# Complexity analysis and algorithms for the college application problem

Max Kapur[1] and Sung-Pil Hong[2]

[1,2]Department of Industrial Engineering, Seoul National University
[1]Email: maxkapur@gmail.com

June 28, 2022

# Contents



# 1 Introduction

This study considers the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \mathrm{E}\Big[\max\big\{t_0, \max\{t_j Z_j : j \in \mathcal{X}\}\big\}\Big] \\ \text{subject to} \quad & \mathcal{X} \subseteq \mathcal{C}, \quad \sum_{j \in \mathcal{X}} g_j \leq H \end{aligned} \tag{1}$$

Here $\mathcal{C} = \{1 \dots m\}$ is an index set; $H > 0$ is a budget parameter; for $j = 1 \dots m$, $g_j > 0$ is a cost parameter and $Z_j$ is a random, independent Bernoulli variable with probability $f_j$; and for $j = 0 \dots m$, $t_j \geq 0$ is a utility parameter.

We refer to this problem as the *optimal college application* problem, as follows. Consider an admissions market with $m$ colleges. The $j$th college is named $c_j$. Consider a single prospective student in this market, and let each $t_j$-value indicate the utility she associates with attending $c_j$, where her utility is $t_0$ if she does not attend college. Let $g_j$ denote the application fee for $c_j$ and $H$ the student's total budget to spend on application fees. Lastly, let $f_j$ denote the student's probability of being admitted to $c_j$ if she applies, so that $Z_j$ equals one if she is admitted and zero if not. It is appropriate to assume that the $Z_j$ are statistically independent as long as $f_j$ are probabilities estimated specifically for this student (as opposed to generic acceptance rates). Then the student's objective is to maximize the expected utility associated with the best school she gets into within this budget. Therefore, her optimal college application strategy is given by the solution $\mathcal{X}$ to the problem above, where $\mathcal{X}$ represents the set of schools to which she applies.

# 2 Preliminaries

For the remainder of the paper, we assume with minimal loss of generality that $f_j \in \mathbb{Q}$, $t_j \in \mathbb{N}$, $g_j \in \mathbb{N}$, and $H \in \mathbb{N}$; that the $f_j$-values have the same denominator $D$; and that $t_0 < t_1 \leq \cdots \leq t_m$, $g_j \leq H$, and $\sum g_j > H$. Unless otherwise noted, we assume that $t_0 = 0$, an assumption we justify below.

### 2.1 The objective function

This subsection derives a closed-form expression for the objective function of (1).

We refer to the set $\mathcal{X} \subseteq \mathcal{C}$ of schools to which a student applies as her *application portfolio*. The expected utility the student receives from $\mathcal{X}$ is called its *valuation*. Given an application portfolio, let $p_j(\mathcal{X})$ denote the probability that the student attends $c_j$. This occurs if and only if she *applies* to $c_j$, is *admitted* to $c_j$, and is *rejected* from any school she prefers to $c_j$; that is, any school with higher index. Hence, for $j = 0 \ldots m$,

$$p_j(\mathcal{X}) = \begin{cases} f_j \prod_{\substack{i \in \mathcal{X}: \\ i > j}} (1 - f_i), & j \in \{0\} \cup \mathcal{X} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The following proposition follows by computing $v(\mathcal{X}) = \sum_{j=0}^{m} t_j p_j(\mathcal{X})$.

**Proposition 1** (Closed form of portfolio valuation function)**.**

$$v(\mathcal{X}) = \sum_{j=0}^{m} t_j p_j(\mathcal{X}) = \sum_{j \in \{0\} \cup \mathcal{X}} \Big( f_j t_j \prod_{\substack{i \in \mathcal{X}: \\ i > j}} (1 - f_i) \Big) \tag{3}$$

Next, we show that without loss of generality, we may assume that $t_0 = 0$.

