대학 입학 지원 최적화 문제

Max Kapur

서울대학교 산업공학과
경영과학/최적화 연구실
maxkapur@snu.ac.kr

지도 교수: 홍성필

June 9, 2022

서론

대학 입학 지원 최적화 문제는 **새로운 조합 최적화 문제**이다.

예산 제약 조건 하에서, 다수 확률 변수로 이뤄진 포트폴리오의 **기대 최댓값을 최대화**하는 문제이다.

발표 요약:

- 대학 지원 전략을 최적화 문제로 **모형화**.
- 문제의 **계산 복잡도 분석**, **해법 제시**, 그리고 **알고리즘 구현**.

방법론적 지향

대학 지원 전략은 조합 최적화 중 여러 분야에 걸쳐 있다.

- 불확실한 성과, 효율적 투자선이 존재하므로 **일종의 포트폴리오 배분** 모형.
- **배낭 문제의 일반화**: 정수 채우기 (packing) 제약식, NP-completeness, 근사 해법의 필요.
- 목적함수는 **submodular 집합 함수**이며, 근사 해법 결과에 따라 대학 지원 문제가 submodular 최적화의 비교적 쉬운 인스턴스로 해석할 수 있다.

모형

입학 과정

단 한 명의 학생의 의사결정에 집중하자.

시장은 m개의 **대학교**를 포함하며, 그의 지표 집합은 $\mathcal{C} = \{1 \dots m\}$. j번째 학교의 이름은 c_j.

학생의 내신, 수능 점수, 기본 정보가 주어지면 각 학교의 **합격 확률** f_j를 알 수 있다.

독립 **확률 변수** $Z_j \sim \text{Bernoulli}(f_j)$는 학생이 합격하면 1, 아니면 0.

학생이 지원하는 학교의 집합 $\mathcal{X} \subseteq \mathcal{C}$를 **지원 포트폴리오**라고 부른다.

지원 전형료 예산, 원서를 작성하는 시간, 또는 나라의 정책에 따라 **지원 행동이 제한된다**. 본 논문은 단일 배낭 제약식 $\sum_{j \in \mathcal{X}} g_j \leq H$를 고려하며, 이때 g_j는 c_j의 **지원 비용**이라고 부른다.

효용 모형

c_j에 다니면 $t_j \geq 0$ 단위의 **효용이** 발생한다. Wlog, $t_j \leq t_{j+1}$.

어떤 대학에도 합격하지 않았을 때 효용은 t_0이며, wlog $t_0 = 0$이라고 할 수 있다 (논문에서 증명).

학생의 총 효용은 그가 지원하고 합격하는 **가장 좋은** 학교의 t_j-값:

$$\text{효용} = \max\{t_0, \max\{t_j Z_j : j \in \mathcal{X}\}\}$$

이의 기댓값은 $\mathcal{X}$의 **가치**라고 부르며 $v(\mathcal{X})$처럼 표기한다.

포트폴리오 가치의 함수 형태

$v(\mathcal{X})$를 함수로 표현하기 위해, 학생이 c_j에 **진학하는** 확률을 $p_j(\mathcal{X})$라고 하자.

c_j에 진학하는 조건은 c_j에 **지원하고**, **합격하고**, c_j보다 선호하는 학교에는 **합격하지 않았을** 때이다.

$$p_j(\mathcal{X}) = \begin{cases} f_j \prod_{\substack{i \in \mathcal{X}: \\ i > j}} (1 - f_i), & j \in \{0\} \cup \mathcal{X} \\ 0, & \text{그렇지 않은 경우.} \end{cases}$$

따라서,

$$v(\mathcal{X}) = \sum_{j=1}^{m} t_j p_j(\mathcal{X}) = \sum_{j \in \mathcal{X}} \left(f_j t_j \prod_{\substack{i \in \mathcal{X}: \\ i > j}} (1 - f_i) \right).$$

문제 정의

문제 1 (대학 지원 최적화 문제)

$$\text{maximize} \quad v(\mathcal{X}) = \sum_{j \in \mathcal{X}} \left(f_j t_j \prod_{\substack{i \in \mathcal{X}: \\ i > j}} (1 - f_i) \right)$$

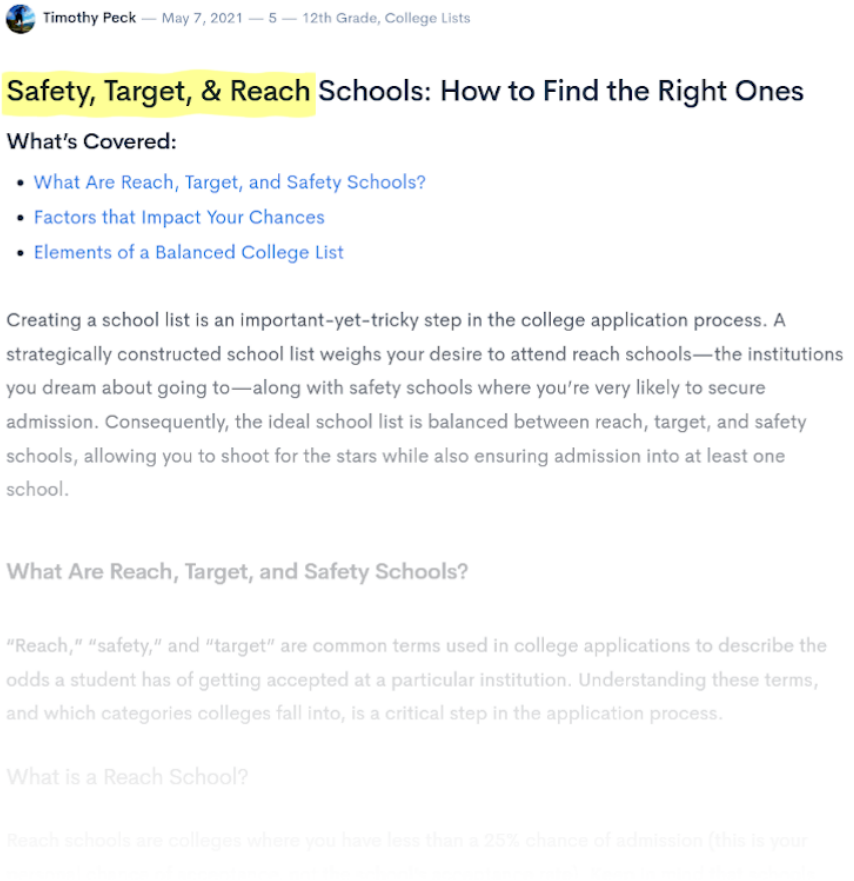
$$\text{subject to} \quad \mathcal{X} \subseteq \mathcal{C}, \quad \sum_{j \in \mathcal{X}} g_j \leq H$$

