Sketching with Test Scores
and Submodular Maximization*

Shreyas Sekar†, Milan Vojnovic‡, and Se-Young Yun§

†Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA
‡Microsoft Research, Cambridge, UK
§Los Alamos National Laboratory, Los Alamos, NM, USA

Abstract

We consider the problem of maximizing a submodular set function that can be expressed as the expected value of a function of an n-size collection of independent random variables with given prior distributions. This is a combinatorial optimization problem that arises in many applications, including the team selection problem that arises in the context of online labour platforms. We consider a class of approximation algorithms that are restricted to use some statistics of the prior distributions to estimate the outputs of value oracle calls, which can be interpreted as test scores. We establish that the existence of test scores that provide an approximation guarantee is equivalent to that a special type of test scores, so called replication test scores, provide this approximation guarantee. We also identify sufficient conditions for replication test scores to provide this approximation guarantee. These sufficient conditions are shown to hold for a large number of models of production, including a monotone concave function of total production, best-shot, and constant elasticity of substitution production function. We also study a more general submodular welfare maximization problem, which accommodates the case of multiple projects, and derive an \( \Omega(1/\log(k)) \)-approximation, algorithm where \( k \) is the maximum number of participants in a project. We evaluate the quality of approximation provided by test scores in a case study using data from a popular online labour platform for software development.

1 Introduction

Submodular functions are a theoretically significant class of set functions that capture the property of diminishing returns, which occurs naturally in a plethora of real-world scenarios. Owing to their far-reaching applications, problems involving submodular functions have been the subject of a formidable body of work: indeed, the problem studied in this paper, that of ‘maximizing a submodular function subject to size or matroid constraints’ has served as the crucial ingredient for problems in diverse fields such as economics, computer vision, and social sciences, e.g., [STK16, KKT15, KO11].

Formally, a submodular function \( u : 2^N \to R \) satisfies diminishing marginal returns, i.e., \( u(S \cup \{i\}) - u(S) \leq u(T \cup \{i\}) - u(T) \) if \( T \subseteq S \). In this work, we study the problem of maximizing such functions subject to a cardinality constraint, referred to as the submodular function maximization problem:

\[
\text{maximize } u(S) \text{ subject to } |S| = k.
\]
In addition, we also look at a strict generalization of the above problem, that of maximizing a submodular function subject to a partition matroid constraint, concisely referred to as the **Submodular Welfare Maximization** problem:

$$\text{maximize} \sum_{j=1}^{m} u_j(S_j) \text{ subject to } S_j \subseteq N, \ |S_j| = k_j \text{ and } S_j \cap S_l = \emptyset, \forall j, l.$$  \hspace{1cm} (2)

**Team Selection and Submodularity arising from Stochastic Behavior.** Although submodular maximization has been studied previously, our interest in the above problems stems from the domain of team selection. The formation of high performance teams has been one of the central topics of study in organization science, where submodularity corresponds naturally to coordination losses or the decrease in productivity that occurs when an individual works within a group, a phenomenon known as *social loafing* [KM86, LWH79]. More recently, team formation has received much attention in the context of online labour platforms; for example, in the context of crowdsourcing platforms where solutions to tasks are derived by aggregating inputs from multiple online workers.

A key issue is the problem of team selection for solving one or more tasks: given a set of individuals \(N = \{1, 2, \ldots, n\}\), the actual productivity of a team of agents \(S \subseteq N\) towards solving task \(j\) is given by \(u_j(S)\). In a departure from tradition, we consider a class of submodular functions motivated strongly by *stochastic models of team production*. In this framework, the productivity function \(u(S)\) is defined as the expected value of a given mapping of individual performances to a team performance output, i.e., \(E[g(\{X_i : i \in S\})]\), where the individual performances \((X_i)\) are independent random variables with given distribution functions. Such a model of team performance was recently introduced by Kleinberg and Raghu [KR15], and is in the spirit of team performance according to a generalized Thurstone model [Thu27], e.g., used by popular rating systems such as TrueSkill [GMH07]. Under the given setting, the submodular maximization objectives (1) and (2) correspond to the problem of forming teams of individuals to tackle one and multiple tasks respectively with the objective of maximizing the total productivity, in general, an NP-Hard problem.

**Local Test Scores for Global Optimization.** In this paper, we take a unique approach towards submodular maximization by focusing on test-score algorithms, a framework where each individual is assigned a score or a weight, and teams are comprised of the agents with the largest test scores. Scores are computed for each individual by performing a test of some kind, e.g., this could be an interview for a job applicant, a screening survey in an online labour platform such as Upwork or TopCoder, or an admission test such as SAT or GRE used for college or graduate school admissions. In a stochastic model of production, these scores summarize some ‘key properties’ of the distribution that controls each individual’s actual performance.

Most if not all of the algorithms for submodular maximization are inextricably tied to the presence of oracles capable of evaluating the exact function value for any given input set. In contrast, the prospect of "previewing the performance" (estimating \(u(S)\)) of different combinations of individuals before even selecting them seems daunting or prohibitively expensive. Given this constraint, our work can be viewed as a concrete first step towards weaning submodular maximization off estimation oracles. In [KR15], the authors present a test-score algorithm that provides a constant-factor approximation for Problem (1) for a particular submodular function (top-\(m\)). In this work, we leverage a more natural type of test scores – replication scores, and show that it is a viable candidate for general approximation algorithms for both Problems (1) and (2), for a broad catalogue of submodular functions.

**Summary of Contributions.** We provide necessary and sufficient conditions for test-score algorithms to yield good approximation factors. For Problem (1), the existence of test scores that guarantee a
Constant-factor approximation is equivalent to the existence of replication scores, implying that in the search for good test scores that guarantee a constant-factor approximation, it suffices to restrict attention to replication test scores. These scores have a very natural definition: for a given function, the replication test score of any individual is defined as the expected team production output of a team consisting of independent replicas of the given individual. Our centerpiece result is a 1/7.328-approximation test-score algorithm (which uses replication test scores) for Problem (1), under certain sufficient conditions, which we verify for many naturally occurring submodular functions including best-shot and constant elasticity of substitution production functions; see Appendix A and B. As an aside, for the latter class of functions, we established lower and upper bounds for approximations provided by two natural definitions of test scores, namely mean test scores and quantile test scores, which imply that these test scores cannot guarantee a constant-factor approximation; see Appendix E. Finally, for Problem (2), we prove that under the same sufficient conditions as mentioned earlier, the replication test scores yield a $\Omega(1/\log(k))$-approximation algorithm, where $k$ is the maximum number of agents assigned to a project, by proving that these test scores can approximate the submodular function everywhere. In the process, we also make inroads on the important theoretical problem of sketching submodular functions. We present average-case performance of using replication test scores in a canonical example of best-shot production, using data about individual performances as observed in a popular platform for software development.

Related Work. Due to the interdisciplinary approach taken in this paper, our model and results bear superficial similarities to other work in disparate fields. The overarching difference between this work and all of the papers cited here is that our algorithms do not depend on evaluating the submodular function (or its marginal increments) $u(S)$ for different subsets of $2^N$. As mentioned above, this distinction plays an important role in team formation settings where one cannot preview $u(S)$ before hiring the users in $S$. Our work could be viewed as extreme case of a recent class algorithms that aim to maximize set functions using a smaller number of oracle queries [BMKK14, OY15]. The techniques in this work are partly inspired by the theoretical work on function sketching [GHIM09, BH11], and their application to optimization problems [IB13]. While the $\Omega(1/\sqrt{n})$ sketch of Goemans et al. [GHIM09] for general submodular functions does apply to our setting, we are able to provide refined bounds (the logarithmic bound of Theorem 3) for a special class of well-motivated utility functions that cannot be captured by existing frameworks such as curvature [SVW15]. Our approach is also similar in spirit to [IB13], where upper and lower bounds in terms of so-called surrogate functions were used for submodular optimization; the novelty in the present work stems from our usage of test scores for function approximation, which are conceptually similar to juntas [FV14]. We believe that intuitive and natural interpretation for test score make them an appealing candidate for other problems as well. Finally, we remark that the stochastic model of team production considered here is uniquely motivated by team formation [KR15] and is only loosely related to other applications where the submodular function value is determined by some individual probabilistic behavior [GHK15, KKT15].

Outline of the Paper. The paper is structured as follows. Section 2 introduces notation, problem definition, and a catalogue of examples of production functions used as running examples throughout the paper. Section 3 introduces test-score algorithms and presents results on their approximation guarantees. Section 4 contains our experimental results. Finally, we conclude in Section 5.
2 Model and Preliminary Results

In this section, we formally define our problem along with the stochastic model of agent behavior, and how it determines the productivity function. Following this, we present a fundamental but somewhat abstract framework for optimizing set functions using the theoretically rich concept of “function sketching”. As we show in future sections, this framework will act as the bridge connecting submodular maximization to test scores.

2.1 Submodular Function and Submodular Welfare Maximization

Given a ground set of agents or elements, \( N = \{1, 2, \ldots, n\} \), a set function \( u : 2^N \to \mathbb{R}_+ \) that returns a positive real-value for any subset \( S \subseteq N \), and an integer \( 1 \leq k \leq n \), the objective of the submodular function maximization problem is to find a set \( S^* \subseteq N \) that maximizes \( u(S) \) over all \( S \subseteq N \) of cardinality \( k \). We also refer to this as the Single Project Assignment problem as it corresponds to selecting a team to tackle a single task specified by the utility function \( u \).