**Lemma 1.** *For some $\gamma \leq t_0$, let $\bar{t}_j = t_j - \gamma$ for $j = 0 \ldots m$. Then $v(\mathcal{X}; \bar{t}_j) = v(\mathcal{X}; t_j) - \gamma$ regardless of $\mathcal{X}$.*

*Proof.* By definition, $\sum_{j=0}^{m} p_j(\mathcal{X}) = \sum_{j \in \{0\} \cup \mathcal{X}} p_j(\mathcal{X}) = 1$. Therefore

$$\begin{aligned} v(\mathcal{X}; \bar{t}_j) &= \sum_{j \in \{0\} \cup \mathcal{X}} \bar{t}_j p_j(\mathcal{X}) = \sum_{j \in \{0\} \cup \mathcal{X}} (t_j - \gamma) p_j(\mathcal{X}) \\ &= \sum_{j \in \{0\} \cup \mathcal{X}} t_j p_j(\mathcal{X}) - \gamma = v(\mathcal{X}; t_j) - \gamma \end{aligned} \tag{4}$$

which completes the proof. ☐

### 2.2 Submodularity

**Definition 1** (Submodular set function)**.** Given a ground set $\mathcal{C}$ and function $v : 2^{\mathcal{C}} \mapsto \mathbb{R}$, $v(\mathcal{X})$ is called a *submodular set function* if and only if $v(\mathcal{X}) + v(\mathcal{Y}) \geq v(\mathcal{X} \cup \mathcal{Y}) + v(\mathcal{X} \cap \mathcal{Y})$ for all $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{C}$. Furthermore, if $v(\mathcal{X} \cup \{k\}) - v(\mathcal{X}) \geq 0$ for all $\mathcal{X} \subset \mathcal{C}$ and $k \in \mathcal{C} \setminus \mathcal{X}$, $v(\mathcal{X})$ is said to be a *nondecreasing* submodular set function.

**Theorem 1.** *The college application portfolio valuation function $v(\mathcal{X})$ is a nondecreasing submodular set function.*

*Proof.* See Kapur (2022, § 2.3). ☐

## 3 NP-completeness

In this section, we will show that the college application problem is NP-complete by transformation from the binary knapsack problem, which is known to be NP-complete (Garey and Johnson 1979, § 3.2.1). We begin by presenting the knapsack and college applications as decision problems.

**Definition 2** (Decision form of knapsack problem (KP)). **Instance:** a set $\mathcal{B}$ of $m$ objects, utility values $u_j \in \mathbb{N}$ and weight $w_j \in \mathbb{N}$ for each $j \in \mathcal{B}$, knapsack capacity $W \in \mathbb{N}$, and target utility $U \in \mathbb{N}$. **Question:** Is there a set $\mathcal{B}' \subseteq \mathcal{B}$ having $\sum_{j\in\mathcal{B}'} u_j \geq U$ and $\sum_{j\in\mathcal{B}'} w_j \leq W$?

**Theorem 2.** *KP is NP-complete.*

*Proof.* See Garey and Johnson (1979, § 3.2.1). □

**Definition 3** (Decision form of the college application problem (CAP)). **Instance:** an instance $(f, t, g, H)$ of the college application problem and a target valuation $V$. **Question:** Is there a a portfolio $\mathcal{X} \subseteq \mathcal{C}$ having $v(\mathcal{X}) \geq V$ and $\sum_{j\in\mathcal{X}} g_j \leq H$?

**Theorem 3.** *CAP is NP-complete.*

*Proof.* (CAP $\in$ NP.) Self-evident.

(KP $\propto$ CAP.) Consider an instance of the knapsack problem, and we will construct an instance of Problem 3 that is a yes-instance if and only if the corresponding knapsack instance is a yes-instance. Without loss of generality, we may assume that the objects in $\mathcal{B}$ are indexed in increasing order of $u_j$, that each $u_j > 0$, and that each $w_j \leq W$.