문제 2 (대학 지원 최적화 문제, 정수 비선형 계획 모형)

$$\text{maximize} \quad v(x) = \sum_{j=1}^{m} \left(f_j t_j x_j \prod_{i > j} (1 - f_i x_i) \right)$$

$$\text{subject to} \quad x_j \in \{0, 1\}, j \in \mathcal{C}; \quad \sum_{j=1}^{m} g_j x_j \leq H$$

기존의 해법



Timothy Peck — May 7, 2021 — 5 — 12th Grade, College Lists

Safety, Target, & Reach Schools: How to Find the Right Ones

What's Covered:

- What Are Reach, Target, and Safety Schools?
- Factors that Impact Your Chances
- Elements of a Balanced College List

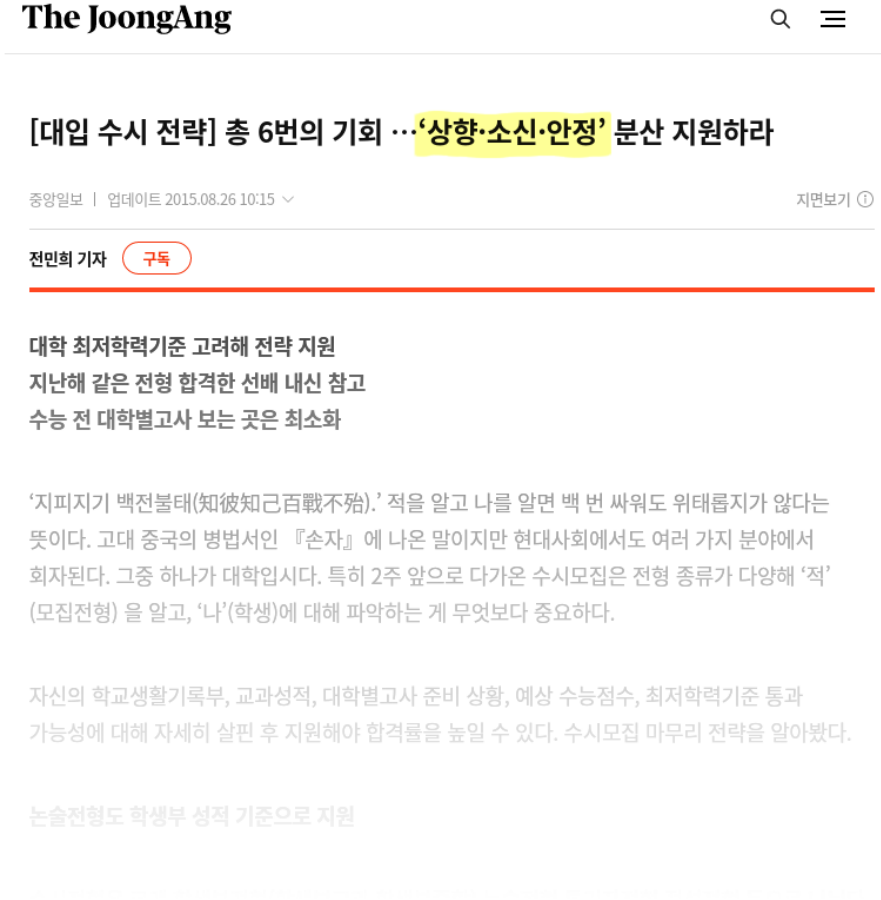
Creating a school list is an important-yet-tricky step in the college application process. A strategically constructed school list weighs your desire to attend reach schools—the institutions you dream about going to—along with safety schools where you're very likely to secure admission. Consequently, the ideal school list is balanced between reach, target, and safety schools, allowing you to shoot for the stars while also ensuring admission into at least one school.

What Are Reach, Target, and Safety Schools?

"Reach," "safety," and "target" are common terms used in college applications to describe the odds a student has of getting accepted at a particular institution. Understanding these terms, and which categories colleges fall into, is a critical step in the application process.

What is a Reach School?

