로써 각 해를 표현하였으며, 교배된 해가 알고리즘의 목적에 맞는지 적합도 함수를 통해 평가하였다.



(그림 1) 설계된 유전 알고리즘의 흐름도

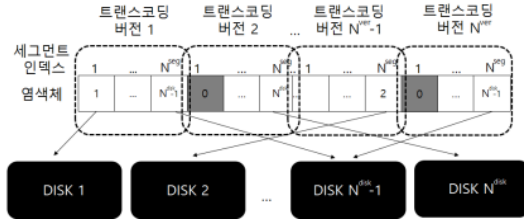
4.2. 해 표현

제시한 문제를 해결하기 위해서는 어떤 세그먼트를 저장하고 저장하지 않을 것인가, 만약 저장한다면 어떤 디스크에 저장할 것인가를 결정해야 한다. 디스크가 N^{disk} 개일 경우, 디스크의 인덱스는 $1 \sim N^{disk}$ 인 정수로 정의하며, 이는 유전자의 값이 된다. 염색체는 비디오의 세그먼트의 수에 각 버전의 수를 곱한 만큼의 유전자로 이루어져 있으며, 유전자가 대응하는 세그먼트가 무엇인지는 (그림 2)에서 확인할 수 있다.

유전자는 대응되는 세그먼트를 저장하지 않거나, 저장한다면 어떤 디스크에 저장하는 지를 정의한다. 유전자의 값이 0의 경우 저장하지 않으며, 1부터 N^{disk} 사이라면 해당 유전자가 나타내는 세그먼트

를, 유전자의 값이 나타내는 디스크에 저장한다. 즉 0부터 N^{disk} 까지의 숫자를 값으로 갖는 정수형 유전자(Integer Gene)이다.

아래 (그림 2)는 실제로 각 염색체가 어떻게 만들어지는지를 보여준다. 염색체의 유전자에 맞춰 세그먼트들이 각각의 디스크에 저장되거나, 값이 0일 경우 저장되지 않는 것을 확인할 수 있다.



(그림 2) 해 표현

4.3. 적합도 함수

$X_{i,j}$ 와 $W_{i,j}$ 를 구하는 최적화 문제를 해결할 수 있는 적합도 함수를 만들기 위해, 정의한 목적의 달성률을 나타내고, 반드시 양의 값을 가지도록 하도록 했다. 아래의 (식 8)은 (식 5)의 달성률, (식 9)는 (식 6)의 달성률을 수식화하였다.

$$F_1 = \left(\sum_{i=1}^{N^{seg}} \sum_{j=2}^{N^{ver}-1} \frac{X_{i,j} p_{i,j}}{p_{i,j}} \right) * 100 \quad (8)$$

$$F_2 = \left(1 - \frac{\sqrt{\frac{\sum_{k=1}^{N^{disk}} (p_k^{disk} - \bar{p})^2}{N^{disk}}}}{\bar{p}} \right) * 100 \quad (9)$$

적합도는 F_1 과 F_2 의 합을 통해 얻으며, 최종적으로 이 합이 가장 높은 해를 구하도록 유전 알고리즘을 구현하였다. 적합도는 반드시 양수이다. 이 때, F_1 과 F_2 중 어떤 하나만 높게 달성하는 것을 막기 위해 F_1 과 F_2 의 차의 절대값을 벌점으로 주어, F_1 과 F_2 의 합에서 빼도록 했다.

단 (식 7)과 맞지 않을 경우, 무조건 적합도 값은 0이 된다. 따라서 적합도 함수는 아래 식으로 표현된다.

$$fitness = F_1 + F_2 - |F_1 - F_2| \quad (10)$$

4.4 초기해 생성

초기해는 랜덤으로 생성된다. 트랜스코딩 버전이 최대 혹은 최저 비트율 버전일 경우 반드시 저장해야하므로 디스크의 인덱스인 1부터 N^{ver} 까지의 정수를 값으로 주고, 그 외의 버전은 세그먼트를 저장하지 않는 경우인 0을 포함하여 0부터 N^{disk} 까지의 정수를 유전자의 값으로 선택할 수 있도록 하였다.