Let $U_{\max} = \sum_{j\in\mathcal{B}} u_j$ and $\delta = 1/mU_{\max} > 0$, and construct an instance of CAP with $\mathcal{C} = \mathcal{B}$, $H = W$, all $f_j = \delta$, and each $t_j = u_j/\delta$. Clearly, $\mathcal{X} \subseteq \mathcal{C}$ is feasible for CAP if and only if it is feasible for the knapsack instance. Now, we observe that for any nonempty $\mathcal{X}$,

$$\begin{aligned}\sum_{j\in\mathcal{X}} u_j = \sum_{j\in\mathcal{X}} f_j t_j > \sum_{j\in\mathcal{X}} \Big( f_j t_j \prod_{\substack{j'\in\mathcal{X}:\\ j'>j}} (1 - f_{j'}) \Big) = v(\mathcal{X}) \\ = \sum_{j\in\mathcal{X}} \Big( u_j \prod_{\substack{j'\in\mathcal{X}:\\ j'>j}} (1-\delta) \Big) \geq (1-\delta)^m \sum_{j\in\mathcal{X}} u_j \\ \geq (1 - m\delta) \sum_{j\in\mathcal{X}} u_j \geq \sum_{j\in\mathcal{X}} u_j - m\delta U_{\max} = \sum_{j\in\mathcal{X}} u_j - 1.\end{aligned} \tag{5}$$

This means that the utility of an application portfolio $\mathcal{X}$ in the corresponding knapsack instance is the smallest integer greater than $v(\mathcal{X})$. That is, $\sum_{j\in\mathcal{X}} u_j \geq U$ if and only if $v(\mathcal{X}) \geq U - 1$. Taking $V = U - 1$ completes the transformation and concludes the proof. □

# 4 Optimization algorithms

In this section, we present three optimization algorithms for the college application problem. The first two are exact dynamic programming (DP) algorithms. The third algorithm is a fully polynomial-time approximation scheme (FPTAS) based on the second DP.

## 4.1 Dynamic program based on application expenditures

The first DP produces an optimal solution to the college application problem in $O(Hm + m \log m)$ time. Because we cannot assume that $H \leq m$, this represents a pseudopolynomial-time solution (Garey and Johnson 1979, § 4.2). However, it is quite effective for typical instances in which the application costs are small integers. It is also a polynomial-time algorithm for the special case of the college application problem in which each $g_j = 1$, meaning that $H$ is a limit on the *cardinality* $\mathcal{X}$, and $H < \sum g_j = m$ in any nontrivial instance. The algorithm resembles a familiar DP algorithm for the binary knapsack problem (Dantzig 1957).

For $j = 0 \dots m$ and $h = 0 \dots H$, let $\mathcal{X}(j, h)$ denote the optimal portfolio that uses only the schools $\{1, \dots, j\}$ and costs no more than $h$, and let $V(j, h) = v\big(\mathcal{X}(j, h)\big)$. Clearly, if $j = 0$ or $h = 0$, then $\mathcal{X}(j, h) = \emptyset$ and $V(j, h) = 0$. It is also convenient to let $V(j, h) = -\infty$ for all $h < 0$.

For the remaining indices, $\mathcal{X}(j,h)$ either contains $j$ or not. If it does not contain $j$, then $\mathcal{X}(j,h) = \mathcal{X}(j-1,h)$. On the other hand, if $\mathcal{X}(j,h)$ contains $j$, then its valuation is $(1-f_j)v(\mathcal{X}[j,h]\setminus\{j\}) + f_j t_j$. Therefore, $\mathcal{X}(j,h)\setminus\{j\}$ must make optimal use of the remaining budget over the remaining schools; that is, $\mathcal{X}(j,h) = \mathcal{X}(j-1,h-g_j) \cup \{j\}$. From these observations, we obtain the following recursion for $j = 1 \dots m$ and $h = 1 \dots H$:

$$V(j,h) = \max\big\{V(j-1,h), (1-f_j)V(j-1,h-g_j) + f_j t_j\big\} \tag{6}$$

with the convention that $-\infty \cdot 0 = -\infty$. Given the values of $V(j,h)$ at each index, the corresponding optimal portfolios are computed by observing that $\mathcal{X}(j,h)$ contains $j$ if and only if $V(j,h) > V(j-1,h)$. The optimal solution is given by $\mathcal{X}(m,H)$. The algorithm below performs these computations and outputs the optimal portfolio $\mathcal{X}$.