Reach schools are colleges where you have less than a 25% chance of admission (this is your



The JoongAng

[대입 수시 전략] 총 6번의 기회 …'상향·소신·안정' 분산 지원하라

중앙일보 | 업데이트 2015.08.26 10:15 지면보기

전민희 기자 구독

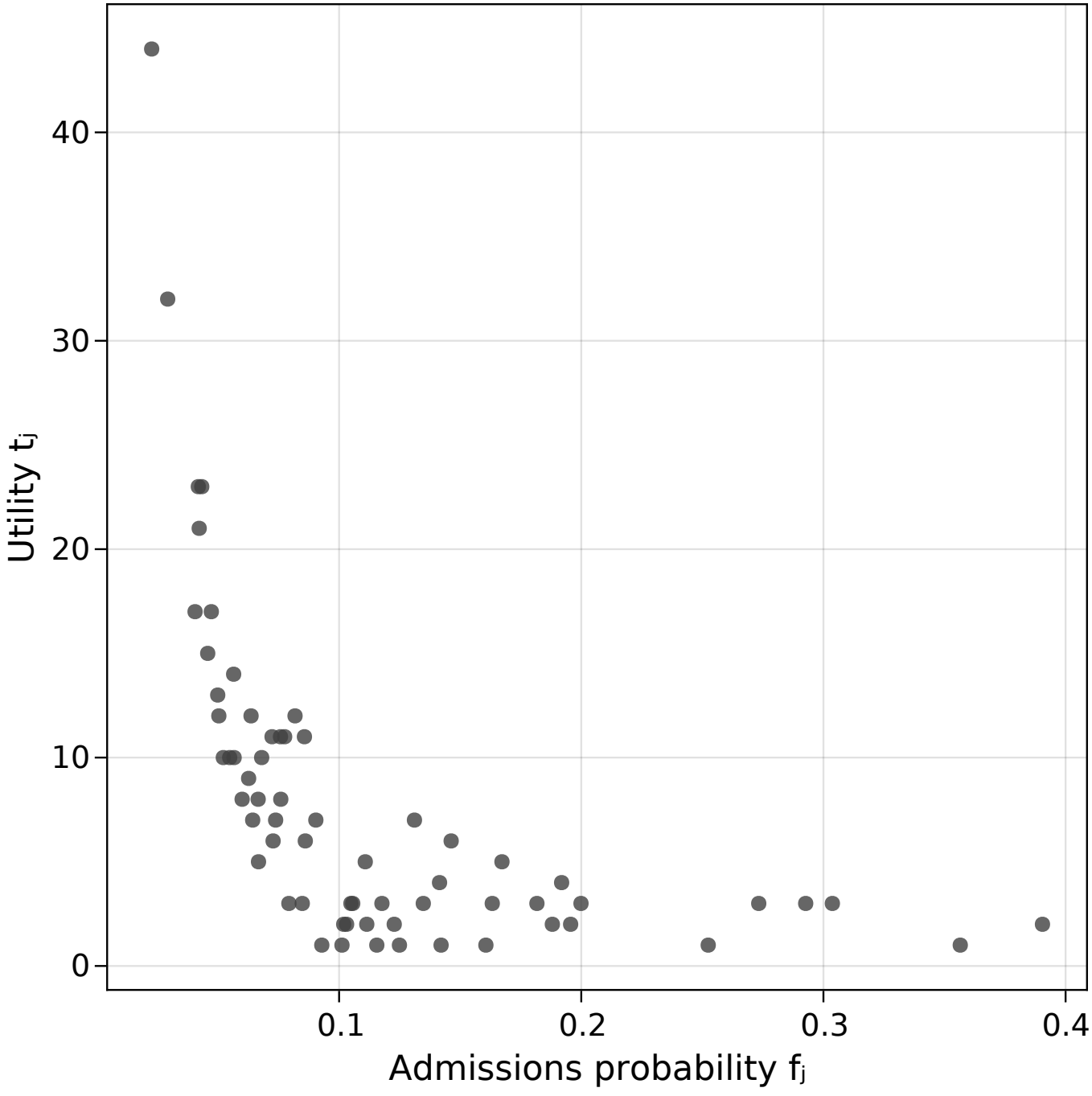
대학 최저학력기준 고려해 전략 지원
지난해 같은 전형 합격한 선배 내신 참고
수능 전 대학별고사 보는 곳은 최소화