또한, 만약 서버의 사용 가능한 공간의 용량 C^{server} ((식 12) 참조)보다, 서버에 저장된 모든 비디오 세그먼트들의 파일 크기의 합 S^{server} ((식 11) 참조)가 더 클 경우, 모든 초기해의 적합도가 0으로 나와 해를 찾기 어려울 수 있다. 그러므로 초기 해 중 일정 비율 $1-r$ ($r \leq 1.0$)만큼의 해의 경우, 트랜스코딩 버전이 최대, 최저 비트율일 경우만 저장하도록 생성하였다((식 13) 참조).

$$S^{server} = \sum_{i=1}^{N^{seg}} \sum_{j=1}^{N^{ver}} X_{i,j} S_{i,j} \quad (11)$$

$$C^{server} = \sum_{k=1}^{N^{disk}} C_k^{disk} \quad (12)$$

$$\begin{aligned} S^{server} &\leq C^{server}, r = 1.0 \\ S^{server} &> C^{server}, r = \frac{C^{server}}{S^{server}} \end{aligned} \quad (13)$$

4.5. 선택 전략 (Selection)

유전 알고리즘은 여러 세대를 거듭하면서 염색체를 선택하여 교배시킴을 통해 더욱 발전한 해를 만들어내는 알고리즘이므로, 우수한 알고리즘을 판단

하는 선택 전략을 필요로 한다. 본 연구에서는 Elitism을 사용하였으며, Elitism은 적합도가 높은 염색체를 무조건 선택하는 방식이다.

Elitism에 의해 선택된 염색체로 채운 이후 남은 교배 풀의 공간을 채울 염색체는 랜덤으로 선택된다. 또한, 중복된 염색체를 허용하지 않아, 해의 다양성을 유지하도록 했다.

4.6. 교배 (Crossover)

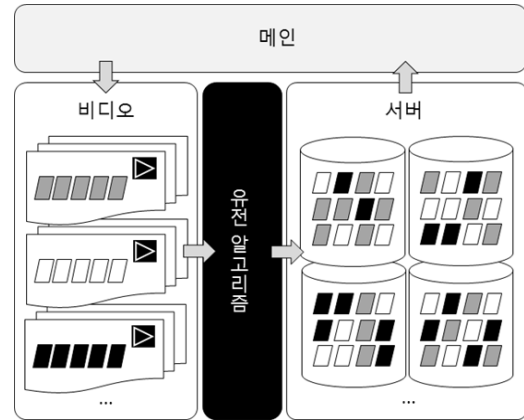
부모 유전체를 어떻게 합성해서 새로운 자손 유전체를 만들지 결정하는 방법이다. 교배에는 Random 1 point 방법을 사용했으며, 이는 첫 번째 부모로부터 염색체의 앞 지점을 받고, 교배 지점부터는 두 번째 부모로부터 염색체의 뒷 지점을 받는 방법이다. 이 때, 부모와 교배 지점은 랜덤하게 결정된다.

4.7. 돌연변이

염색체의 몇몇 유전자가 랜덤 확률로 변이되는 것으로, 유전적 형질의 다양성을 주어, 진화의 결과가 국지 최적해(Local Optimal Solution)로 조기 수렴하는 것을 막기 위해 사용된다. JGAP 라이브러리는 Desire rates라는 값을 입력함으로써 돌연변이의 확률을 결정할 수 있는데, 이 값이 만약 12일 경우, 각 유전자는 1/12의 확률로 돌연변이가 발생하게 된다.

5. 시뮬레이션 환경

본 논문에서 제시하는 데이터 배치 기법은 우리가 개발한 시뮬레이션 모델에 적용되었다(그림 3).



(그림 3) 데이터 배치 시뮬레이션 모델 구조도

해당 시뮬레이션 모델은 서버를 이루는 각 디스크의 용량과, 각각의 모든 비디오 세그먼트의 비트율, 용량, 사용자 접근률을 메인 화면에서 입력받아 데이터 배치 전 시뮬레이션 할 수 있도록 했다.

우리는 설계된 데이터 배치 기법의 성능을 확인하기 위하여 시뮬레이션을 진행하였다. 서버를 구성하는 디스크의 설정은 다음과 같다. 용량이 1TB인 디스크 4개로 구성되어, 총 4TB이다. ($N^{disk} = 4$).