**Algorithm 1:** Application expenditures DP.

**Input:** $f \in \mathbb{Q}^m$, $t \in \mathbb{N}^m$, $g \in \mathbb{N}^m$, $H \in \mathbb{N}$.

1 Sort schools by $t_j$ ascending;
2 Fill a lookup table with the values of $V(j,h)$;
3 $h \leftarrow H$;
4 $\mathcal{X} \leftarrow \emptyset$;
5 **for** $j = m, m-1, \dots, 1$ **do**
6     **if** $V[j-1,h] < V[j,h]$ **then**
7         $\mathcal{X} \leftarrow \mathcal{X} \cup \{j\}$;
8         $h \leftarrow h - g_j$;
9     **end**
10 **end**
11 **return** $\mathcal{X}$

**Theorem 4** (Validity of Algorithm 1)**.** *Algorithm 1 produces an optimal application portfolio for the college application problem in $O(Hm + m\log m)$ time.*

*Proof.* Optimality follows from the foregoing discussion. Sorting $t$ is $O(m \log m)$. The bottleneck step is the creation of the lookup table for $V(j,h)$ in line 2. Each entry is generated in unit time, and the size of the table is $O(Hm)$. ☐

### 4.2 Dynamic program based on portfolio valuations

As with the knapsack problem, the college application problem admits a complementary DP that iterates on the value of the cheapest portfolio instead of on the cost of the most valuable portfolio.

Let $\mathbb{Z}/D$ denote the set of rational numbers having denominator $D$. For integers $0 \le j \le m$ and $v \in \mathbb{Q}_{D^m}$, let $\mathcal{W}(j,v)$ denote the least expensive portfolio that uses only schools $\{1,\dots,j\}$ and has valuation at least $v$, if such a portfolio exists. Denote its cost by $G[j,v] = \sum_{j\in\mathcal{W}(j,v)} g_j$, where $G(j,v) = \infty$ if $\mathcal{W}(j,v)$ does not exist. Clearly, if $v \le 0$, then $\mathcal{W}(j,v) = \emptyset$ and $G(j,h) = 0$, and that if $j = 0$ and $v > 0$, then $G[j,h] = \infty$. For the remaining indices (where $j, v > 0$), we claim that

$$G[j,v] = \begin{cases} \infty, & t_j < v \\ \min\big\{G[j-1,v], g_j + G[j-1, v - \Delta_j(v)]\big\}, & t_j \ge v \end{cases} \tag{7}$$

$$\text{where} \qquad \Delta_j(v) = \begin{cases} r\left[\frac{f_j}{1-f_j}(t_j - v)\right], & f_j < 1 \\ \infty, & f_j = 1. \end{cases} \tag{8}$$

In the $t_j < v$ case, any feasible portfolio must be composed of schools with utility less than $v$, and therefore its valuation can not equal $v$, meaning that $\mathcal{W}[j, v]$ is undefined. In the $t_j \geq v$ case, the first argument to $\min\{\}$ says simply that omitting $j$ and choosing $\mathcal{W}[j-1, v]$ is a permissible choice for $\mathcal{W}[j, v]$. If, on the other hand, $j \in \mathcal{W}[j, v]$, then

$$v(\mathcal{W}[j, v]) = (1 - f_j)v(\mathcal{W}[j, v] \setminus \{j\}) + f_j t_j. \tag{9}$$

Therefore, the subportfolio $\mathcal{W}[j, v] \setminus \{j\}$ must have a valuation of at least $v - \Delta$, where $\Delta$ satisfies $v = (1 - f_j)(v - \Delta) + f_j t_j$. When $f_j < 1$, the solution to this equation is $\Delta = \frac{f_j}{1-f_j}(t_j - v)$. By rounding this value down, we ensure that the true valuation of $\mathcal{W}[j, v]$ is *at least* $v - \Delta$. When $t_j \geq v$ and $f_j = 1$, the singleton $\{j\}$ has $v(\{j\}) \geq v$, so

$$G[j, v] = \min\big\{G[j-1, v], g_j\big\}. \tag{10}$$

Defining $\Delta_j(v) = \infty$ in this case ensures that $g_j + G[j-1, v - \Delta_j(v)] = g_j + G[j-1, v - \infty] = g_j$ as required.

Once $G[j, v]$ has been calculated at each index, the associated portfolio can be found by applying the observation that $\mathcal{W}[j, v]$ contains $j$ if and only if $G[j, v] < G[j-1, v]$. Then an approximate solution to the college application problem is obtained by computing the largest achievable objective value $\max\{w : G[m, w] \leq H\}$ and corresponding portfolio.