Moreover, we also consider a strict generalization of the above problem, the welfare maximization problem, where we are provided a set of \( m \) utility functions \((u_1, u_2, \ldots, u_m)\) and the concomitant cardinality constraints \((k_1, k_2, \ldots, k_m)\) with the objective of maximizing the social welfare or the aggregate utilities. Formally, the submodular welfare maximization problem asks for an assignment \( S = (S_1, S_2, \ldots, S_m) \) maximizing \( u(S) = \sum_{j=1}^{m} u_j(S_j) \) that also satisfies (i) \(|S_j| = k_j \) for all \( j \); (ii) \( S_j \cap S_l = \emptyset \) for all \( j \neq l \). The problem is known as the Project Assignment problem as it models a setting with \( m \) different tasks or projects and the goal is to assign some number agents to each project. We denote with \( M = \{1, 2, \ldots, m\} \) the set of projects. For the rest of this work, we assume that all of the utility functions \( u_j \) are non-negative, monotonically increasing, and submodular with \( u_j(\emptyset) = 0 \). Under these assumptions, the goal is to solve a combinatorial optimization problem of maximizing a submodular function subject to certain cardinality constraints; this is known to be an NP-Hard problem even for single project assignment with success-probability functions \([KOTT]\). We are primarily interested in approximation algorithms for both of these problems: for \( c \leq 1 \), a \( c \)-approximation algorithm always returns a solution whose utility is at least a factor \( c \) of the optimum.

2.2 A Class of Set Functions derived from Expectations

Suppose that each project \( j \) is associated with a collection of functions \( g_j = (g_j^k : \mathbb{R}^k \to \mathbb{R}_+)_{k=1,\ldots,n} \) and each agent \( i \) is associated with a cumulative distribution function \( F_{i,j} \) for project \( j \). We consider set functions of the following form, for every \( S \subseteq N \) such that \(|S| = k\),

\[
\begin{align*}
u_j(S) &= \mathbb{E}[g_j^k(\{X_{i,j} : i \in S\})] \\
&= \text{(3)}
\end{align*}
\]

where \( X_{i,j} \) are independent random variables with distributions \( F_{i,j} \).

We refer to \((G, F)\) as a stochastic productivity system, where \( G = (g_j, j \in M) \) and \( F = (F_{i,j}, i \in N, j \in M) \). For the single project assignment problem, \( G = g \) and \( F = (F_i, i \in N) \).

2.3 Theoretical Framework: Approximation of a Set Function by Sketches

The key mathematical tool that enables all of the approximation algorithms in this paper is the abstract notion of a sketch: the approximation of a potentially complex set function using simpler poly-time computable upper and lower bound functions. In what follows, we describe these sketches at a high level of generality, showing what properties allow us to leverage these sketches to obtain good algorithms, and
in Section 3.1 we expand on this notion in the context of a stochastic productivity system, providing a link to the concept of a test score.

**Definition 1.** A pair of set functions \((v, \bar{v})\) is said to be a \((p, q)\)-sketch of a set function \(u\), if the following condition holds:

\[
pv(S) \leq u(S) \leq q\bar{v}(S), \text{ for all } S \subseteq N.
\]

**Intuition:** Although the above definition is quite general, and subsumes many trivial sketches (for e.g, \(v = 0, \bar{v} = \infty\)), it is reasonable to expect practically useful sketches to satisfy a few fundamental properties

\((P1)\) Faced with a set function whose description may be exponential in \(n\), \(v, \bar{v}\) must be polynomially expressible in a computationally useful sense, and \((P2)\) \(v\) and \(\bar{v}\) must be sufficiently close to each other at points of interest for the sketch to be meaningful. Our first black-box result provides sufficient conditions on the sketches to obtain an approximation algorithm for the single project assignment problem.

**Lemma 1.** Suppose that \((v, \bar{v})\) is a \((p, q)\)-sketch of a submodular function \(u\), which for any \(S^* \subseteq \arg \max_{S:|S|=k} v(S)\) satisfies: for every \(S \subseteq N\) of size \(k\) that is completely disjoint from \(S^*\), \(v(S) \leq \bar{v}(S^*)\). Then, \(u(S^*) \geq \frac{p}{q+p}u(OPT)\), where \(u(OPT)\) is the optimum value for cardinality \(k\).

Moving on, we now focus on the special class of sketches where \(v = \bar{v} = v\): such a sketch indicates that \(v\) is truly an excellent global approximation for \(u\) at all points. We then show that good algorithms for optimizing \(v\) could be leveraged to obtain approximation algorithms for the submodular welfare maximization problem.

**Lemma 2.** Consider an instance of the submodular welfare maximization problem with input \((u_1, u_2, \ldots, u_m)\) and cardinality constraints \((k_1, k_2, \ldots, k_m)\) with optimum solution \(OPT\). Suppose that for each \(j\), \((v_j, v_j)\) is a \((p, q)\)-sketch of \(u_j\), and that \((S_1, S_2, \ldots, S_m)\) is an \(\alpha\)-approximation to the submodular welfare maximization problem with input \((v_1, v_2, \ldots, v_m)\) and the same cardinality constraints. Then,

\[
\sum_{j=1}^{m} u_j(S_j) \geq \alpha \frac{p}{q} u(OPT).
\]

**3 Approximation Guarantees of Test-Score Algorithms**

In this section, we introduce a class of algorithms that we refer to as test-score algorithms, and then study conditions under which such algorithms can provide approximate solutions for Submodular Function Maximization problem (1) and Submodular Welfare Maximization problem (2).

**3.1 Test-Score Algorithms**

We consider a class of algorithms, we refer to as test-score algorithms, which are defined for the problem of maximizing a submodular welfare function with the utility functions according to a stochastic productivity system \((G, F)\). This class of algorithms is defined by restricting utility value oracle calls to approximate answers that can be computed in polynomial-time using only as input \(A_{G,F}\). Specifically, we consider inputs of the form \(A_{G,F} = (a_{i,j}^k)\), where \(a_{i,j}^k\) is a real-valued test score. Intuitively, \(a_{i,j}^k\) is a parameter that represents the ability of agent \(i\) as a member of a team of size \(k\) working on project \(j\). For the special case of submodular function maximization, we write \(A_{G,F} = (a_i^k)\). In particular, we may think of \(a_{i,j}^k\) as of a
parameter that is computed for given input \( g_j, k, \) and \( F_{i,j} \). For example, a test score may defined to be indicative of an agent’s individual performance; for instance, indicative of his or her typical performance by using a mean test score \( a_{i,j}^k = E[X_{i,j}] \), or indicative of his or her maximum performance by using a quantile test score \( a_{i,j}^k = E[X_{i,j} | F_{i,j}(X_{i,j}) \geq \theta] \), for a suitable value of parameter \( \theta \in [0, 1] \). On may come up with various definitions of test scores; the challenge is to identify conditions under which there exist good test scores for which a test score algorithm can guarantee an approximate solution to the given submodular welfare maximization problem.

### 3.2 Submodular Function Maximization

In this section we consider the problem of computing an approximate solution to Submodular Function Maximization problem (1) using a test-score algorithm. Our approach utilizes the theoretical framework of approximating a set function by a sketch, which we introduced in Section 2.3 in particular, it relies on finding a sketch \((\bar{\nu}, \bar{\upsilon})\) for a set function \(u\) as per Definition [1].

We introduce the concept of good test scores, which is defined as follows.

**Definition 2** (good test scores). Suppose that a set function \(u\) is according to a stochastic productivity system \((G, F)\). For any test scores \(A_{G,F} = (a_{i}^{k})\), let \((\nu^{*}, \bar{\upsilon}^{*})\) be a pair of set functions that for any \(S \subseteq N\) such that \(|S| = k\) are defined by

\[
\nu^{*}(S) = \min\{a_{i}^{k} : i \in S\} \text{ and } \bar{\upsilon}^{*}(S) = \max\{a_{i}^{k} : i \in S\}.
\]

We say that \(A_{G,F}\) are \((p,q)\)-good test scores if \((\nu^{*}, \bar{\upsilon}^{*})\) is a \((p,q)\)-sketch of the set function \(u\). Furthermore, if \(p/q\) is a constant, we say that \(A_{G,F}\) are good test scores.

The existence of \((p,q)\)-good test scores ensures the existence of a test-score algorithm that yields an approximate solution to the submodular function maximization problem as stated more precisely in the following corollary.

**Corollary 1.** Suppose that a set function \(u\) is according to a stochastic productivity system \((G, F)\), and that \(A_{G,F} = (a_{i}^{k})\) are \((p,q)\)-good test scores. Then, for any given cardinality \(k\), a greedy algorithm that outputs set \(S^{*}\) that contains \(k\) distinct elements from \(N\) with largest test scores values \(a_{1}^{k}, a_{2}^{k}, \ldots, a_{n}^{k}\), has the following guarantee:

\[
u(S^{*}) \geq \frac{p}{q + p} u(OPT).
\]

This is a corollary of Lemma [1] which is readily checked by verifying that the conditions of the lemma are satisfied by the given choice of the set functions \(\nu^{*}\) and \(\bar{\upsilon}^{*}\). Corollary [1] implies that for any good test scores, i.e. such that \((\nu^{*}, \bar{\upsilon}^{*})\) is a \((p,q)\)-sketch of the set function \(u\) with \(p/q\) being a constant, a natural greedy algorithm using these test scores yields a constant-factor approximation to the submodular function maximization problem.

We next introduce a special kind of test scores, we refer to as replication test scores, which we will show to be special in the sense that whenever for a set function according to a stochastic productivity system good test scores exist, then replication test scores are good test scores. The replication scores are formally defined as follows.

**Definition 3** (replication test scores). For a stochastic productivity system \((G, F)\), we refer to replication test scores \(A_{G,F} = (a_{i}^{k})\) defined as follows:

\[
a_{i}^{k} = E[g(\{X_{i}^{(1)}, X_{i}^{(2)}, \ldots, X_{i}^{(k)}\})]
\]
where $X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(k)}$ are independent random variables with distribution $F_i$.