비디오 세그먼트의 길이 L 는 30초로 설정했으며, 비디오 세그먼트의 트랜스코딩 버전은 총 5가지로 구성된다. 즉 $N^{ver} = 5$ 이다. 비디오 세그먼트의 접근률은 zipf 분포를 따르며, $\theta = 0.271$ 이다. 세그먼트 선택 전, 모든 세그먼트들의 접근률 합은 1이다.

각 버전의 비트율은 스트리밍 업체 및 시중 카메라의 원본 촬영 데이터의 비트율을 참고하여 <표 1>과 같이 정의하였다. 즉, 각 비디오 세그먼트의 용량은 모두 다르다.

<표 1> 각 트랜스코딩 버전의 정보

버전	유형	비트율
1	원본 데이터	10000Kbps
2	2160p/48~60fps	66000~85000Kbps
3	2160p/24~30fps	53000~68000Kbps
4	1440p/48~60fps	44000~56000Kbps
5	1440p/24~30fps	35000~45000Kbps

6. 매개 변수 선택 실험

6.1. 매개변수 선택을 위한 실험 환경 설정

유전 알고리즘에는 다양한 매개변수가 필요하며, 이 매개변수에 따라 최종 결과 역시 변화한다. 하지만 이 매개변수의 적합한 값은 실험을 통해서만 확인할 수 있다. 매개변수는 개체군의 크기, 선택 비율, 교배 비율, 돌연변이 비율이 있으며 이들의 최적 값을 찾기 위한 실험을 진행했다.

실험에 사용한 비디오 세그먼트는 총 20000개로, 총 용량은 약 4.678TB이다. 이는 7.2절의 시뮬레이션 1에서 사용한 것과 동일하며 7.2절에서 추가적으로 설명할 것이다.

모든 실험은 5번씩 수행하여 평균값을 내었으며, 비교하려고 하는 매개변수를 제외한 나머지 매개변수를 JGAP의 디폴트 설정으로 고정한 상황에서 세대 수를 100으로 하여 실험하였다.

〈표 2〉 JGAP의 디폴트 설정의 값

개체군	선택 비율	교배 비율	돌연변이 비율
100개	90%	35%	1/12 (8.33%)

실험 결과를 바탕으로 ANOVA 검정을 시행하여 모든 집단의 적합도 평균이 집단의 평균과 같다는 영가설을 세우고, 이를 검증한다.

ANOVA 검정의 영가설이 기각될 경우, 집단 별 적합도 평균이 집단의 평균과 다른 것이므로, 적합도의 평균이 가장 높은 값을 매개변수로 결정했다. 반대로 채택될 경우, JGAP의 디폴트 설정의 값을 매개변수로 결정하였다.

해당 유전 알고리즘은 위에서 정의한 식 중, (식 5)는 최대화하고 (식 6)은 최소화할 때, 한 쪽에 치우치지 않고 둘 다 만족하는 결과를 찾는 데에 목적이 있다. 그러므로 아래의 결과에는 적합도의 평균만이 아니라 해당 식들의 값의 평균 역시 함께 기술했다.

6.2. 개체군

개체군이 100개, 200개, 300개, 400개, 500개의 경우에 대하여 실험했다.

(표 3) 개체군 크기 설정의 테스트 결과

개체군	100개	200개	300개	400개	500개
(식 5)	0.397	0.408	0.419	0.42	0.423
(식 6)	0.009	0.008	0.01	0.01	0.009
적합도	158.845	163.365	167.675	168.139	169.23

저장된 세그먼트의 평균 개수는 16727.2개, 16733.4개, 16854개, 16873.6개, 16945.6개이다. 또한 평균 3.952TB, 3.945TB, 3.944TB, 3.957TB, 3.966TB가 저장되었다.

ANOVA 검정의 결과, F 비는 6.772, F 기각치는 2.866이 나옴으로써 F 비 > F 기각치이다. 즉 영가설이 기각되어 통계적으로 차이가 유의하므로 이 중 가장 적합도 평균이 높은 500개로 개체군의 크기를 선정하였다.

6.3. 선택 비율

선택 비율이 10%, 30%, 50%, 70%, 90%일 경우에 대하여 실험하였다. 이 때 100%일 경우는 제외했다. 이는 해의 다양성을 해쳐 일찍 해가 조기수렴 되는 것을 막기 위함이다.