**Algorithm 2:** Fully polynomial-time approximation scheme for the college application problem.

**Input:** Utility values $t \in (0, \infty)^m$, admissions probabilities $f \in (0, 1]^m$, application costs $g \in (0, \infty)^m$, budget $H \in (0, \infty)^m$.
**Parameters:** Tolerance $\varepsilon \in (0, 1)$.

Index schools in ascending order by $t$;
Set precision $P \leftarrow \lceil \log_{10}(m^2/\varepsilon\bar{U}) \rceil$;
Fill a lookup table with the entries of $G[j, h]$;
$v \leftarrow \max\{w \in \mathcal{V} : G[m, w] \leq H\}$;
$\mathcal{X} \leftarrow \emptyset$;
**for** $j = m, m-1, \ldots, 1$ **do**
  **if** $G[j, v] < \infty$ *and* $G[j, v] < G[j-1, v]$ **then**
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{j\}$;
    $v \leftarrow v - \Delta_j(v)$;
  **end**
**end**
**return** $\mathcal{X}$

**Theorem 5** (Validity of Algorithm 3). *Algorithm 3 produces a $(1-\varepsilon)$-optimal application portfolio for the college application problem in $O(m^3/\varepsilon)$ time.*

### 4.3 Fully polynomial-time approximation scheme

We will represent approximate portfolio valuations using a fixed-point decimal with a precision of $P$, where $P$ is the number of digits to retain after the decimal point. Let $r[x] = 10^{-P}\lfloor 10^P x \rfloor$ denote the value of $x$ rounded down to its nearest fixed-point representation. Since $\bar{U} = \sum_{j \in \mathcal{C}} f_j t_j$ is an upper bound on the valuation of any portfolio, and since we will ensure that each fixed-point approximation is an underestimate of the portfolio's true valuation, the set $\mathcal{V}$ of valuations observable in the fixed-point framework is finite:

For the remainder of this subsection, unless otherwise specified, the word *valuation* refers to a portfolio's valuation within the fixed-point framework, with the understanding that this is an approximation. We will account for the approximation error below when we prove the dynamic program's validity.

For integers $0 \le j \le m$ and $v \in [-\infty, 0) \cup \mathcal{V}$, let $\mathcal{W}[j,v]$ denote the least expensive portfolio that uses only schools $\{1, \dots, j\}$ and has valuation at least $v$, if such a portfolio exists. Denote its cost by $G[j,v] = \sum_{j \in \mathcal{W}[j,v]} g_j$, where $G[j,v] = \infty$ if $\mathcal{W}[j,v]$ does not exist. It is clear that if $v \le 0$, then $\mathcal{W}[j,v] = \emptyset$ and $G[j,h] = 0$, and that if $j = 0$ and $v > 0$, then $G[j,h] = \infty$. For the remaining indices (where $j, v > 0$), we claim that

$$G[j,v] = \begin{cases} \infty, & t_j < v \\ \min\bigl\{G[j-1,v], g_j + G[j-1, v - \Delta_j(v)]\bigr\}, & t_j \ge v \end{cases} \tag{11}$$

$$\text{where} \qquad \Delta_j(v) = \begin{cases} r\left[\frac{f_j}{1-f_j}(t_j - v)\right], & f_j < 1 \\ \infty, & f_j = 1. \end{cases} \tag{12}$$

In the $t_j < v$ case, any feasible portfolio must be composed of schools with utility less than $v$, and therefore its valuation can not equal $v$, meaning that $\mathcal{W}[j,v]$ is undefined. In the $t_j \ge v$ case, the first argument to min{} says simply that omitting $j$ and choosing $\mathcal{W}[j-1,v]$ is a permissible choice for $\mathcal{W}[j,v]$. If, on the other hand, $j \in \mathcal{W}[j,v]$, then

$$v(\mathcal{W}[j,v]) = (1 - f_j) v(\mathcal{W}[j,v] \setminus \{j\}) + f_j t_j. \tag{13}$$