Intuitively, for a utility of production function $g$ and a team size $k$, an agent’s replication test score is defined as the expected utility of production of a hypothetical team that consists of $k$ independent replica of this agent. Such a test score reflects the ability of an agent as a member of a team of given size for assumed utility of production function.

The fact that replication test scores are a special kind of test scores, in the sense as already alluded before, is formally established in the following theorem.

**Theorem 1.** Suppose that for a set function according to a stochastic productivity system $(G, F)$, there exist $(p, q)$-good test scores. Then, the replication test scores are $(p/q, q/p)$-good test scores.

This theorem implies that for any set function according to a stochastic productivity system, we can check whether there exist good test scores by just checking whether replication test scores are good test scores. If the replication test scores are not good test scores, then there exist no good test scores.

We proceed with showing a set of sufficient conditions that ensure the replication test scores to be good test scores. In Appendix B these sufficient conditions are shown to hold for a wide range of classic models of production systems. We define the following conditions:

(S) $u(S) = \mathbb{E}[g(\{X_i : i \in S\})]$ is a non-negative, monotone submodular set function;

(M) $g(\{x, y\}) - g(\{x\})$ is decreasing in $x$ for every $y \in \mathbb{R}_+$;

(B) $g(g^{-1}(g(\{x_1, \ldots, x_{l-1}\}), x_l)) \leq g(\{x_1, \ldots, x_l\})$, for all $(x_1, \ldots, x_l) \in \mathbb{R}^l$ and $l > 1$ where $g^{-1}(x) = \max\{y \in \mathbb{R}_+ : g(\{y\}) \leq x\}.$

Both conditions (S) and (M) can be seen to impose a diminishing returns property. Condition (B) is somewhat harder to explain in simple intuitive terms. However, in Appendix, this condition is shown to be trivially verified for many examples of utilities of production; in particular, for the best-shot production, it holds trivially with equality.

In the next theorem, we show that conditions (S),(M), and (B) are sufficient for replication test scores to be good test scores, and provides the factors of the sketch approximation.

**Theorem 2.** Suppose that a set function $u$ according to stochastic productivity system $(G, F)$ satisfies conditions (S), (M), and (B) and that $A_{G, F}$ are replication test scores. Then, replication test scores are $(1 - 1/e, 4)$-good test scores.

Furthermore, for the submodular function maximization problem with objective function $u$ and cardinality constraint $k$, selecting $k$ distinct elements in decreasing order of replication test scores $a_{g,1}^k, a_{g,2}^k, \ldots, a_{g,n}^k$ guarantees an $(1 - 1/e)/(5 - 1/e)$-approximation, i.e., $\approx 1/7.328$.

### 3.3 Submodular Welfare Maximization

Having devised a constant-factor test score algorithm for single project assignment, we now tackle the more general multiple project case, where each task could have its own (submodular) utility function as well as a (potentially) different cardinality constraint. Our main contribution in this section is an $\Omega(1/\log(k))$-approximation algorithm also based on replication scores, but set against the backdrop of a stronger sketching result which may be of independent interest: each utility function can be globally approximated by a single test-score function, i.e., a $(v^*, v^*)$-sketch. Note that here $k$ is the largest size constraint on any project.
Theorem 3. Value oracle calls are computed in polynomial time using replication test scores. We will then apply this to a stochastic productivity system that satisfies conditions (S), (M), and (B). Applying Theorem 3, we can approximate utility function \( u \) to a stochastic productivity system that satisfies conditions (S), (M), and (B), and let \( A \) be a permutation of elements \( S \) defined as follows:

\[
\pi_1(S,j) := \arg\max_{i \in S} a_{i,j}^1, \quad \pi_2(S,j) := \arg\max_{i \in S \setminus \{\pi_1(S,j)\}} a_{i,j}^2, \quad \text{and so on.}
\]

Then, for \( S = (S_1, S_2, \ldots, S_m) \) such that \( |S_j| = k_j \) for all \( j \), define

\[
v(S) = \sum_{j=1}^m v_j(S_j) \quad \text{and} \quad v_j(S_j) = \sum_{r=1}^{k_j} a_{\pi_r(S_j),j}^r.
\]

Claim 1. Consider Problem (2) where we are provided with oracle access to a \((p, q)\)-sketch \( (v^*_j, \bar{v}^*_j) \) for each set function \( v_j \) such that the sketch is consistent with Definition 2. Let Alg_1 and Alg_2 be algorithms that maximize \( v^*_j(S) \) and \( \bar{v}^*_j(S) \) respectively over all assignments \( S \). There exist instances where each of these algorithms are suboptimal by a factor \( O(1/\sqrt{n}) \).

A natural starting point is to ask whether the theory of good test scores developed previously for submodular function maximization directly lends itself to the general problem. Given an instance, one could use the characterization in Definition 2 to maximize either \( v^*_j(S) = \sum_{j=1}^m v_{ij}^*(S_j) \) or \( \bar{v}^*_j(S) = \sum_{j=1}^m v_{ij}^*(S_j) \), as defined therein, over all assignments \( S \). Unfortunately, such approaches may lead to highly sub-optimal assignments even for simple instances.

\textbf{Algorithm 1:} Greedy Algorithm for Project Assignment Problem

\begin{verbatim}
Initialize assignment \( S_1 = S_2 = \ldots = S_m = \emptyset \) \( A = \{1, 2, \ldots, n\}, P = \{1, 2, \ldots, m\} \);

\textbf{while} \(|A| > 0\) \textbf{and} \(|P| > 0\) \textbf{do}

\textbf{end}

\end{verbatim}

In a sense, the negative examples are driven by the large gap between the sketch functions \( \min \) and \( \max \) on arbitrary sets. As a toy example, consider \( A = \{x, 0, 0, 0, 0\} \): there is an infinitely large gap between \( \min A \) and \( \max A \) when \( x > 0 \).

Motivated by the above observation, and by Lemma 2, we instead consider a more challenging question: can we approximate utility function using a single test score-based sketch? Our first result establishes that any set function satisfying conditions (S), (M), and (B) has a sketch set function whose value oracle calls are computed in polynomial time using replication test scores. We will then apply this result to derive a test score algorithm for Submodular Welfare Maximization problem (2).

Theorem 3. Suppose that a set function \( u \) is according to stochastic productivity system \((G, F)\) that satisfies conditions (S), (M), and (B), and let \( A_{G,F} \) be replication test scores. Then, there exists a set function \( v \), whose value oracle call for any input set of size \( k \) can be computed in polynomial time using only input \( A_{G,F} \), which is \((1/[2(\log(k) + 1)], 6)\)-sketch of the set function \( u \).

In particular, a sketch function \( v \) that satisfies the statement in Theorem 3 can be defined as follows. For any set \( S \subseteq N \) such that \( |S| = k \), let \( \pi(S,j) = (\pi_1(S,j), \ldots, \pi_k(S,j)) \) be a permutation of elements of \( S \) defined as follows: \( \pi_1(S,j) := \arg\max_{i \in S} a_{i,j}^1, \pi_2(S,j) := \arg\max_{i \in S \setminus \{\pi_1(S,j)\}} a_{i,j}^2, \) and so on. Then, for \( S = (S_1, S_2, \ldots, S_m) \) such that \( |S_j| = k_j \) for all \( j \), define

\[
v(S) = \sum_{j=1}^m v_j(S_j) \quad \text{and} \quad v_j(S_j) = \sum_{r=1}^{k_j} a_{\pi_r(S_j),j}^r.
\]

Armed with the above theorem and sketch, we are ready to tackle the Submodular Welfare Maximization problem (2). Consider any instance of this problem where each utility function is in accordance to a stochastic productivity system that satisfies conditions (S), (M), and (B). Applying Theorem 3 we can approximate utility function \( u(S) \) by sketch function \( v(S) \) in (5). Finally, exploiting the ‘sequential
Figure 1: The expected utility of production by a set of coders $S^*$ of size $k$ selected by using replication test scores versus the optimum expected utility of production: (left) $k = 2$, (middle) $k = 3$, and (right) $k = 4$.

The greedy nature of Equation (5), we use an intuitive algorithm (Algorithm 1) that greedily assigns agents to projects to obtain our final $\Omega(1/\log(k))$-approximately optimal assignment.

In Appendix, we show that this algorithm is at least half as good as the optimal assignment for the project assignment problem with objective function (5). Hence, we have the following theorem:

**Theorem 4.** Suppose that submodular welfare maximization problem (2) is such that the utility set functions are according to stochastic productivity systems that satisfy conditions (S), (M), and (B), and that cardinality constraints are such that $k_j \leq k$, for all $j$. Then, there exists a test-score algorithm that uses replication test scores, which yields an $1/[24(\log(k) + 1)]$-approximation.

### 4 Experimental Results

In this section, we evaluate average-case performance of using replication test scores for a canonical situation in which multiple agents provide independent solutions to a task and the utility is derived only from the best submitted solution. Such a mechanism is commonly used in the context of online labour platforms to increase the quality of production. In particular, we use a dataset from the popular competition-based software development platform TopCoder. This dataset contains information about submitted solutions that allows us to infer distributions of agents’ individual performances, which we use in our evaluation. Overall, our analysis suggests that using replication test scores yields nearly optimal performance.