〈표 4〉 선택 비율 설정의 테스트 결과

선택 비율	10%	30%	50%	70%	90%
(식 5)	0.404	0.398	0.405	0.398	0.397
(식 6)	0.014	0.01	0.014	0.009	0.009
적합도	161.796	159.169	161.831	159.341	158.845

저장된 세그먼트의 평균 개수는 16708.6개, 16608.6개, 16601개, 16753.2개, 16727.2개이며, 평균 3.925TB, 3.907TB, 3.907TB, 3.923TB, 3.952TB가 저장되었다.

ANOVA 검정 결과, F 비는 0.45, F 기각치는

2.866이므로 F 비 < F 기각치이다. 즉 영가설이 채택되어 통계적으로 차이가 유의하지 않으므로, JGAP 디폴트 설정의 값인 90%로 선택 비율을 선정하였다.

6.4. 교배 비율

교배 비율이 10%, 30%, 50%, 70%, 90%일 경우에 대하여 실험하였다. 선택 비율과 마찬가지로 이는 해의 다양성을 해치는 것을 막기 위해 100%인 경우는 제외했다.

〈표 5〉 교배 비율 설정의 테스트 결과

교배비율	10%	30%	50%	70%	90%
(식 5)	0.41	0.408	0.406	0.408	0.405
(식 6)	0.014	0.01	0.01	0.009	0.01
적합도	163.891	163.039	162.561	163.333	161.945

저장된 세그먼트의 평균 개수는 16717.2개, 16717.2개, 16800개, 16686.6개, 16642.8개이며, 평균 3.923TB, 3.926TB, 3.936TB, 3.944TB, 3.916TB가 저장되었다.

ANOVA 검정결과, F 비는 0.431, F 기각치는 2.866이므로 F 비 < F 기각치이다. 즉 영가설이 채택되어 통계적으로 차이가 유의하지 않으므로, JGAP 디폴트 설정의 값인 35%로 선택비율을 선정하였다.

6.5 돌연변이 비율

〈표 6〉과 같이 돌연변이 비율이 1%, 2.5%, 5%, 10%일 경우에 대하여 실험하였다.

〈표 6〉 돌연변이 비율 설정의 테스트 결과

돌연변이율	1%	2.5%	5%	10%
(식 5)	0.424	0.413	0.407	0.398
(식 6)	0.012	0.014	0.012	0.012
적합도	169.583	165.105	163.191	159.371

저장된 세그먼트의 평균 개수는 16997.8개,

16802.8개, 16777.4개, 16662.2개이며, 평균 3.992TB, 3.965TB, 3.937TB, 3.94TB가 저장되었다.

ANOVA 검정의 결과, F 비는 6.618, F 기각치는 3.239이 나옴으로써 F 비 > F 기각치이다. 즉 영가설이 기각되어 통계적으로 차이가 유의하므로 이 중 가장 적합도 평균이 높은 1%로 돌연변이 비율을 선정하였다.

7. 실험 결과

7.1. 유전 알고리즘의 매개 변수

모든 시뮬레이션에서 사용한 유전 알고리즘의 매개변수는 6장의 결과를 결정하였으며 이는 〈표 7〉에서 확인할 수 있다.

〈표 7〉 사용한 유전 알고리즘의 매개변수

개체군	선택 비율	교배 비율	돌연변이 비율
500개	90%	90%	1%

7.2. 시뮬레이션 1

실험에 사용한 세그먼트는 총 20000개로, 세그먼트의 파일 크기는 모두 각기 다르다. 세그먼트들의 총 용량은 약 4.678TB이므로 $S^{server} > C^{server}$ 이다. 이 중 2.062TB는 각 세그먼트를 위한 최대와 최소 비트율을 저장하기 위한 공간이며, 나머지 2.616TB에 대하여 저장 여부를 선택하게 된다. 버전별 인기도는 아래와 같게 설정했다.