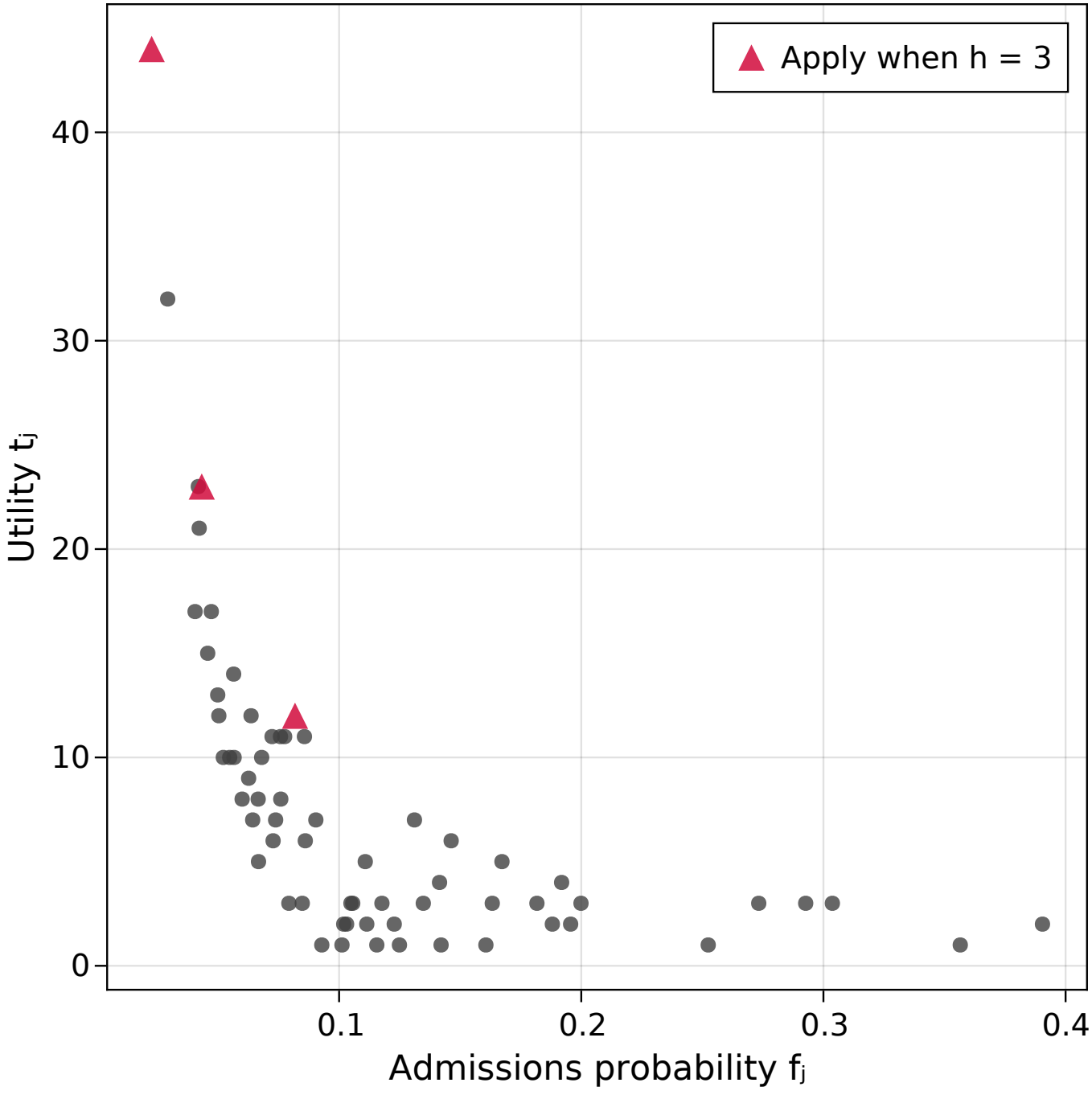
'지피지기 백전불태(知彼知己百戰不殆).' 적을 알고 나를 알면 백 번 싸워도 위태롭지가 않다는 뜻이다. 고대 중국의 병법서인 『손자』에 나온 말이지만 현대사회에서도 여러 가지 분야에서 회자된다. 그중 하나가 대학입시다. 특히 2주 앞으로 다가온 수시모집은 전형 종류가 다양해 '적'(모집전형) 을 알고, '나'(학생)에 대해 파악하는 게 무엇보다 중요하다.

자신의 학교생활기록부, 교과성적, 대학별고사 준비 상황, 예상 수능점수, 최저학력기준 통과 가능성에 대해 자세히 살핀 후 지원해야 합격률을 높일 수 있다. 수시모집 마무리 전략을 알아봤다.

논술전형도 학생부 성적 기준으로 지원

입학 컨설턴트의 조언, 믿을 만한가?


40
30
20
10
0
Utility t_j
0.1
0.2
0.3
0.4
Admissions probability f_j


Apply when h = 3
Utility t_j
Admissions probability f_j
0
10
20
30
40
0.1
0.2
0.3
0.4

기존의 해법

입학 컨설팅 산업에서는 주로 "상향 · 소신 · 안정 지원 학교" 각각 균일하게 지원하는 **배분적 휴리스틱**을 권장하며, 이는 **위험 회피적인** 전략임을 보일 수 있다.

또한 c_j에만 지원할 때 기대 효용이 $f_j t_j$이므로 다음 배낭 문제를 대리 문제로 푸는 **선형화 휴리스틱**이 있다.

$$\text{maximize} \sum_{j \in \mathcal{X}} f_j t_j \qquad \text{subject to} \sum_{j \in \mathcal{X}} g_j \leq H$$

그러나 **최적해보다 아주 안 좋은 해를 출력할 수 있다**.

Fu (2014)는 비슷한 문제를 **열거법**으로 풀었으나, $m \geq 20$일 때 비현실적인 방법이다.

본 연구는 **계산 시간과 정확도가 보장된** 알고리즘을 제시한다.

알고리즘 제시

해법 개관

대학 지원 문제를 2개의 경우로 나눌 수 있다.

- **알마의 문제:** 모든 $g_j = 1$, 즉 배낭 제약 대신 집합 크기 제약이 있는 특수한 경우.

  정확한 다항 시간 탐욕 해법을 제시한다.

- **엘리스의 문제:** NP-complete한 일반 경우.

  총 4개의 해법을 제시하며, **완전 다항 시간** $(1-\varepsilon)$**-근사 해법**이 하이라이트다.

일반적으로, 배낭 제약식 위에서 submodular 함수를 최대화하는 문제는 $(1-1/e)$보다 좋은 근사 계수를 가지는 해법이 존재하지 않으므로 (Kulik et al. 2013), 엘리스 문제의 FPTAS가 존재하는 것은 의미가 있는 결과다.

동일한 지원 비용의 경우

"알마의 문제"란 **특수한 경우**에서 먼저 모든 $g_j = 1$이다.

이때 H는 단순한 지원 개수 제한이 되며, $h = 3, m = 202$인 한국 정시 입학 과정과 비슷한 상황이다.

(이 경우, $H < m$임을 가정할 수 있으므로 모든 H의 다항식에 대해 m의 다항식인 상한이 존재한다. 이를 강조하기 위해 h처럼 소문자로 표기한다.)

집합 크기 제약 위에 단조 submodular 함수를 최대화하는 문제에 대해, 탐욕 해법의 근가 계수가 $(1 - 1/e)$임이 알려져 있다 (Fisher 외 1978). 대학 지원 문제에 대해, 같은 해법이 정확한 해법임을 증명하고 계산 시간을 줄일 수 있다.