The dataset that we use contains information about solutions to web software development tasks submitted by coders over a 10-year period, between November 2003 and January 2013. For each solution, our dataset contains a unique identifier of the coder who made this solution, a unique identifier of the task, and the value of a rating score assigned to this solution. These rating scores take values in $[0, 100]$, and are assigned by a rating procedure used by TopCoder. We use these rating scores as indicators of individual performances. Specifically, our dataset contains information for 7,127 solutions, 658 distinct coders, and 2,924 distinct tasks. The average number of solutions submitted by a coder is 10.83; 75 of them submitted at least 20 solutions and 124 of submitted at least 10 solutions. The average number of
solutions submitted for a task is 2.44. The values of rating scores assigned to solutions vary greatly and have the mean value of 87.54.

Our analysis consists of defining prior distributions of individual performances to be equal to empirical distributions of rating scores achieved by individual coders, and then comparing the expected utility of production achieved by the greedy algorithm that uses replication test scores with the optimum expected utility of production. We compute the optimum value by a brute-force search over all possible sets of given cardinality, which is feasible as we restrict to sets of small cardinalities. Specifically, we conduct the following type of experiments. Given a number of coders \( n \) and a team size \( k \), we sample \( n \) coders uniformly at random without replacement from the input set of coders. For this sampled set of coders, we compute the expected utility achieved for a set of \( k \) coders selected from this set according to the greedy algorithm that uses replication test scores, and the value of the optimum utility. For each configuration setup, we run 10,000 such experiments. We run experiments for different values of parameter \( n \); for space reasons, we report results only for \( n = 10 \) as for the other values we observed similar results. We show outcomes of these experiments in Figure 1 for different values of parameter \( k \); we observe that selecting coders by using the greedy algorithm that uses replication test scores yields nearly optimal performance.

5 Conclusion

We have showed how the framework of sketching set functions can be combined with approximating set functions with test scores to derive approximation algorithms for submodular function maximization and submodular welfare maximization. Our result show that a special kind of test scores, replication test scores, have a particular role in sketching set functions with test scores. We established sufficient conditions under which natural greedy algorithms that use replication test scores provide approximate solutions, for both submodular function maximization and submodular welfare maximization problems.

There are several interesting avenues to pursue for future work. One is to better understand what approximations can be achieved by sketches based on test scores for different classes of submodular set functions, going beyond the class of functions identified in this paper. Another one is to consider the use of test scores in stochastic productivity systems where agents have a multitude of skill types and tasks require specific mixtures of skill types. Yet another one is to consider test-score algorithms whose input are noisy estimators of test scores based on observed samples of individual or team performances.
References

[ACMS61] K. J. Arrow, H. B. Chenery, B. S. Minhas, and R. M. Solow. Capital-labor substitution and economic efficiency. Review of Economics and Statistics (The MIT Press), 43(3):225–250, 1961.

[Arm69] Paul S. Armington. A theory of demand for products distinguished by place of production. Staff Papers (International Monetary Fund), 16(1):159–178, 1969.

[BH11] Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011, pages 793–802, 2011.

[BMKK14] Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: massive data summarization on the fly. In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14, New York, NY, USA - August 24 - 27, 2014, pages 671–680, 2014.

[DS77] Avinash K Dixit and Joseph E Stiglitz. Monopolistic Competition and Optimum Product Diversity. American Economic Review, 67(3):297–308, June 1977.

[FV14] Vitaly Feldman and Jan Vondrák. Optimal bounds on approximation of submodular and XOS functions by juntas. In 2014 Information Theory and Applications Workshop, ITA 2014, San Diego, CA, USA, February 9-14, 2014, pages 1–10, 2014.

[GHIM09] Michel X. Goemans, Nicholas J. A. Harvey, Satoru Iwata, and Vahab S. Mirrokni. Approximating submodular functions everywhere. In Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009, pages 535–544, 2009.

[GHK15] Alkis Gotovos, S. Hamed Hassani, and Andreas Krause. Sampling from probabilistic submodular models. In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 1945–1953, 2015.

[GMH07] Thore Graepel, Tom Minka, and Ralf Herbrich. Trueskill(tm): A bayesian skill rating system. Proc. of NIPS ’07, 19:569–576, 2007.

[HLP52] G. H. Hardy, J. E. Littlewood, and G. Pólya. Inequalities. Cambridge University Press, 2 edition, 1952.

[IB13] Rishabh K. Iyer and Jeff A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 2436–2444, 2013.

[KKT15] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. Theory of Computing, 11:105–147, 2015.

[KM86] David A. Kravitz and Barbara Martin. Ringelmann rediscovered: The original article. Journal of Personality and Social Psychology, 50(5):936–941, 1986.
[KO11] Jon M. Kleinberg and Sigal Oren. Mechanisms for (mis)allocating scientific credit. In Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011, pages 529–538, 2011.

[KR15] Jon Kleinberg and Maithra Raghu. Team performance with test scores. In Proc. of ACM EC ’15, pages 511–528. ACM, 2015.

[LLN01] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In Proceedings of the 3rd ACM conference on Electronic Commerce, pages 18–28. ACM, 2001.

[LWH79] Bibb Latané, Kipling Williams, and Stephen Harkins. Many hands make light the work: The causes and consequences of social loafing. Journal of Personality and Social Psychology, 37(6):822–832, 1979.

[McF63] Daniel McFadden. Constant elasticity of substitution production functions. The Review of Economic Studies, 30(2):73–83, 1963.

[OY15] Naoto Ohsaka and Yuichi Yoshida. Monotone k-submodular function maximization with size constraints. In Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 694–702, 2015.

[Sol56] R. M. Solow. A contribution to the theory of economic growth. The Quaterly Journal of Economics, 70:65–94, 1956.

[STK16] Adish Singla, Sebastian Tschiatschek, and Andreas Krause. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., pages 2037–2043, 2016.

[SVW15] Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. In Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015, pages 1134–1148, 2015.

[Thu27] L. L. Thurstone. A law of comparative judgment. Psychological Review, 34(2):273–286, 1927.

[Uza62] Hirofumi Uzawa. Production functions with constant elasticities of substitution. The Review of Economic Studies, 29(4):291–299, 1962.

[Var92] Hal R. Varian. Microeconomic Analysis. W. W. Norton & Company, 3 edition, 1992.
A Catalogue of Submodular Set Functions

In this section, we provide several examples of submodular set functions of the form according to a stochastic productivity system \((G, F)\) where \(G = g\). These set functions are derived from classic models of production that are defined as follows:

1. **Total production**: 
   \[
   g(\{x_i : i \in S\}) = h \left( \sum_{i \in S} x_i \right)
   \]  
   (6)
   
   where \(h : \mathbb{R} \rightarrow \mathbb{R}_+\) is a non-negative monotone increasing function.

2. **Best-shot**: 
   \[
   g(\{x_i : i \in S\}) = \max_{i \in S} x_i.
   \]  
   (7)

3. **Top-\(m\)**: given an integer \(m\) such that \(1 \leq m \leq |S|\), 
   \[
   g(\{x_i : i \in S\}) = \sum_{i=1}^{m} x_{(S,i)}
   \]  
   (8)
   
   where \(x_{(S,1)}, x_{(S,2)}, \ldots, x_{(S,|S|)}\) are the values of \(x_S\) rearranged in decreasing order.

4. **Constant Elasticity of Substitution (CES)**: for given parameter \(p > 0\), 
   \[
   g(\{x_i : i \in S\}) = \left( \sum_{i \in S} x_i^p \right)^{1/p}.
   \]  
   (9)

5. **Success-Probability**: 
   \[
   g(\{x_i : i \in S\}) = 1 - \prod_{i \in S} (1 - h(x_i)),
   \]  
   (10)
   
   where \(h : \mathbb{R} \rightarrow [0, 1]\) is an increasing function.

A production function defined as an increasing function of the total individual investment in a production activity, as given in (6), is a natural model of production. For a concave function \(h\), this production function exhibits a diminishing returns increase property, where the marginal increase of the production output becomes smaller or remains constant, the larger the total investment in the production.

The best-shot production function in (7) defines the production output to be the largest production input invested in the production activity. This type of production is common in online crowdsourcing systems where often multiple solutions are solicited for a task, and eventually only the best solution is selected.

The top-\(m\) production function in (8) is a natural generalization of the best-shot production function, where instead of restricting to selecting only the best solution for a task, a given number of best solutions is selected.

The constant elasticity of substitution or CES production function, defined in (9), was considered in [Sol56] and has been much popularized following [ACMS61]. The CES production function is a textbook example of a production function, e.g., see Chapter 1 in [Var92]. [ACMS61] used a CES production function to describe how capital and labour map to a value of production. [Arm69] used a CES production function as a model of demand for products distinguished by place of production.
used a CES production function as a model of demand for commodities that are substitutes among themselves in a monopolistic market to study optimum product diversity. Several properties of the CES production functions were studied by [Uza62] and [McF63]. The CES production function exhibits constant returns to scale, meaning that scaling the production inputs by a factor results in scaling the production output by the same factor. This is equivalent to saying that the production function is homogenous of degree 1, i.e. \( f(tx_1,tx_2,\ldots,tx_n) = tf(x_1,x_2,\ldots,x_n) \) for all \( t \geq 0 \). The CES production function corresponds to a weighted mean defined in [HLP52] as follows: for given values \( x = (x_1,x_2,\ldots,x_n) \in \mathbb{R}^n \), and fixed parameters \( q_i > 0 \) for \( i = 1,2,\ldots,n \), a weighted mean of \( x \) is given by \( M_p(x) = (\sum_{i=1}^{n} q_i x_i^p) / \beta )^{1/p} \), where \( p \) is any real value except for (i) \( p = 0 \) and (ii) \( p < 0 \) and \( x_i = 0 \) for some \( i \in \{1,2,\ldots,n\} \). The family of CES production functions accommodates different types of productions by a suitable choice of parameter \( p \). The CES production function models a production that is linear in the total production input for the value of parameter \( p = 1 \), and it corresponds to the best-shot production in the limit as the value of parameter \( p \) goes to infinity. The success-probability production function, defined in [HLP52], is often used as a model of tasks for which each individual solution is either good or bad, and it suffices to have at least one good solution for the task to be successfully solved.