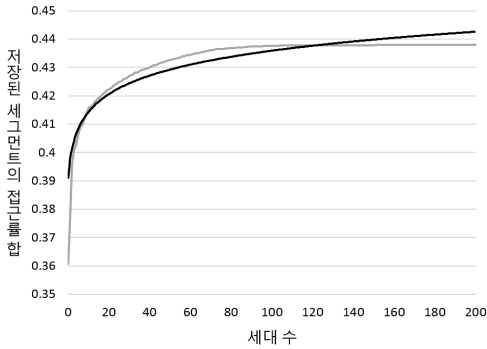
$$\forall i, p_{i,1} = p_i * 0.3, p_{i,2} = p_i * 0.1, p_{i,3} = p_i * 0.1, p_{i,4} = p_i * 0.3, p_{i,5} = p_i * 0.2 \quad (14)$$

트랜스코딩 버전 중 최대, 최소 비트율인 경우를 제외한($j = 2, \dots, N^{ver} - 1$) 세그먼트들의 접근률 합은 0.5이다.

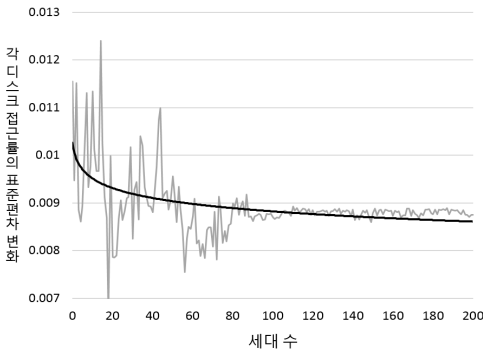
(그림 4,5,6)에 나오는 그래프의 회색 선은 평균 값의 선, 검은 선은 추세선이다.

진화의 세대 수가 늘어날수록 (그림 4)과 (그림

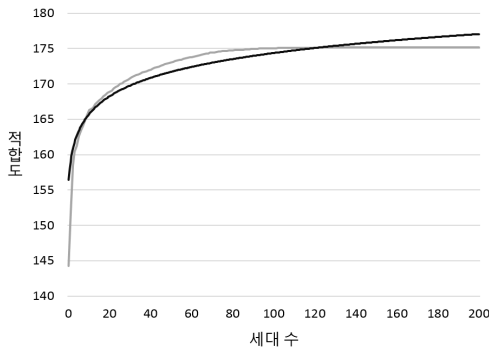
5)를 통해 (식 5)의 결과 값을 증가시키면서 (식 6)을 감소시킴을 확인함으로써 3장에서 정의한 문제가 만족함을 알 수 있다.



(그림 4) 시뮬레이션 1에서 저장된 세그먼트의 접근률 합의 변화



(그림 5) 시뮬레이션 1의 각 디스크 접근률의 표준편차 변화



(그림 6) 시뮬레이션 1의 적합도 변화

시뮬레이션을 시행한 결과, 평균 17076.6개의 세그먼트가 저장되며, 디스크 1은 999.794GB, 디스크 2는 999.854GB, 디스크 3은 999.909GB, 디스크 4는 999.871GB가 저장됨으로써, 총 3.999TB가 저장된다. 저장된 세그먼트들이 (식 7)을 만족시키므로 각 디스크 및 서버의 용량을 넘지 않음을 확인할 수 있다.

해당 실험에서 도출한 최종 결과의 평균값은 (식 5)는 0.438, (식 6)은 0.009이며, 평균 적합도는 175.184이다.

7.2. 시뮬레이션 2

7.1과는 다른 환경일 때에 대한 시뮬레이션을 시행하였다. 또한 디스크 2와 3에는 각각 250GB, 디스크 4에는 500GB의 비디오 세그먼트가 이미 저장되어 있다고 가정하였다.

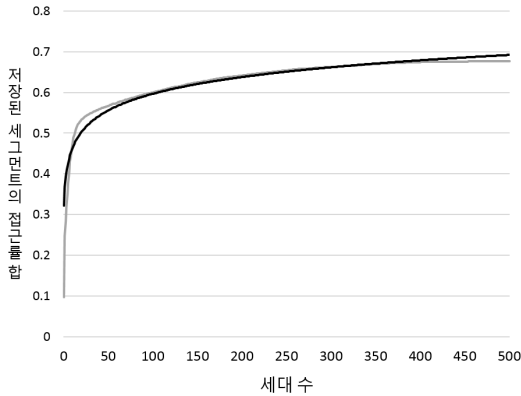
실험에 사용한 세그먼트는 총 15000개로, 세그먼트의 파일 크기는 모두 각기 다르다. 세그먼트들의 총 용량은 약 3.51TB이며, 미리 서버에 1TB가 저장되어 있었으므로 $S^{server} > C^{server}$ 이다. 이 중 1.547TB는 각 세그먼트를 위한 최대와 최소 비트율을 저장하기 위한 공간이며, 나머지 1.964TB에 대하여 저장 여부를 선택하게 된다. 버전별 인기도는 아래와 같게 설정했다.