탐욕 해법의 정확성

알마의 문제에 대해 다음 **포함 사슬 관계 (nestedness)** 성질은 $v(\mathcal{X})$를 최대화하는 순서대로 학교를 하나씩 추가하는 **탐욕 해법**의 최적성을 의미한다.

정리 1 (최적 포트폴리오의 포함 사슬 관계)

각 $\mathcal{X}_h$가 지원 제한 h에 대한 최적 포트폴리오며 위의 포함 사슬 관계를 만족하는 포트폴리오 수열 $\{\mathcal{X}_h\}_{h=1}^{m}$가 존재한다.

$$\mathcal{X}_1 \subset \mathcal{X}_2 \subset \cdots \subset \mathcal{X}_m$$

$v(\mathcal{X})$를 $O(1)$ 시간 안에 계산할 수 있는 변수 소거 기법을 개발하여 전체 해법의 계산 시간을 $O(hm)$으로 줄였다.

엘리스의 문제를 위한 알고리즘

$f_j = \varepsilon$으로 넣으면 $v(\mathcal{X})$를 선형 함수에 원하는 만큼 가깝게 만들 수 있다. 배낭 문제가 엘리스의 문제로 변환할 수 있음을 의미하므로 엘리스의 문제는 NP-complete하다.

4개의 알고리즘 제시:

- 선형 완화 문제와 해당 **분지한계** (branch-and-bound) 해법. 일반적인 INLP 문제에 대해 자주 쓰이는 방법이다.
- **총 지원 비용 기반 동적 계획**. $O(Hm + m \log m)$ (의사 다항) 시간에 정확한 해를 구하며, g_j가 작은 정수가 되는 "전형적" 인스턴스에 대해 매우 효율적인 해법.
- 포트폴리오 가치의 라운딩을 이용한 동적 계획. $O(m^3/\varepsilon)$ 시간에 $(1-\varepsilon)$-근사해를 출력하므로 **FPTAS!**
- **Simulated annealing** 휴리스틱. 빠르고 대부분 98% 이상의 최적성을 얻었다.

분지한계법: 선형 완화 문제

다음 선형 완화 문제 바탕으로 엘리스의 문제를 위한 분지한계법을 구성할 수 있다.

문제 3 (엘리스의 문제를 위한 선형 완화 문제)

$$\text{maximize} \quad v_{\mathrm{LP}}(x) = \sum_{j=1}^{m} f_j t_j x_j$$

$$\text{subject to} \quad \sum_{j=1}^{m} g_j x_j \leq H, \quad x \in [0,1]^m$$

포함 사슬 관계의 증명 과정에서 도출한 변수 소거법은 재활용해서 위에 **상한을 더 타이트하게** 조정한다.

문제 3은 연속 배낭 문제로서 쉽게 풀 수 있다 (Dantzig 1957; Balas와 Zemel 1980).

분지한계법: 개선 방향

이 기반으로 만든 간단한 알고리즘은 작은 $(m \leq 35)$ 인스턴스에 괜찮지만 분지 마디를 선택하는 휴리스틱으로 **개선할 여지가** 보인다.

Julia 코드 패키지에서 **부문제와 선형 완화 문제를 위한 modular한 데이터 구현을** 제시하며, 이를 더 세련된 INLP 솔버에 어렵지 않게 연결할 수 있다.

나중에 더 복합한 제약 조건을 도입하는 데에 분지한계법 알고리즘이 필요할 수 있으므로 유리하다.

지원 지출액 기반 동적 계획

$\{1, \ldots, j\}$에 속한 학교만 사용하면서 지원 지출액이 h를 넘지 않은 최적 포트폴리오의 가치를 $V[j, h]$라고 하자.

그러면 다음과 같은 **Bellman 식**으로 모든 $V[j, h]$-값은 재귀적으로 계산할 수 있다.

$$V[j, h] = \max\{V[j-1, h], (1 - f_j)V[j-1, h - g_j] + f_j t_j\}$$

이 식의 타당성은 학교를 t_j-값 순서대로 배열함에 의존.

따라서 $V[j, h]$-값으로 표를 채우는 시간이 $O(Hm + m \log m)$이며 이를 참고하면 $\mathcal{X}$를 쉽게 구할 수 있다.

전형적인 인스턴스에서 g_j가 작은 상수이므로 **매우 효율적인** 해법.

포트폴리오 가치 기반 동적 계획: 고정소숫점 산술

엘리스의 문제는 배낭 문제와 같이, 가치가 가장 높은 포트폴리오의 비용 대신 비용이 가장 낮은 포트폴리오의 가치를 탐색하는 **보완적인 동적 계획**이 존재한다.

포트폴리오의 근사적 가치를 정확도 P로 구성된 고정소수점 십진수 (fixed-point decimal)로 나타내자. 다, P는 소수점 뒤에 등장하는 숫자의 수이다. 이때 x를 가장 가까운 고정소수점 십진수로 내림한 것을 $r[x] = 10^{-P}\lfloor 10^P x \rfloor$라고 하자.

임의의 포트폴리오의 가치가 $\bar{U} = \sum_{j \in \mathcal{C}} f_j t_j$를 넘을 수 없다. 따라서 고정소수점 환경에서 발생할 수 있는 포트폴리오 가치로 이루어진 집합 $\mathcal{V}$는 유한 집합이다.

$$\mathcal{V} = \left\{0, 1 \times 10^{-P}, 2 \times 10^{-P}, \ldots, r\left[\bar{U} - 1 \times 10^{-P}\right], r\left[\bar{U}\right]\right\}$$

그러면 $|\mathcal{V}| = \bar{U} \times 10^P + 1$이다.