The utility of production function is a submodular set function under the following conditions on the production function \( g \) for our given examples. For the total production model, it suffices to assume in addition that function \( g \) is a concave function, i.e. it exhibits a diminishing increase with the value of the total production. The utility of production under either the best-shot, the top-\( m \) or the success-probability production function is a submodular function without making any additional assumptions. The utility of production under the CES production function is a submodular function if and only if \( p \geq 1 \).

### B Verifying Conditions for Some Production Functions

One can easily check that all production functions from our catalogue of examples are non-negative, monotone submodular set function. In this section, therefore, we show that conditions (M) and (B) hold for all production functions in the catalogue of examples in Section 2.

**Total production:** \( g\{x_i : i \in S\} = h(\sum_{i \in S} x_i) \) Condition (M) holds because \( g\{x,y\} - g\{x\} = h(x+y) - h(x) \) and \( h \) has decreasing increments because by assumption it is a concave function. Condition (B) holds because \( g^{-1}(x) = h^{-1}(x) \) where \( h^{-1} \) is the inverse function of \( h \), and hence \( g\{g^{-1}(g\{x_1,\ldots,x_{l-1}\})\},x_l\} = h(h^{-1}(h(x_1 + \cdots + x_{l-1}) + x_l) = h(x_1 + \cdots + x_l) = g\{x_1,\ldots,x_l\}).

**Best-shot:** \( g\{x_i : i \in S\} = \max_{i \in S} x_i \) Note that \( g^{-1}(x) = x \). Condition (M) holds because \( g\{x,y\} - g\{x\} = \max\{x,y\} - x \), which is indeed decreasing in \( x \), for every fixed value \( y \in \mathbb{R}_+ \). Condition (B) holds because \( g\{g^{-1}(g\{x_1,\ldots,x_{l-1}\})\},x_l\} = \max\{\max\{x_1,\ldots,x_{l-1}\},x_l\} = \max\{x_1,\ldots,x_l\} = g\{x_1,\ldots,x_l\}).

**CES:** \( g\{x_i : i \in S\} = (\sum_{i \in S} x_i^p)^{1/p}, \text{ for parameter } p \geq 1 \) Note that \( g^{-1}(x) = x \). Condition (M) holds because \( g\{x,y\} - g\{x\} = (x^p + y^p)^{1/p} - x \), and hence,

\[
\frac{\partial}{\partial x} (f\{x,y\} - f\{x\}) = \left( \frac{x}{(x^p + y^p)^{1/p}} \right)^{p-1} - 1 \leq 0.
\]

Condition (B) holds because \( g\{g^{-1}(g\{x_1,\ldots,x_{l-1}\})\},x_l\} = ((x_1^p + \cdots + x_{l-1}^p)^{1/p})^p + x_l^p = (x_1^p + \cdots + x_l^p)^{1/p} = g\{x_1,\ldots,x_l\} \).
Success-Probability: \( g(\{x_i : i \in S\}) = 1 - \prod_{i \in S}(1 - h(x_i)) \) Condition (M) holds because \( g(\{x, y\}) - g(\{x\}) = h(y)(1 - h(x)) \) and \( h \) is an increasing function. Condition (B) holds because \( g\left(g^{-1}(g(\{x_1, \ldots, x_{l-1}\}), x_l)\right) = 1 - \prod_{i=1}^{l-1}(1 - h(x_i)) = g(\{x_1, \ldots, x_l\}) \).

C Proof of Lemmas

C.1 Proof of Lemma

We first note the following inequalities

\[ u(\text{OPT}) \leq u(S^*) + u(\text{OPT} \setminus S^*) \leq u(S^*) + q\bar{v}(\text{OPT} \setminus S^*). \]

The first inequality comes from the fact that all submodular functions are subadditive and must satisfy \( u(A \cup B) \leq u(A) + u(B) \). The next inequality comes from the sketch upper bound. Now, consider any size \( k \) superset \( (T_k) \) of \( \text{OPT} \setminus S^* \) that is disjoint from \( S^* \). By condition C1, we know that \( \bar{v}(T_k) \leq v(S^*) \). There we get

\[ u(\text{OPT}) \leq u(S^*) + q\bar{v}(S^*) \leq u(S^*) + \frac{q}{p} u(S^*). \]

This completes the proof.

C.2 Proof of Lemma

Suppose that \( S^* \) is the optimum solution to the project assignment problem with the sketched utilities \((v_1, \ldots, v_m)\), and \( u(S) \geq \alpha u(S^*) \). Then,

\[ u(\text{OPT}) = \sum_{j=1}^{m} u_j(\text{OPT}_j) \]

\[ \leq q \sum_{j=1}^{m} v_j(\text{OPT}_j) \]

\[ \leq q \sum_{j=1}^{m} v_j(S^*_j) \quad \text{(since this solution is optimal for sketched utilities)} \]

\[ \leq \frac{1}{\alpha} q \sum_{j=1}^{m} v_j(S_j) \]

\[ \leq \frac{1}{p\alpha} q \sum_{j=1}^{m} u_j(S_j). \]

D Proof of Theorems

D.1 Proof of Theorem

Suppose that for a set function \( u \) according to a stochastic productivity system \((\mathcal{G}, \mathcal{F}), A_{\mathcal{G}, \mathcal{F}} = (a_i^k) \) are \((p, q)\)-good test scores, i.e. for every \( S \subseteq N \) such that \( |S| = k \),

\[ p \min\{a_i^k : i \in S\} \leq u(S) \leq q \max\{a_i^k : i \in S\}. \]  \hspace{1cm} (11)
Let $R_{\mathcal{G}, \mathcal{F}} = (r_i^k)$ be the replication test scores, i.e.

$$r_i^k = E[g((X_i^{(1)}, \ldots, X_i^{(k)}))] = u((i^{(1)}, \ldots, i^{(k)})$$  \hspace{1cm} (12)

where $X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(k)}$ are independent random variables with distribution $F_i$.

By the assumption that $\mathcal{A}_{\mathcal{G}, \mathcal{F}}$ are $(p, q)$-good test scores, we have

$$p a_i^k \leq u((i^{(1)}, \ldots, i^{(k)}) \leq q a_i^k.$$  \hspace{1cm} (13)

From (11), (12), and (13), we have

$$\frac{p}{q} \min\{r_i^k : i \in S\} \leq p \min\{a_i^k : i \in S\} \leq u(S) \leq q \max\{a_i^k : i \in S\} \leq \frac{q}{p} \max\{r_i^k : i \in S\}$$

which implies that replication test scores are $(p/q, q/p)$-good test scores.

\section{D.2 Proof of Theorem 2}

We prove the lower bound and upper bound as follows.

**Proof of the lower bound.** Without loss of generality, consider the set $S = \{1, 2, \ldots, k\}$ and assume that $a_i^k = \min_{i \in S} a_i^k$. We claim that for every $j \in \{1, 2, \ldots, k\}$,

$$u(\{1, 2, \ldots, j\}) \geq 1 - \frac{1}{k} u(\{1, \ldots, j-1\}) + \frac{1}{k} a_1^k.$$

From this, we can use a cascading argument to show that $u(S) \geq (1 - \frac{1}{k}) a_1^k \geq (1 - \frac{1}{e}) a_1^k$. We begin by proving the claim by mathematical induction. Base case $j = 1$: since $u$ is a non-negative, monotone submodular set function, we have

$$u(\{1\}) = \frac{1}{k} \sum_{i=1}^k u(\{1^{(i)}\}) \geq \frac{1}{k} u(\{1^{(1)}\}, \ldots, 1^{(k)}) = \frac{1}{k} a_1^k.$$  \hspace{1cm} (14)

Induction step: suppose that $u(\{1, \ldots, j\}) \geq \frac{j}{2k} a_1^k$ holds for $1 \leq j < k$ and we need to show that it holds that $u(\{1, \ldots, j + 1\}) \geq \frac{j+1}{2k} a_1^k$. Note that

$$\begin{align*}
    u(\{1, \ldots, j + 1\}) &= u(\{1, \ldots, j\}) + [u(\{1, \ldots, j + 1\}) - u(\{1, \ldots, j\})] \\
    &\geq u(\{1, \ldots, j\}) + \frac{1}{k} [u(\{1, \ldots, j + 1\}^{(1)}, \ldots, (j + 1)^{(k)}) - u(\{1, \ldots, j\})] \\
    &\geq u(\{1, \ldots, j\}) + \frac{1}{k} [u((j + 1)^{(1)}, \ldots, (j + 1)^{(k)}) - u(\{1, \ldots, j\})] \\
    &\geq 1 - \frac{1}{k} u(\{1, \ldots, j\}) + \frac{1}{k} a_{j+1}^k \\
    &\geq 1 - \frac{1}{k} u(\{1, \ldots, j\}) + \frac{1}{k} a_j^k.
\end{align*}$$

where (a) and (b) hold by the assumption that $u$ is a non-negative, monotonically increasing, and submodular function, and the penultimate inequality follows by the induction hypothesis. We now proceed with the cascading argument.
\[
\begin{align*}
\mathcal{u}(\{1, \ldots, k\}) & \geq \left(1 - \frac{1}{k}\right) \mathcal{u}(\{1, \ldots, k - 1\}) + \frac{1}{k} a_1^k \\
& \geq \left(1 - \frac{1}{k}\right)^2 \mathcal{u}(\{1, \ldots, k - 2\}) + \left(1 - \frac{1}{k}\right) \frac{a_1^k}{k} + \frac{a_1^k}{k} \\
& \geq \cdots \\
& \geq \frac{a_1^k}{k} \left(\sum_{j=0}^{k-1} \left(1 - \frac{1}{k}\right)^j\right) \\
& \geq a_1^k \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \\
& \geq \left(1 - \frac{1}{e}\right) a_1^k
\end{align*}
\]

For the last step, we use the fact that \((1 - 1/k)^k \leq 1/e\), for any \(k \geq 1\).