$$\begin{aligned} \forall i, p_{i,1} &= p_i^* 0.1, p_{i,2} = p_i^* 0.3, p_{i,3} = p_i^* 0.3 \\ p_{i,4} &= p_i^* 0.2, p_{i,5} = p_i^* 0.1 \end{aligned} \quad (15)$$

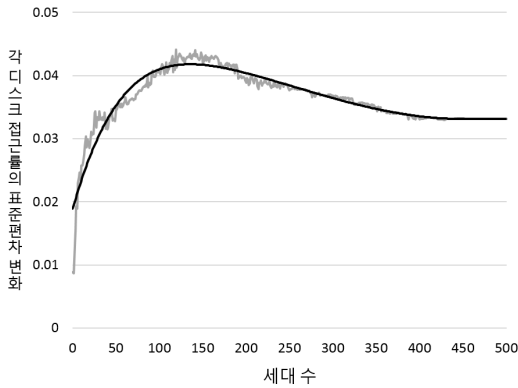
트랜스코딩 버전 중 최대, 최소 비트율인 경우를 제외한($j = 2, \dots, N^{ver} - 1$) 세그먼트들의 접근률 합은 0.8이다.

(그림 7,8,9)에 나오는 그래프의 회색 선은 평균값의 선, 검은 선은 추세선이다.

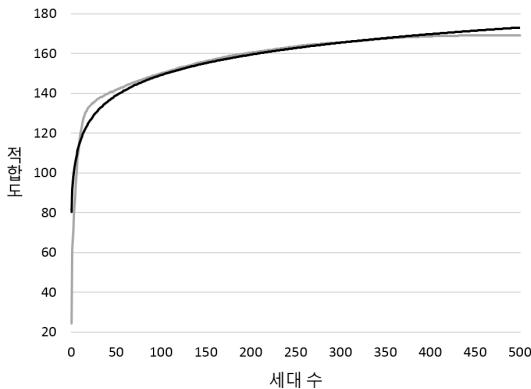
진화의 세대 수가 늘어날수록 (그림 7)과 (그림 8)을 통해 (식 5)의 결과 값을 증가시키면서 (식 6)을 감소시킴을 확인함으로써 3장에서 정의한 문제가 만족함을 알 수 있다.



(그림 7) 시뮬레이션 2에서 저장된 세그먼트의 접근률 합의 변화



(그림 8) 시뮬레이션 2의 각 디스크 접근률의 표준편차 변화



(그림 9) 시뮬레이션 2의 적합도 변화

시뮬레이션을 시행한 결과, 평균 12617.6개의 세그먼트가 저장되며, 디스크 1은 996.38GB, 디스크 2는 749.843GB, 디스크 3은 749.77GB, 디스크 4는 499.84GB가 저장됨으로써, 총 2.996TB가 저장된다. 또한 해당 실험은 1TB의 비디오 세그먼트가 미리 저장되어 있었다. 그러므로 저장된 세그먼트들이 (식 7)을 만족시켜 각 디스크 및 서버의 용량을 넘지 않음을 확인할 수 있다.

해당 실험에서 도출한 최종 결과의 평균값은 (식 5)는 0.676, (식 6)은 0.033이며, 평균 적합도는 169.047이다.

7.3. 신뢰성 확인

선택한 유전 알고리즘의 신뢰성을 확인하기 위해, 유전 알고리즘을 통해 구한 값과, 브루트 포스 (brute-force)을 통하여 구한 적합도의 최적값을 비교하였다.

비디오의 길이가 t 초일 경우, $N^{seg} = t/L$ 이다. 정의된 시뮬레이션 환경은 $N^{ver} = 5$, $N^{disk} = 4$ 이므로, 브루트 포스는 모든 경우의 수인 $4^{2t/L} * 5^{3t/L}$ 번을 계산하게 된다. 즉 시간이 매우 오래 걸리지만 모든 경우의 수를 확인함으로써 반드시 배낭문제의 최적값을 찾는다.