Therefore, the subportfolio $\mathcal{W}[j,v] \setminus \{j\}$ must have a valuation of at least $v - \Delta$, where $\Delta$ satisfies $v = (1 - f_j)(v - \Delta) + f_j t_j$. When $f_j < 1$, the solution to this equation is $\Delta = \frac{f_j}{1-f_j}(t_j - v)$. By rounding this value down, we ensure that the true valuation of $\mathcal{W}[j,v]$ is *at least* $v - \Delta$. When $t_j \ge v$ and $f_j = 1$, the singleton $\{j\}$ has $v(\{j\}) \ge v$, so

$$G[j,v] = \min\bigl\{G[j-1,v], g_j\bigr\}. \tag{14}$$

Defining $\Delta_j(v) = \infty$ in this case ensures that $g_j + G[j-1, v - \Delta_j(v)] = g_j + G[j-1, v - \infty] = g_j$ as required.

Once $G[j,v]$ has been calculated at each index, the associated portfolio can be found by applying the observation that $\mathcal{W}[j,v]$ contains $j$ if and only if $G[j,v] < G[j-1,v]$. Then an approximate solution to the college application problem is obtained by computing the largest achievable objective value $\max\{w : G[m,w] \le H\}$ and corresponding portfolio.

**Theorem 6** (Validity of Algorithm 3)**.** *Algorithm 3 produces a* $(1-\varepsilon)$*-optimal application portfolio for the college application problem in* $O(m^3/\varepsilon)$ *time.*

*Proof.* (Optimality.) Let $\mathcal{W}$ denote the output of Algorithm 3 and $\mathcal{X}$ the true optimum. We know that $v(\mathcal{X}) \le \bar{U}$, and because each singleton portfolio is feasible, $\mathcal{X}$ must be more valuable than the average singleton portfolio; that is, $v(\mathcal{X}) \ge \bar{U}/m$.

Because $\Delta_j(v)$ is rounded down in the recursion relation defined by (11) and (12), if $j \in \mathcal{W}[j,v]$, then the true value of $(1 - f_j) v\bigl(\mathcal{W}[j-1, v - \Delta_j(v)]\bigr) + f_j t_j$ may exceed the fixed-point valuation $v$ of $\mathcal{W}[j,v]$, but not by more than $10^{-P}$. This error accumulates additively with each school added to $\mathcal{W}$, but the number of additions is at most $m$. Therefore, where $v'(\mathcal{W})$ denotes the fixed-point valuation of $\mathcal{W}$ recorded in line 4 of the algorithm, $v(\mathcal{W}) - v'(\mathcal{W}) \le m 10^{-P}$.

We can define $v'(\mathcal{X})$ analogously as the fixed-point valuation of $\mathcal{X}$ when its elements are added in index order and its valuation is updated and rounded down to the nearest multiple of $10^{-P}$ at each addition in accordance with (13). By the same logic, $v(\mathcal{X}) - v'(\mathcal{X}) \le m 10^{-P}$. The optimality of $\mathcal{W}$ in the fixed-point environment implies that $v'(\mathcal{W}) \ge v'(\mathcal{X})$.

Applying these observations, we have

$$v(\mathcal{W}) \ge v'(\mathcal{W}) \ge v'(\mathcal{X}) \ge v(\mathcal{X}) - m 10^{-P} \ge \left(1 - \frac{m^2 10^{-P}}{\bar{U}}\right) v(\mathcal{X}) \ge (1 - \varepsilon)\, v(\mathcal{X}) \tag{15}$$

**Algorithm 3:** Fully polynomial-time approximation scheme for the college application problem.

**Input:** Utility values $t \in (0,\infty)^m$, admissions probabilities $f \in (0,1]^m$, application costs $g \in (0,\infty)^m$, budget $H \in (0,\infty)^m$.