**Proof of the upper bound.** Without loss of generality, assume that \(S = \{1, 2, \ldots, k\}\) and \(a_1^k \leq a_2^k \leq \cdots \leq a_k^k\). Let \(i^*\) be an agent such that \(X_{i^*} = x^*\) with probability 1, for \(x^*\) such that \(g(\{x^*\}) = ca_k^k\), for a constant \(c \geq 1\). Since \(u\) is a non-negative, monotone increasing, and submodular function, we have

\[
\begin{align*}
\mathcal{u}(S) & \leq \mathcal{u}(\{i^*\} \cup S) \\
& \leq \mathcal{u}(\{i^*\}) + \sum_{i=1}^{k} (\mathcal{u}(\{i^*\} \cup \{i\}) - \mathcal{u}(\{i^*\})) \\
& = ca_k^k + \sum_{i=1}^{k} (\mathcal{u}(\{i^*\} \cup \{i\}) - \mathcal{u}(\{i^*\})) \\
& \geq \sum_{i=1}^{k} \left(\mathcal{u}(\{i^*\} \cup \{i\}) - \mathcal{u}(\{i^*\})\right).
\end{align*}
\]

Now, note that

\[
\begin{align*}
\mathcal{u}(\{i^*\} \cup \{i\}) - \mathcal{u}(\{i^*\}) &= \mathbb{E}[g(\{x^*\}, X_i) - ca_k^k] \\
& \leq \mathbb{E}\left[g(\{x^*\}, \{X_{i^*}^{(1)}, \ldots, X_{i^*}^{(k-1)}\}) - g(\{X_{i^*}^{(1)}, \ldots, X_{i^*}^{(k-1)}\})g(\{X_{i^*}^{(1)}, \ldots, X_{i^*}^{(k-1)}\}) \leq ca_k^k\right] \\
& \leq \mathbb{E}\left[g(\{X_i^{(1)}, \ldots, X_i^{(k)}\}) - g(\{X_i^{(1)}, \ldots, X_i^{(k-1)}\})g(\{X_{i^*}^{(1)}, \ldots, X_{i^*}^{(k-1)}\}) \leq ca_k^k\right] \\
& \leq \mathbb{E}\left[g(\{X_i^{(1)}, \ldots, X_i^{(k)}\}) - g(\{X_i^{(1)}, \ldots, X_i^{(k-1)}\}) \leq ca_k^k\right] \\
& \leq \mathbb{E}\left[g(\{X_i^{(1)}, \ldots, X_i^{(k)}\}) \leq ca_k^k\right] \\
& \leq \mathbb{E}\left[g(\{X_i^{(1)}, \ldots, X_i^{(k)}\}) \leq ca_k^k\right] \\
& \leq \mathbb{E}\left[g(\{X_i^{(1)}, \ldots, X_i^{(k-1)}\}) \leq ca_k^k\right]
\end{align*}
\]

where (a), (b), and (c) hold by conditions (M), (B), and (S), respectively, and (d) holds because by Markov’s inequality and condition (S)

\[
\begin{align*}
\mathbb{E}[g(\{X_i^{(1)}, \ldots, X_i^{(k)}\})] & \leq \mathbb{E}\left[g(\{X_i^{(1)}, \ldots, X_i^{(k-1)}\}) \leq ca_k^k\right] \\
& \leq \frac{a_k^k}{k} \\
& \leq \left(1 - \frac{1}{e}\right) a_k^k.
\end{align*}
\]
From (16) and (17), we obtain \( u(S) \leq \frac{e^2}{c-1}a_k^k \), which implies Claim 2 when \( c = 2 \).

D.3 Proof of Claim 1

**Negative Example for Alg1**  
Recall that \( g^*_j(S) = \min \{ a_{i,j}^k : i \in S \} \), where \( k_j \) is the cardinality constraint on project \( j \). Consider an example with \( r^2 \) users and \( m = r \) projects with each project having a cardinality constraint of \( k_j = r \) for all \( r \). All of the users are deterministic, \( r \) of the users (call them heavy users), have value 1, i.e., \( X_i = 1 \) with probability 1. The remaining users are all zero valued. Next, the stochastic productivity system is defined such that \( g^k_j(S) = \max \{ x_i : i \in S \} \) for all \( j, k \).

The optimum solution for this instance is when each of the heavy users is assigned to a different project, leading to a social welfare of \( r \). On the contrary, Alg1 returns a solution where all of the heavy users belong to the same project giving only a welfare of \( 1 \).

**Negative Example for Alg2**  
Recall that \( \bar{v}^*_j(S) = \max \{ a_{i,j}^k : i \in S \} \), where \( k_j \) is the cardinality constraint on project \( j \). There are \( 2^r \) users and \( r + 1 \) projects, where project 1 has a cardinality constraint of \( r \), for \( j \geq 2 \), \( k_j = 1 \). All of the users are deterministic once again: one heavy user has a value of \( \sqrt{r} \), \( r - 1 \) medium users have a value of 1, and finally the remaining users are zero valued. Next, the stochastic system is assumed to be such that \( g^k_j(S) = \sum_{i \in S} x_i \) and, for \( j \geq 2 \), \( g^k_j(S) = (1/\sqrt{r}) \max \{ x_i : i \in S \} \).

The optimum solution assigns the all of the users to project 1 giving a welfare of \( O(r) \), whereas Alg2’s solution is to assign the heavy user to project 1 and the medium users spread across giving a welfare of only \( 2\sqrt{r} \). This completes the proof.

D.4 Proof of Theorem 3

Recall our definition of the set permutation and the sketch function. Let \( \varphi_{\mathcal{F}} = (a_i^k) \) be replication test scores. For any set \( S \subseteq N \) such that \( |S| = k \), let \( \pi(S) = (\pi_1(S), \pi_2(S), \ldots, \pi_k(S)) \) be a permutation of elements of \( S \) defined as follows: \( \pi_1(S) := \arg \max_{i \in S} a_i^1 \), \( \pi_2(S) = \arg \max_{i \in S \setminus \{ \pi_1(S) \}} a_i^2 \), and so on. Informally, \( \pi_j(S) \) is the element with the largest value of \( a_i^j \) in the set \( S \) minus the ones in \( \{ \pi_1(S), \pi_2(S), \ldots, \pi_{j-1}(S) \} \).

Let \( v \) be a set function, which for any \( S \subseteq N \) such that \( |S| = k \) is defined by

\[
v(S) = \sum_{\pi = 1}^{k} \frac{1}{r} a_{\pi_\ell}(S). \tag{18}
\]

We will establish the following lower and upper bounds: for every \( S \subseteq N \),

\[
u(S) \geq \frac{1}{2(\log(k) + 1)} v(S) \tag{19}
\]

and

\[
u(S) \leq 6v(S). \tag{20}
\]

**Proof of Lower Bound**  
We first establish several lemmas and then prove the lower bound in (19). Suppose that \( S \) is of cardinality \( k \) and suppose that \( \ell := \arg \max_j a_{\pi_j}(S) \), i.e., for every \( j, a_{\pi_j}(S) \geq a_{\pi_j}(S) \). We begin with the following simple property that holds for all \( \pi_j(S) \).
Lemma 3. Consider any \( \pi_j(S) \in S \) as per the permutation described above. We have the following relation:

\[
\frac{a^t_{\pi_j(S)}}{t} \geq \frac{a^j_{\pi_j(S)}}{j}, \quad \text{if} \quad t \leq j, \quad \text{and} \quad \frac{a^t_{\pi_j(S)}}{t} \leq \frac{a^j_{\pi_j(S)}}{j}, \quad \text{if} \quad t \geq j.
\]

This follows from the concavity of \( a^t_{\pi_j(S)} \) in the variable \( t \). Before showing the lower bound on \( u(S) \), let us first show an upper bound on \( a^\ell_{\pi_\ell(S)} \), that is the largest of the \( a^j_{\pi_j(S)} \)'s.