브루트 포스의 결과가 나오는 것이 너무 오래 걸림을 감안하여, 간단한 1개의 1분 30초짜리의 비디오의 경우로 실험을 진행했다. 모든 버전의 용량은 3.535GB이며, $S^{server} \leq C^{server}$ 이므로 $r=1.0$ 이다. 버전별 인기도는 아래와 같게 설정했다.

$$\begin{aligned} \forall i, \quad p_{i,1} &= p_i * 0.3, p_{i,2} = p_i * 0.1, p_{i,3} = p_i * 0.2 \\ p_{i,4} &= p_i * 0.3, p_{i,5} = p_i * 0.1 \end{aligned} \quad (16)$$

트랜스코딩 버전 중 최대, 최소 비트율인 경우를 제외한($j = 2, \dots, N^{ver} - 1$) 세그먼트들의 접근률 합은 0.4이다. 유전 알고리즘의 세대 수는 20번이다.

〈표 8〉 유전 알고리즘과 브루트 포스의 결과 비교

	유전 알고리즘	브루트 포스
소요시간	0.647초	733.114초
(식 5)	0.4	0.4
(식 6)	0.004	0.001
적합도	197.125	198.982

해당 실험은 총 5번 진행하여 평균값을 내었다. 최적값의 허용 오차 범위를 1%로 둘 경우, 적합도는 196.992 이상의 값을 가져야하며, 구해낸 유전 알고리즘의 적합도의 평균값은 197.125이므로 이에 해당한다. 그러므로 해당 유전 알고리즘은 최적 혹은 그에 근접한 값을 찾았다는 것을 확인할 수 있다.

8. 결론

본 논문에서는 비디오 스트리밍 서버에서 유전 알고리즘 기법을 활용한 사용자 인기도를 고려한 데이터 배치 기법을 제안했다. 기술의 발전으로 인해 비디오 스트리밍 서비스가 대중화 되었다. 많은 스트리밍 서비스에서는 동적 적응 스트리밍을 채택하였고, 이를 위해서는 다양한 비트율의 세그먼트들이 필요하다. 그러나 콘텐츠 별로 사용자에게 제공되는 빈도가 다르기에 모든 버전의 세그먼트를 저장할 필요가 없다. 또한 한 디스크만이 사용자의 접근이 몰리는 것은 비효율적이므로 어떤 디스크에 저장할 지에 대한 고려가 필요하다.

데이터 배치 기법을 설계하기 위해 우리는 서버를 구성하는 디스크의 용량 안에서, 트랜스코딩한 비디오의 세그먼트들 중 무엇을 저장할 지 선택하고, 어떻게 세그먼트를 디스크들에 분산시킬 지에 대한 문제를 정의하였다. 또한 이를 동시에 만족시키는 적합도 함수를 구현했다.

설계한 알고리즘을 테스트하기 위해, 실험을 거쳐 개체군의 크기, 선택 비율, 교배 비율, 돌연변이 비율을 정하여 시뮬레이션하였으며, 설정한 매개변수를 이용한 유전 알고리즘의 결과 값과 브루트 포스 기법을 이용하여 찾아낸 최적값을 비교하여 실험

의성을 검증했다.

본 논문의 데이터 배치 기법은 유전 알고리즘을 이용하여 효율적으로 비디오 세그먼트의 데이터 배치를 최적화할 수 있음을 보여준다. 차후 연구에서는 데이터 배치만이 아니라, 비디오 세그먼트 요청을 받았을 때, 서버가 사용자에게 이를 제공하는 부분에 대해 고려할 것이다. 또한 서버의 구성 및 제공하는 트랜스코딩 버전의 숫자 등의 시뮬레이션 환경을 실제와 더욱 근접하게 구성할 것이다.