포트폴리오 가치 기반 동적 계획: 재귀 관계

$\{1, \ldots, j\}$에 속한 학교만 사용하면 (내림한) 가치가 최소한 v의 포트폴리오 중 지원 비용이 최소한 포트폴리오의 지출액을 $G[j, v]$라고 하자. 이때 다음 재귀 관계가 성립한다고 주장한다.

$$G[j,v] = \begin{cases} \infty, & t_j < v \\ \min\{G[j-1,v], g_j + G[j-1, v-\Delta_j(v)]\}, & t_j \geq v \end{cases}$$

where $$\Delta_j(v) = \begin{cases} r\left[\frac{f_j}{1-f_j}(t_j - v)\right], & f_j < 1 \\ \infty, & f_j = 1 \end{cases}$$

이제 $G[j, v]$-값으로 표를 채우면 근사 최적 포트폴리오를 구할 수 있다.

논문에서 $P \leftarrow \left\lceil \log_{10}\left(m^2/\varepsilon \bar{U}\right)\right\rceil$로 넣으면 $(1-\epsilon)$-근사해가 보장되며, 해당 표의 크기가 $O(m^3/\varepsilon)$이므로 이 알고리즘은 **FPTAS**이다.

Simulated annealing 해법

SA는 유명한 확률적 지역 탐색 기법이다.

정확도 보장은 없지만, 계산 시간이 아주 짧다. $\Rightarrow$ 차량 라우팅처럼, "충분히 좋은" 해가 실시간에 필요한 분야에서 인기가 많다.

SA의 재료:

- **초기해**. 본 연구에서 선형화 휴리스틱을 적용하고 해당 배낭 문제를 탐욕 휴리스틱으로 풀어서 초기해를 구한다.
- **해의 이웃해를 생성하는** 확률적 방법. $\mathcal{X}$의 가능성이 깨어질 때까지 학교를 무작위로 추가하고, 가능성이 복원될 때까지 무작위로 빼는 방법을 사용한다.
- 해법이 이웃해로 바꾸는 확률을 결정하는 **온도 모수**. 그리드 탐색을 적용한 결과로 $T = 1/4$ 그리고 $r = 1/16$을 선택했다.

계산 실험 결과

실험 방법

해법의 효율성과 정확성을 탐구하고자 **3개의 계산 실험을** 진행했다.

m개의 학교로 구성된 가상 인스선트 레시피:

- 각 t_j는 지수 분포에서 생성한다.
- (상향/안정 지원 학교를 만들기 위해) $1/t_j$과 비례하도록 f_j를 생성한다.
- 알마의 문제: $h = \lfloor m/2 \rfloor$.
- 엘리스의 문제: g_j 작은 정수, $H = \lfloor \frac{1}{2} \sum g_j \rfloor$.

모든 알고리즘은 Julia (v1.8.0b1) 언어로 구현했다. Github 리포 (Kapur 2022) 혹은 Julia 패키지 등록 (OptimalApplication.jl)에서 코드를 공유한다.

실험 1

실험 1에서 알마 문제의 탐욕 해법을 위한 **2개의 데이터 구현을** 비교한다.

각 칸에서 50개의 시장을 생성하고 최적해를 3번 계산한 것 중에 최소 시간을 기록. 표에서 나타나는 값은 평균 (표준편차) 시간이며 단위는 ms.

m	**탐욕 알고리즘 (목록 구현)**	**탐욕 알고리즘 (힙 구현)**
16	0.00 (0.00)	0.01 (0.00)
64	0.03 (0.00)	0.08 (0.02)
256	0.15 (0.01)	0.97 (0.22)
1024	2.31 (0.05)	14.44 (1.66)
4096	37.85 (0.61)	245.74 (17.33)
16384	585.75 (2.11)	4728.59 (552.25)

반복 단계마다 $\arg\max$ 연산이 필요하므로 힙 구현에는 매력이 있지만, 그의 모든 원소를 수정해야 하므로 만회하지 않는다.

Experiment 2

실험 2에서 엘리스 문제를 위한 **정확도가 보장된** 해법의 계산 시간을 비교한다.