We next establish the following lemma:

Lemma 4. The following relation holds:

\[
a^\ell_{\pi_\ell(S)} \leq \frac{2}{\ell} \sum_{j=1}^{\ell} a^j_{\pi_j(S)}.
\]

Proof. By definition, \( a^1_{\pi_1(S)} \geq a^1_{\pi_j(S)} \) for all \( \pi_j(S) \in S \). Applying this to \( \pi_\ell(S) \), we have

\[
a^1_{\pi_1(S)} \geq a^1_{\pi_\ell(S)} \geq \frac{a^\ell_{\pi_\ell(S)}}{\ell},
\]

where the second inequality follows from Lemma 3. Similarly, for \( a^2_{\pi_2(S)} \), we know that \( a^2_{\pi_2(S)}/2 \geq a^2_{\pi_\ell(S)}/2 \geq a^\ell_{\pi_\ell(S)}/\ell \), and so on for \( \pi_3(S), \pi_4(S), \ldots \pi_{\ell-1}(S) \). Adding up these inequalities (after transposing the denominator) along with the trivial inequality \( a^\ell_{\pi_\ell(S)}/\ell \geq a^\ell_{\pi_\ell(S)}/\ell \), we have

\[
\sum_{j=1}^{\ell} a^j_{\pi_j(S)} \geq \frac{a^\ell_{\pi_\ell(S)}}{\ell} \sum_{j=1}^{\ell} j \geq \frac{a^\ell_{\pi_\ell(S)}}{\ell} \frac{\ell(\ell + 1)}{2},
\]

which implies

\[
\frac{2}{\ell} \sum_{j=1}^{\ell} a^j_{\pi_j(S)} \geq \frac{2}{\ell + 1} \sum_{j=1}^{\ell} a^j_{\pi_j(S)} \geq a^\ell_{\pi_\ell(S)}.
\]

We now prove the lower bound in (19), which we restate as follows:

\[
u(S) \geq \frac{1}{2(\log(k) + 1)} \left( a^1_{\pi_1(S)} + \frac{a^2_{\pi_2(S)}}{2} + \ldots + \frac{a^k_{\pi_k(S)}}{k} \right).
\]

We begin by making the following simple claim that we prove by induction:

\[
u(S) \geq \frac{1}{\ell} \sum_{j=1}^{\ell} a^j_{\pi_j(S)}.
\]

The inductive statement is as follows, for every \( 1 \leq j \leq \ell \), \( u(\pi_1(S), \ldots, \pi_j(S)) \geq \frac{1}{j} \sum_{t=1}^{j} a^t_{\pi_t(S)} \).

The statement is trivially true for \( j = 1 \).

Assume it is true up to \( j - 1 \) and consider,
where in the last inequality we use the fact that

\[
\text{Proof of Upper Bound}
\]

We prove the upper bound stated in (20). The proof of the upper bound is analogously to the upper bound proof for good test scores. Analogous (but slightly different) to that proof, consider a deterministic quantity \( s^* \) such that \( u(\{s^*\}) = 2a^*_{\pi(S)} \). By definition, we know that for every \( j \), \( a^j_{\pi_j(S)} \leq a^*_{\pi(S)} \). Moreover, we can upper bound \( u(S) \) as follows,

\[
u(S) \leq u(S \cup \{s^*\}) \leq u(\{s^*\}) + \sum_{j=1}^{k} u(\{s^*, \pi_j(S)\}) - u(\{s^*\}).
\]

From the inductive hypothesis, we know that \( u(\{\pi_1(S), \ldots, \pi_j(S)\}) \geq \frac{1}{j-1} \sum_{t=1}^{j-1} a^t_{\pi(S)} \), so we add \( u(\{\pi_1(S), \ldots, \pi_j(S)\}) \) to both sides of the above equation and get

\[
u(\{\pi_1(S), \ldots, \pi_j(S)\}) \geq \frac{a^j_{\pi_j(S)}}{j} + \frac{j-1}{j} u(\{\pi_1(S), \ldots, \pi_j(S)\}) \geq \frac{1}{j} \sum_{t=1}^{j} a^t_{\pi(S)}.
\]

This proves the claim that \( u(S) \geq \frac{1}{j} \sum_{j=1}^{\ell} a^j_{\pi_j(S)} \). Now, we apply Lemma 4 to obtain that \( u(S) \geq a^\ell_{\pi(S)}/2 \). Finally, we conclude the lower bound as follows

\[
u(S) \geq \frac{1}{2} a^\ell_{\pi(S)} = \frac{a^\ell_{\pi(S)}}{2} \left( 1 + \frac{1}{2} + \ldots + \frac{1}{k} \right) \geq \frac{a^1_{\pi_1(S)} + a^2_{\pi_2(S)} + \ldots + a^k_{\pi_k(S)}}{2 \log(k) + 1}
\]

where in the last inequality we use the fact that \( a^\ell_{\pi_j(S)} \geq a^j_{\pi_j(S)} \) for all \( j \), and the fact \( 1 + 1/2 + \ldots + 1/k \leq \log(k) + 1 \), for all \( k \geq 1 \).

**Proof of Upper Bound**  We prove the upper bound stated in (20). The proof of the upper bound is almost identically to the upper bound proof for good test scores. Analogous (but slightly different) to that proof, consider a deterministic quantity \( s^* \) such that \( u(\{s^*\}) = 2a^*_{\pi(S)} \). By definition, we know that for every \( j \), \( a^j_{\pi_j(S)} \leq a^*_{\pi(S)} \). Moreover, we can upper bound \( u(S) \) as follows,
Now,
\[ u(\{s^*, \pi_j(S)\}) - u(\{s^*\}) \]
\[ = \mathbb{E}[g(\{x^*, X_{\pi_j(S)}\}) - 2a_{\pi_j(S)}^\ell] \]
\[ \leq \mathbb{E}[g(\{g^{-1}(X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}), X_{\pi_j(S)}^{(j)}) - g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\})| \]
\[ g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\}) \leq 2a_{\pi_j(S)}^\ell] \]
\[ \leq \mathbb{E}[g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\}) - g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\})] \]
\[ \Pr[g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\}) \leq 2a_{\pi_j(S)}^\ell] \]
\[ \leq \frac{a_j^j}{\pi_j(S)/j}. \]

The denominator can be bounded using Markov’s inequality:
\[ \Pr \left[ g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\}) \geq 2a_{\pi_j(S)}^\ell \right] \leq \frac{\mathbb{E}[g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\})]}{2a_{\pi_j(S)}^\ell} \leq \frac{a_{\pi_j(S)}}{2a_{\pi_j(S)}^\ell}. \]

Recall that \( E[g(\{X_{\pi_j(S)}^{(1)}, \ldots, X_{\pi_j(S)}^{(j-1)}\})] = a_{\pi_j(S)}^{j-1} \leq a_j^j \leq a_{\pi_j(S)}^\ell \).

In conclusion, we have that \( u(\{s^*, \pi_j(S)\}) - u(\{s^*\}) \leq 2a_j^j/\ell \). Summing these up, gives us, \( u(S) \leq 2a_{\pi_j(S)}^\ell + 2(a_{\pi_1(S)} + a_{\pi_2(S)}/2 + \cdots + a_{\pi_k(S)}/k) \). Applying Lemma 4 to \( a_{\pi_j(S)}^\ell \), we obtain that
\[ u(S) \leq 6 \left( a_{\pi_1(S)} + \frac{a_{\pi_2(S)}}{2} + \cdots + \frac{a_{\pi_k(S)}}{k} \right). \]

### D.5 Proof of Theorem 4

Since \( v(S) \) is not a submodular set function, we cannot use directly the existing results for the submodular welfare maximization. Instead of using \( v \), we first introduce another set function \( \bar{v} \) such that \( \max_S \bar{v}(S) \geq \max_S v(S) \) and \( \bar{v}(S(t)) = v(S(t)) \) for all \( t \geq 1 \) where \( S(t) = (S_1(t), \ldots, S_m(t)) \) be the assignment by the greedy algorithm at time \( t \). We then show that the greedy algorithm is an 1/2-approximation algorithm for \( \max_S \bar{v}(S) \).

Let \( \bar{v}(S) = \sum_{j=1}^m \sum_{r=1}^{S_j} \frac{1}{r} a_{r,j}^r \) where \( S = (S_1, \ldots, S_m) \) and each \( S_j = (e_{1,j}, \ldots, e_{|S_j|,j}) \) has an order. Let \( \mathcal{O}(S) \) be the set of all permutation functions for \( S \). Then,
\[ \max_{\phi \in \mathcal{O}(S)} \bar{v}(\phi(S)) \geq v(S), \tag{21} \]
which holds because \( v(S) \) assumes for each \( S_j \) an order of elements according to \( \pi \).

Then, from Lemma 3 the greedy assignment rule and \( \bar{v} \) has the following properties:
Moreover, this bound is tight.

Theorem 5. Suppose that a set function \( u \) is according to stochastic productivity system with a CES production function with parameter \( p \geq 1 \). Then, for every given cardinality \( k \geq 1 \), let \( M \) be a set of \( k \) elements in \( N \) with highest mean test scores. Then, we have

\[
u(M) \geq \frac{1}{k^{1-1/p}} \nu(\text{OPT}).
\]

Moreover, this bound is tight.

Proof. Without loss of generality, assume that \( \mathbf{E}[X_1] \geq \mathbf{E}[X_2] \geq \cdots \geq \mathbf{E}[X_n] \). Let \( S = \{i_1, i_2, \ldots, i_k\} \) be an arbitrary set. Then, we have

\[
u(S) = \mathbf{E}[g(\{X_i : i \in S\})]
= \mathbf{E}[(g(\{X_i : i \in S\}) - g(\{X_i : i \in S \setminus \{i_k\}\})) + (g(\{X_i : i \in S \setminus \{i_k\}\}) - g(\{X_i : i \in S \setminus \{i_{k-1}, i_k\}\})) + \cdots + (g(\{X_{i_1}\}) - g(\emptyset))] \\
\leq \mathbf{E}[g(\{X_{i_k}\}) + g(\{X_{i_{k-1}}\}) + \cdots + g(\{X_{i_1}\})]
= \sum_{i \in S} \mathbf{E}[X_i]
\leq \sum_{i=1}^{k} \mathbf{E}[X_i]
\]

where the first inequality follows by the submodularity of function \( u(S) \), the second inequality is by the assumption that individuals are enumerated in decreasing order of the mean test scores.
Now, observe that for every \((x_1, x_2, \ldots, x_k) \in \mathbb{R}_+^k\),
\[
\frac{1}{k} \sum_{i=1}^k x_i = \frac{1}{k} \sum_{i=1}^k (x_i^p)^{1/p} \leq \left( \frac{1}{k} \sum_{i=1}^k x_i^p \right)^{1/p}
\]
which is by Jensen’s inequality. Hence,
\[
\sum_{i=1}^k \mathbb{E}[X_i] \leq k^{1-1/p} \mathbb{E} \left[ \left( \sum_{i=1}^k X_i^p \right)^{1/p} \right].
\]
Combining with (22), it follows that for every \(S \subseteq N\) such that \(|S| = k\),
\[
\mathbb{E} \left[ \left( \sum_{i=1}^k X_i^p \right)^{1/p} \right] \geq \frac{1}{k^{1-1/p}} \mathbb{E} \left[ \left( \sum_{i \in S} X_i^p \right)^{1/p} \right].
\]