**Parameters:** Tolerance $\varepsilon \in (0,1)$.

1 Index schools in ascending order by $t$;
2 Set precision $P \leftarrow \left\lceil \log_{10}\left(m^2/\varepsilon\bar{U}\right)\right\rceil$;
3 Fill a lookup table with the entries of $G[j,h]$;
4 $v \leftarrow \max\{w \in \mathcal{V} : G[m,w] \leq H\}$;
5 $\mathcal{X} \leftarrow \varnothing$;
6 **for** $j = m, m-1, \ldots, 1$ **do**
7     **if** $G[j,v] < \infty$ *and* $G[j,v] < G[j-1,v]$ **then**
8         $\mathcal{X} \leftarrow \mathcal{X} \cup \{j\}$;
9         $v \leftarrow v - \Delta_j(v)$;
10     **end**
11 **end**
12 **return** $\mathcal{X}$

which establishes the approximation bound.

(Computation time.) The bottleneck step is the creation of the lookup table in line 3, whose size is $m \times |\mathcal{V}|$. Since

$$|\mathcal{V}| = \bar{U} \times 10^P + 1 = \bar{U} \times 10^{\left\lceil \log_{10}\left(m^2/\varepsilon\bar{U}\right)\right\rceil} + 1 \leq \frac{m^2}{\varepsilon} \times \text{const.} \tag{16}$$

is $O(m^2/\varepsilon)$, the time complexity is as promised. □

Since its time complexity is polynomial in $m$ and $1/\varepsilon$, Algorithm 3 is an FPTAS for the college application problem (Vazirani 2001).

Algorithms 1 and 3 can be written using recursive functions instead of lookup tables. However, since each function references itself *twice,* the function values at each index must be recorded in a lookup table or memoized to prevent an exponential number of calls from forming on the stack.

### 4.4 Fully polynomial-time approximation scheme

## 5 Conclusion

## 6 References


Acharya, Mohan S., Asfia Armaan, and Aneeta S. Antony. 2019. “A Comparison of Regression Models for Prediction of Graduate Admissions.” In *Second International Conference on Computational Intelligence in Data Science.* https://doi.org/10.1109/ICCIDS.2019.8862140.

Ashlagi, Itai, and Afshin Nikzad. 2020. “What Matters in School Choice Tie-Breaking? How Competition Guides Design.” *Journal of Economic Theory* 190: article no. 105120. https://doi.org/10.1016/j.jet.2020.105120.

Assad, Arjang. 1985. “Nested Optimal Policies for Set Functions with Applications to Scheduling.” *Mathematics of Operations Research* 10 (1): 82–99.

Azevedo, Eduardo and Jacob Leshno. 2016. “A Supply and Demand Framework for Two-Sided Matching Markets.” *Journal of Political Economy* 124 (5): 1235–68. https://doi.org/10.1086/687476.

Badanidiyuru, Ashwinkumar and Jan Vondrák. 2014. "Fast Algorithms for Maximizing Submodular Functions." In *Proceedings of the 2014 Annual ACM–SIAM Symposium on Discrete Algorithms*, 1497–1514. https://doi.org/10.1137/1.9781611973402.110.

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.

Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B. Shah. 2017. "Julia: A Fresh Approach to Numerical Computing." *SIAM Review* 59: 65–98. https://doi.org/10.1137/141000671.

Blum, Manuel, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. 1973. "Time Bounds for Selection." *Journal of Computer and System Sciences* 7 (4): 448–61. https://doi.org/10.1016/S0022-0000(73)80033-9.

Bodoh-Creed, Aaron. 2020. "Optimizing for Distributional Goals in School Choice Problems." *Management Science* 66 (8): 3657–76. https://doi.org/10.1287/mnsc.2019.3376.

Calinescu, Gruia, Chandra Chekuri, Martin Pál, and Jan Vondrák. 2011. "Maximizing a Monotone Submodular Function Subject to a Matroid Constraint." *SIAM Journal on Computing* 40 (6): 1740–66. https://doi.org/10.1137/080733991.

Carraway, Robert, Robert Schmidt, and Lawrence Weatherford. 1993. "An Algorithm for Maximizing Target Achievement in the Stochastic Knapsack Problem with Normal Returns." *Naval Research Logistics* 40 (2): 161–73. https://doi.org/10.1002/nav.3220400203.

Chekuri, Chandra, Jan Vondrák, and Rico Zenklusen. 2014. "Submodular Function Maximization via the Multilinear Relaxation and Contention Resolution Schemes." *SIAM Journal on Computing* 43 (6): 1831–79. https://doi.org/10.1137/110839655.