참고문헌

- [1] Z. Wang, L. Sun, W. Z. C. Wu, Q. Zhuang and S. Yang, "A joint online transcoding and delivery approach for dynamic adaptive streaming", IEEE Transactions on Multimedia, 17(6):867-879, June 2015.
- [2] Gerassimos Barlas, "Cluster-based optimized parallel video transcoding", Parallel Computing, 38(4-5):226-244, April - May 2012.
- [3] H. Zhao, Q. Zheng, W. Zhang, B. Du and H. Li, "A segment-based storage and transcoding trade-off strategy for multi-version vod systems in the cloud", IEEE Transactions on Multimedia, 19(1):149-159, Sep. 2016.
- [4] L. Toni, R. Aparicio-Pardo, G. Simon, A. Blanc and P. Frossard, "Optimal set of video representations in adaptive streaming", In Proceedings of the ACM Multimedia Systems Conference, pp. 271-282, Mar. 2014.
- [5] R. Aparicio-Pardo, K. Pires, A. Blanc and G. Simon, "Transcoding live adaptive video streams at a massive scale in the cloud", In Proceedings of the ACM Multimedia Systems Conference, pp. 49-60, Mar. 2015.
- [6] 류기동, 김우제, "유전자 알고리즘을 이용한 IT 서비스 담당자 권역할당 최적화 모델 연구", 한국정보기술학회

- 논문지, 제12권 제12호, pp. 101-114, 2014.
- [7] M. Qiu, Z. Ming, J. Li, K. Gai and Z. Zong, "Phase-Change Memory Optimization for Green Cloud with Genetic Algorithm", IEEE Transactions on Computers, 64(12):3528-3540, Dec. 2015.
- [8] K. Dasgupta, B. Mandal, P. Dutta, Jyotsna K. Mandal and S. Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", Procedia Technology, vol. 10, pp. 340-347, 2013.
- [9] G. Portaluri, S. Giordano, D. Kliazovich and B. Dorronsoro, "A power efficient genetic algorithm for resource allocation in cloud computing data centers", In Proceedings of the IEEE International Conference on Cloud Networking (CloudNet), Oct. 2014.
- [10] X. Cheng and P. Mohapatra, "Quality-optimized downlink scheduling for video streaming applications in LTE networks", In Proceedings of IEEE Global Communications Conference (GLOBECOM), Dec. 2012.
- [11] H. Han and M. Song, "QoE-Aware Video Storage Power Management Based on Hot and Cold Data Classification", In Proceedings of the ACM Multimedia Systems Conference 2018, June. 2018.
- [12] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP Standards and Design Principles", In Proceedings of the ACM Multimedia Systems Conference, pp. 133 - 144, 2011.
- [13] D. Pisinger, "Algorithms for Knapsack Problems", PhD thesis, University of Copenhagen, 1995.
- [14] D. Krishnappa, M. Zink and R. Sitaraman, "Optimizing the video transcoding workflow in content delivery networks", In Proceedings of the ACM Multimedia Systems Conference, pp. 37 - 48, 2015.
- [15] S. Kaur and A. Verma, "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment", International Journal of Information Technology and Computer Science 10, pp. 74-79, Sep. 2012.
- [16] 윤경목, "유전알고리즘과 신경망 파라미터 값들의 변동에 의한 영화 추천시스템 성능향상", 한국차세대컴퓨팅학회 논문지, 제10권 제5호, pp. 56-64, 2014.
- [17] 김형석, 박준석, "가변 비트레이트 스트리밍을 위한 명암도 기반의 효율적인 비디오 화질측정 기법", 한국차세대컴퓨팅학회 논문지, 제 11권 제5호, pp. 63-71, 2015.
- [18] 백치선, 박준석, "비디오 트랜스코딩 서버의 QoS 관리 기법", 한국차세대컴퓨팅학회 논문지, 제9권 제4호, pp. 49-56, 2013.

■ 저자소개

◆ 이다영



- 2016년 인하대학교 컴퓨터공학부 학사
- 2018년 인하대학교 컴퓨터공학부 석사
- 2018년~현재 인하대학교 컴퓨터공학부 박사과정
- 관심분야 : 모바일 플랫폼, 실시간 시스템, 임베디드 시스템, 멀티미디어 시스템

◆ 송민석



- 1996년 서울대학교 컴퓨터공학과 학사
- 1998년 서울대학교 컴퓨터공학부 석사
- 2004년 서울대학교 전기컴퓨터공학부 박사
- 2005년~현재 인하대학교 컴퓨터공학부 교수
- 관심분야 : 실시간 시스템, 임베디드 시스템, 멀티미디어 시스템