The tightness can be established as follows. Let \(N\) consists of two subsets of individuals \(M\) and \(R\), where \(M\) consists of \(k\) individuals whose each individual performance is of value \(1 + \epsilon\) with probability \(1\), for a parameter \(\epsilon > 0\), and \(R\) consists of \(k\) individuals whose each individual performance is of value \(a\) with probability \(1/a\) and of value 0 otherwise, for parameter \(a \geq 1\). Then, we note that
\[
u(M) = k^{1/p}(1 + \epsilon)
\]
and
\[
u(OPT) \geq \nu(R) = \mathbb{E} \left[ \left( \sum_{i \in R} X_i^p \right)^{1/p} \right] 
\geq a \mathbb{P} \left[ \sum_{i \in R} X_i > 0 \right] 
= a \left( 1 - \left( 1 - \frac{1}{a} \right)^k \right) 
\geq a \left( 1 - e^{-k/a} \right).
\]
Hence, it follows that
\[
\frac{\nu(M)}{\nu(OPT)} \leq (1 + \epsilon) \frac{1}{k^{1-1/p}} \frac{k/a}{1 - e^{-k/a}}.
\]
The tightness claim follows by taking \(a\) such that \(k = o(a)\), so that \((k/a)/(1 - e^{-k/a}) = 1 + o(1)\).

Note that for the value of parameter \(p = 1\), selecting a team of individuals with the largest mean test scores is optimal. Intuitively, one would expect that for small enough values of parameter \(p > 1\), the mean test scores would be good test scores. The result of Theorem 5 tells us that this is so if and only if \(p = 1 + O(1/\log(k))\). In the limit as \(p\) goes to infinity, in which the CES production function corresponds to the best-shot production function, we have that the expected utility of a team with the largest mean test scores is guaranteed to be at least \(1/k\) of the optimum expected utility, and this is a tight bound; this conforms to the result in [KR15].
E.2 Quantile Test Scores

Since the CES production function corresponds to the best-shot function in the limit of large values of parameter $p$, and we know from the result in [KR15] that quantile test scores are good test scores for the best-shot function, one would expect that quantile test scores are good test scores for the CES production function provided that the value of parameter $p$ is large enough. In the next theorem, we characterize a tight threshold for the parameter $p$ below which the quantile test scores are not good test scores for the CES production function.

**Theorem 6.** The following claims hold for quantile test scores with $q = 1 - \theta/k$:

1. If $p = o(\log(k))$ and $p > 1$, the quantile test scores are not good test scores for any value of parameter $\theta > 0$;

2. If $p = \Omega(\log(k))$, the test-score algorithm with quantile test scores with $\theta = 1$ yields a solution that is a constant-factor approximation of the optimum solution.

**Proof.** **Proof of Claim 1:**

If $k$ is a constant, there is no $p$ satisfying both conditions $p = o(1)$ and $p > 1$. Hence, it suffices to consider $k = \omega(1)$ and show that the following statement holds: for any given $\theta > 0$, there exists an instance for which the quantile test-score based team selection cannot give a constant-factor approximation.

Consider the following distributions for $X_i$:

1. Let each $X_i$ be equal to $a$ with probability $1$ for $1 \leq i \leq k$. Then, each quantile test-score is equal to $a$ and each replication test-score is equal to $ak^{1/p}$.

2. Let each $X_i$ be equal to $0$ with probability $1 - 1/n$, and equal to $b\theta n/k$ with probability $1/n$ for $k + 1 \leq i \leq 2k$. Then, in the limit as $n$ grows large, each quantile test-score is equal to $b$ and each replication test score is equal to $b\theta$.

3. Let each $X_i$ be equal to $0$ with probability $1 - \theta/k$ and equal to $c$ with probability $\theta/k$ for $2k + 1 \leq i \leq 3k$. Then, each quantile test-score is equal to $c$ and each replication test-score is less than or equal to $c\theta^{1/p}$.

4. Let $X_i$ be equal to $0$ for $3k + 1 \leq i \leq n$.

If $\theta$ is a constant (i.e., $\theta = O(1)$), we can easily check that the quantile test-score algorithm cannot give a constant-factor approximation with $a = b = 1$ and $c = 2$. Under this condition, the set of individuals $\{2k + 1, \ldots, 3k\}$ is selected by the quantile test-score algorithm. However,

$$
\frac{\mathbb{E} \left[ \left( \sum_{i=2k+1}^{3k} X_i^p \right)^{1/p} \right]}{\mathbb{E} \left[ \left( \sum_{i=1}^{b} X_i^p \right)^{1/p} \right]} = \frac{\mathbb{E} \left[ \left( \sum_{i=2k+1}^{3k} X_i^p \right)^{1/p} \right]}{k^{1/p}} \leq \frac{\left( \sum_{i=2k+1}^{3k} \mathbb{E} \left[ X_i^p \right] \right)^{1/p}}{k^{1/p}} = 2 \left( \frac{\theta}{k} \right)^{1/p} = o(1),
$$

since $k = \omega(1)$, $\theta = O(1)$, and $p = o(\log(k))$. 

24
If \( \theta \) goes to infinity as \( n \) goes to infinity (i.e., \( \theta = \omega(1) \)), we have
\[
\frac{\mathbb{E} \left[ \left( \sum_{i=2k+1}^{3k} X_i^p \right)^{1/p} \right]}{\mathbb{E} \left[ \left( \sum_{i=k+1}^{2k} X_i^p \right)^{1/p} \right]} \leq \frac{\sum_{i=2k+1}^{3k} \mathbb{E} \left[ X_i^p \right]^{1/p}}{\mathbb{E} \left[ X_i^p \right]^{1/p}} = \frac{2\theta(1-p)^{1/p}}{\theta} = o(1),
\]
because \( p > 1 \). Therefore, the quantile test-score based team selection has a negligible utility compared to the optimal utility.

**Proof of Claim 2** Let \( T(X_S) \) be a subset of \( S \) such that \( i \in T(X_S) \) if, and only if, \( X_i \geq F^{-1}_i(1 - h/k) \), for \( i \in S \). Let \( a_{\max}^k = \max_{i \in S} a_i^k \) and \( a_{\min}^k = \min_{i \in S} a_i^k \). In this proof, we will show that there exist constants \( q \) and \( p \) such that
\[
pq a_{\min}^k \leq \mathbb{E} \left[ \left( \sum_{i \in S} X_i^p \right)^{1/p} \right] \leq qa_{\max}^k.
\]
Since \( (x + y)^{1/p} \leq x^{1/p} + y^{1/p} \) when \( x, y \geq 0 \) and \( p > 1 \),
\[
\mathbb{E} \left[ \left( \sum_{i \in S} X_i^p \right)^{1/p} \right] = \mathbb{E} \left[ \left( \sum_{i \in T(X_S)} X_i^p + \sum_{i \in S \setminus T(X_S)} X_i^p \right)^{1/p} \right] \\
\leq \mathbb{E} \left[ \left( \sum_{i \in T(X_S)} X_i^p \right)^{1/p} + \left( \sum_{i \in S \setminus T(X_S)} X_i^p \right)^{1/p} \right] \\
\leq \mathbb{E} \left[ \sum_{i \in T(X_S)} X_i^p + \left( \sum_{i \in S \setminus T(X_S)} (a_{\max}^k)^p \right)^{1/p} \right] \\
\leq \left( \mathbb{E} \left[ |T(X_S)| \right] + k^{1/p} \right) a_{\min}^k = (1 + k^{1/p})a_{\max}^k.
\]

By the Minkowski inequality, \( \left( \sum_{i \in A} \mathbb{E} \left[ X_i^p \right]^{1/p} \right) \leq \mathbb{E} \left[ \left( \sum_{i \in A} X_i^p \right)^{1/p} \right] \) for all \( A \subseteq S \). Thus, we have
\[
\mathbb{E} \left[ \left( \sum_{i \in S} X_i^p \right)^{1/p} \right] = \mathbb{E} \left[ \left( \sum_{i \in T(X_S)} X_i^p + \sum_{i \in S \setminus T(X_S)} X_i^p \right)^{1/p} \right] \\
\geq \mathbb{E} \left[ \left( \sum_{i \in T(X_S)} X_i^p \right)^{1/p} \right] \\
= \sum_{A \subseteq S} \mathbb{P}(T(X_S) = A) \mathbb{E} \left[ \left( \sum_{i \in A} X_i^p \right)^{1/p} \right] \mathbb{P}(T(X_S) = A)
\]
\[ \geq \sum_{A \subseteq S} \Pr\{T(X_S) = A\} \left( \sum_{i \in A} \mathbb{E}[X_i | i \in T(X_S)]^p \right)^{1/p} \]
\[ \geq \sum_{A \subseteq S} \Pr\{T(X_S) = A\} |A|^{1/p} a_{\min}^k \]
\[ \geq \left(1 - (1 - 1/k)^k\right) a_{\min}^k \geq (1 - 1/e) a_{\min}^k. \]

Therefore, the quantile test-score team selection is a constant-factor approximation algorithm. \qed