Cormen, Thomas, Charles Leiserson, and Ronald Rivest. 1990. *Introduction to Algorithms.* Cambridge, MA: The MIT Press.

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

Dean, Brian, Michel Goemans, and Jan Vondrák. 2008. "Approximating the Stochastic Knapsack Problem: The Benefit of Adaptivity." *Mathematics of Operations Research* 33 (4): 945–64. https://doi.org/10.1287/moor.1080.0330.

Edmonds, Jack. 1971. "Matroids and the Greedy Algorithm." *Mathematical Programming* 1: 127–36. https://doi.org/10.1007/BF01584082.

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.

Gale, David, and Lloyd Shapley. 1962. "College Admissions and the Stability of Marriage." *American Mathematics Monthly* 69 (1): 9–15. https://doi.org/10.2307/2312726.

Garey, Michael and David Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman and Company.

Hartlaub, Vanessa and Thorsten Schneider. 2012. "Educational Choice and Risk Aversion: How Important Is Structural vs. Individual Risk Aversion?" *SOEPpapers on Multidisciplinary Panel Data Research,* no. 433. https://www.diw.de/documents/publikationen/73/diw_01.c.394455.de/diw_sp0433.pdf.

Jeon, Minhee. 2015. "[College application strategy] Six chances total... divide applications across reach, target, and safety schools" (in Korean). Jungang Ilbo, Aug. 26. https://www.joongang.co.kr/article/18524069.

Kahneman, Daniel. 2011. *Thinking, Fast and Slow.* New York: Macmillan.

Kapur, Max. 2021. "Characterizing Nonatomic Admissions Markets." ArXiv, July 3, 2021. https://arxiv.org/abs/2107.01340.

Kapur, Max. 2022. "The College Application Problem." Master's thesis, Seoul National Uni-

versity.
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.
Lim, Daniel Kibum. 2013. "A Simulation Approach to Predicting College Admissions." Master's thesis, University of California Los Angeles. https://escholarship.org/uc/item/8r2695n5.
Markowitz, Harry. 1952. "Portfolio Selection." *The Journal of Finance* 7 (1): 77–91. https://www.jstor.org/stable/2975974.
Martello, Silvano and Paolo Toth. 1990. *Knapsack Problems: Algorithms and Computer Implementations.* New York: John Wiley & Sons.
Meucci, Attilio. 2005. *Risk and Asset Allocation.* Berlin: Springer-Verlag, 2005.
Nemhauser, George and Laurence Wolsey. 1978. "Best Algorithms for Approximating the Maximum of a Submodular Set Function." *Mathematics of Operations Research* 3 (3): 177–88. https://doi.org/10.1287/moor.3.3.177.
Parker, R. Gary and Ronald L. Rardin. 1988. *Discrete Optimization.* San Diego: Academic Press.
Roth, Alvin E. 1982. "The Economics of Matching: Stability and Incentives." *Mathematics of Operations Research* 7 (4): 617–28. https://www.jstor.org/stable/3689483.
Rozanov, Mark and Arie Tamir. 2020. "The Nestedness Property of the Convex Ordered Median Location Problem on a Tree." *Discrete Optimization* 36: 100581. https://doi.org/10.1016/j.disopt.2020.100581.
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.
Sniedovich, Moshe. 1980. "Preference Order Stochastic Knapsack Problems: Methodological Issues." *The Journal of the Operational Research Society* 31 (11): 1025–32. https://www.jstor.org/stable/2581283.
Steinberg, E. and M. S. Parks. 1979. "A Preference Order Dynamic Program for a Knapsack Problem with Stochastic Rewards." *The Journal of the Operational Research Society* 30 (2): 141–47. https://www.jstor.org/stable/3009295.
Van Huizen, Thomas and Rob Alessie. 2019. "Risk Aversion and Job Mobility." *Journal of Economic Behavior & Organization* 164: 91–106. https://doi.org/10.1016/j.jebo.2019.01.021.
Vazirani, Vijay. 2001. *Approximation Algorithms.* Berlin: Springer.
Wolsey, Laurence. 1998. *Integer Programming.* New York: John Wiley & Sons.