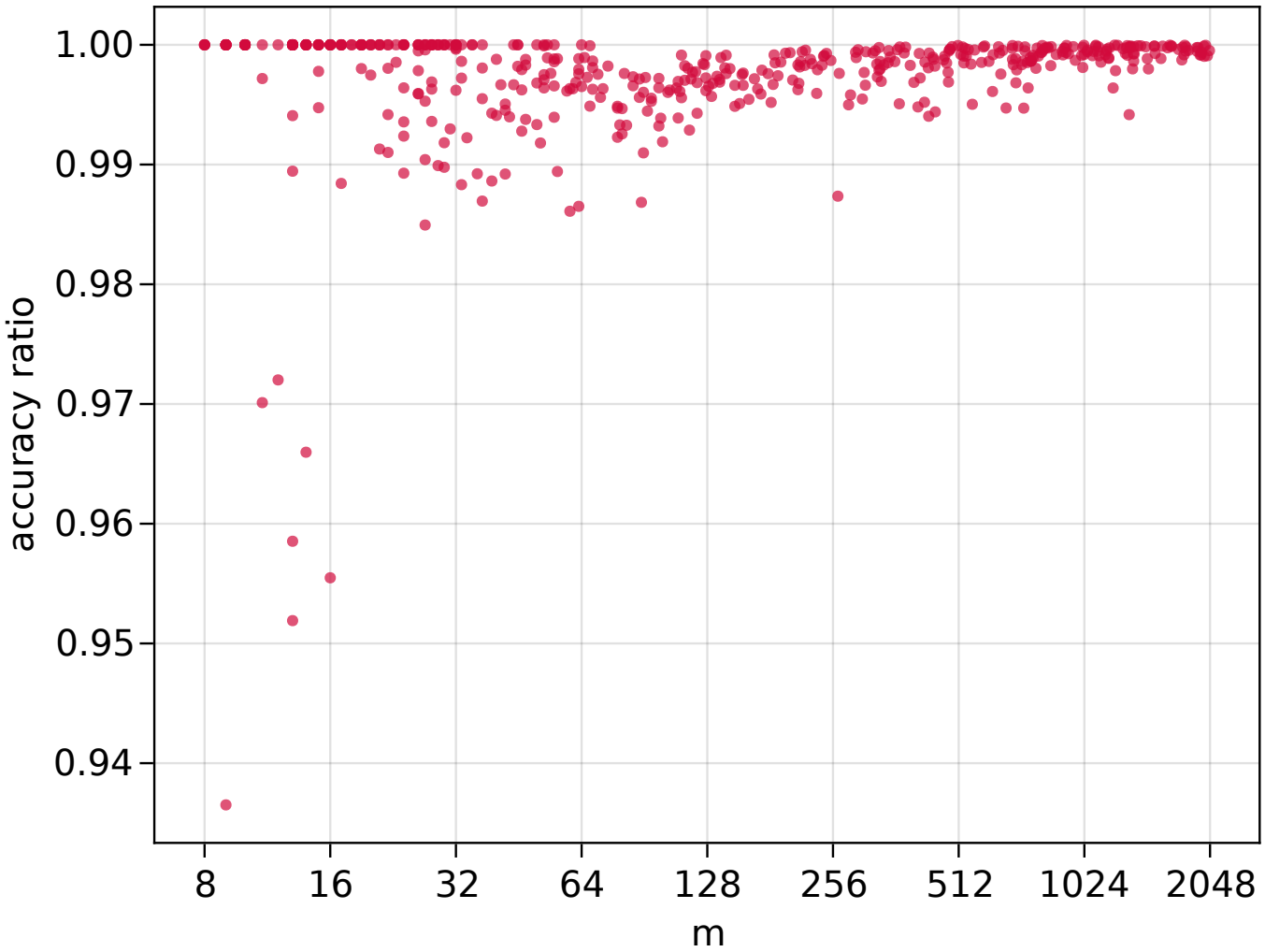
m	**분지한계법**	**지출액 DP**	**FPTAS,** $\varepsilon = 0.5$	**FPTAS,** $\varepsilon = 0.05$
8	0.04 (0.02)	0.01 (0.00)	0.05 (0.01)	0.21 (0.06)
16	0.22 (0.11)	0.07 (0.01)	0.43 (0.10)	3.15 (0.74)
32	166.20 (422.31)	0.31 (0.05)	2.38 (0.38)	33.38 (11.68)
64	— (—)	1.36 (0.18)	15.32 (2.77)	405.73 (125.98)
128	— (—)	6.52 (0.72)	84.50 (22.59)	2362.19 (1095.11)
256	— (—)	31.23 (1.91)	1085.60 (1186.24)	22129.92 (6588.40)

실험 데이터에서 g_j가 작은 정수이므로 **지원 지출액 기반 동적 계획은 상당한 우위**를 발휘한다.

FPTAS의 병목 요소는 계산 시간이 아니라 메모리 소모량.

Experiment 3

실험 3에서 simulated annealing 해법의 **실무적 정확도를** 가상 인스턴스를 통해 탐구한다.



결론

결론

"Maximax" 형태와 정수 조건 때문에 대학 지원 문제는 **이론적으로 흥미로운 문제이다**. Submodular 최대화 문제지만, 근사 해법의 성질은 배낭 문제에 더 가깝다 (cf. Fisher 외 1978; Kulik 외 2013; Kellerer 외 2004).

좋은 대학 지원 전략에는 **금전적 가치가 있다**. 미국 입학 컨설턴트의 시간당 급료는 평균 200달러 (Sklarow 2018)!

$\Rightarrow$ 공공 이익을 위해 코드는 open-source license로 공개 (Kapur 2022).

향후 연구:

- 고전적 Markowitz (1952)자산배분 모형처럼, 위험 회피를 명시적으로 다루는 새로운 목적함수.
- 한국 입학 과정의 가나다군과 같은 다각화 제약 조건.
- 동적 계획의 메모리 소모량 절감.

참고 문헌 I

Balas, Egon and Eitan Zemel. 1980. "An Algorithm for Large Zero-One Knapsack Problems." *Operations Research* 28 (5): 1130–54. https://doi.org/10.1287/opre.28.5.1130.

Dantzig, George B. 1957. "Discrete-Variable Extremum Problems." *Operations Research* 5 (2): 266–88.

Fisher, Marshall, George Nemhauser, and Laurence Wolsey. 1978. "An analysis of approximations for maximizing submodular set functions—I." *Mathematical Programming* 14: 265–94.

Fu, Chao. 2014. "Equilibrium Tuition, Applications, Admissions, and Enrollment in the College Market." *Journal of Political Economy* 122 (2): 225–81. https://doi.org/10.1086/675503.

Kapur, Max. 2022. "OptimalApplication." GitHub repository. https://github.com/maxkapur/OptimalApplication.jl.

참고 문헌 II

Kellerer, Hans, Ulrich Pferschy, and David Pisinger. 2004. *Knapsack Problems.* Berlin: Springer.

Kulik, Ariel, Hadas Shachnai, and Tami Tamir. 2013. "Approximations for Monotone and Nonmonotone Submodular Maximization with Knapsack Constraints." *Mathematics of Operations Research* 38 (4): 729–39. https://doi.org/10.1287/moor.2013.0592.

Markowitz, Harry. 1952. "Portfolio Selection." *The Journal of Finance* 7 (1): 77–91. https://www.jstor.org/stable/2975974.

Sklarow, Mark. 2018. *State of the Profession 2018: The 10 Trends Reshaping Independent Educational Consulting.* Technical report, Independent Educational Consultants Association. https://www.iecaonline.com/wp-content/uploads/2020/02/IECA-Current-Trends-2018.pdf.

알고리즘 요약

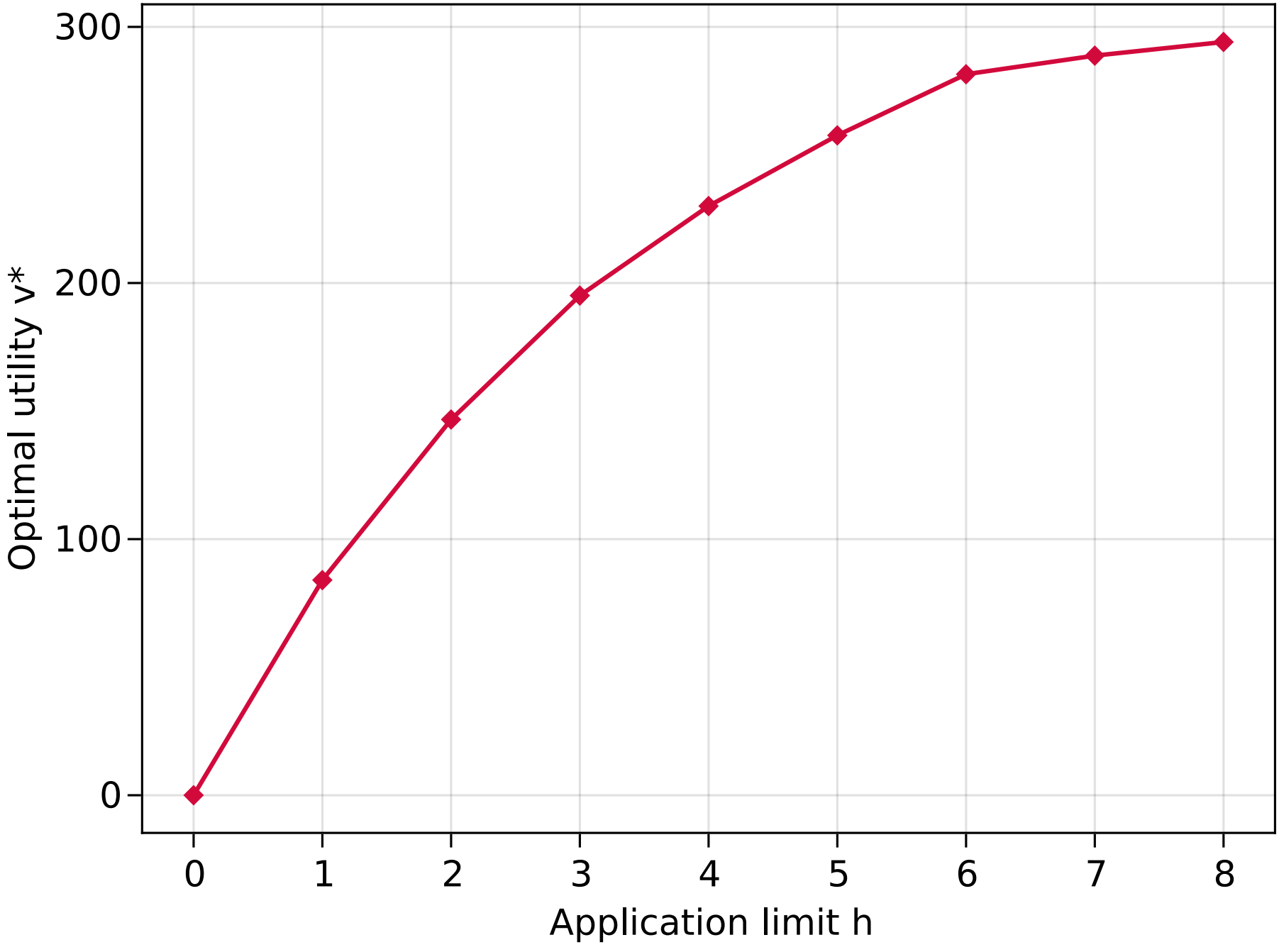
알고리즘	문제	제한	정확도	계산 시간
나이브	동일한 지원 비용	없음	$(1/h)$-근사	$O(m)$
탐욕 해법	동일한 지원 비용	없음	정확	$O(hm)$
Branch and bound	일반 문제	없음	정확	$O(2^m)$
지원 비용 동적 계획	일반 문제	g_j 정수	정확	$O(Hm + m \log m)$
FPTAS	일반 문제	없음	$(1-\varepsilon)$-근사	$O(m^3/\varepsilon)$
Simulated annealing	일반 문제	없음	0-근사	$O(Nm)$

작은 예제

$m = 8$개의 학교로 이루어진 가상 입학 시장의 대학교 자료와 최적 지원 포트폴리오.

지표 j	**학교** c_j	**합격 확률** f_j	**효용** t_j	**지원 순위**	$v(\mathcal{X}_h)$
1	수성대	0.39	200	4	230.0
2	금성대	0.33	250	2	146.7
3	화성대	0.24	300	6	281.5
4	목성대	0.24	350	1	84.0
5	토성대	0.05	400	7	288.8
6	천왕성대	0.03	450	8	294.1
7	해왕성대	0.10	500	5	257.7
8	명왕성대	0.12	550	3	195.1

포함 사슬 관계 성질에 따라, 지원 제한이 h일 때 최적 포트폴리오는 지원 순위가 h 이하인 h개의 학교로 구성된다.


300
200
100
0
Optimal utility v*
0
1
2
3
4
5
6
7
8
Application limit h