Constrained Best Arm Identification

in Grouped Bandits

Abstract—We study a grouped bandit setting where each

arm comprises multiple independent sub-arms referred to as

attributes. Each attribute of each arm has an independent

stochastic reward. We impose the constraint that for an arm

to be deemed feasible, the mean reward of all its attributes

should exceed a specified threshold. The goal is to find the

arm with the highest mean reward averaged across attributes

among the set of feasible arms in the fixed confidence

setting. We propose a confidence interval-based policy to

solve this problem and provide instance-dependent analytical

guarantees for the policy.We compare the performance of the

proposed policy with that of two suitably modified versions

of action-elimination via simulations.

Index Terms—Stochastic multi-armed bandits, best arm

identification, fixed confidence setting

I. INTRODUCTION

Many services are a collection of independent components

and each customer of such a service may use one or

more of these at a time. For instance, a typical auto garage

may offer car wash services, AC repair and servicing, tyre

and wheel care services, car inspections, etc. Similarly,

a typical salon offers hair services, skin services, nail

services, make-up services, etc. To evaluate such services,

it makes sense to have customers rate each component

separately and maintain a rating for each component of the

service. A reasonable metric for evaluating such a service

as a whole is the (weighted) average of the ratings of

the different components. In addition, for a service to be

deemed acceptable, it may be desirable that the ratings for

each of its components exceed a threshold.

Motivated by this, we consider a grouped bandit setting

where each arm is a group of independent sub-arms. We

refer to these sub-arms as attributes. Each attribute of

each arm has a reward that is modeled as an independent

stochastic process with a corresponding mean reward. We

say that a group is feasible if the mean reward of all its

attributes exceeds a given threshold. In this work, we focus

on the problem of identifying the feasible group with the

highest mean reward averaged across all its attributes in

the fixed confidence setting [1]. We consider the setting

where the learner can choose to sample a specific attribute

of a specific arm.

A. Our Contributions

We propose a policy that determines which arm-attribute

pair(s) to sample in each round. We provide instancespecific

analytical performance guarantees for this policy.

Further, we empirically compare the performance of the

proposed policy with suitably adapted versions of widely

studied policies like action elimination. Our numerical

results show that our algorithm outperforms suitably modified

versions of the action-elimination algorithm.

B. Related Work

The best arm identification problem for multi-arm bandits

has been widely studied. The two settings of this

problem that have received the most attention are the fixed

confidence setting [1] and the fixed budget setting [2]. As

mentioned above, in this work, we focus on the former.

In the fixed confidence setting, we are given δ ∈ [0, 1] as

an input parameter. The goal of the learner is to identify

the best arm with probability at least 1 − δ using as few

samples of the arms as possible. Refer to [3] for a detailed

survey of various algorithms proposed for this setting.

The two key features of the problem we are interested

in are: (i) each arm is a group of independent arms, and

(ii) our goal is to identify the best arm among those which

satisfy a “feasibility” constraint. We now discuss existing

literature that looks at the best arm identification problem

with these two features.

A version of the grouped arm problem is the focus of

[4], where the goal is to identify the group with the highest

minimum mean reward. Another grouped arm problem is

studied in [5], where the goal is to identify the best arm in

each group. In [6], the goal is to identify the best m arms

that attain the highest rewards out of the group of arms.

Another related problem is called the categorized multiarmed

bandits [7]. Here arms are grouped into different

categories with an existing order between these categories,

and the knowledge of the group structure is known. The

goal is to find the best overall arm.

Multiple works have added a feasibility constraint to

the best arm identification problem, where the learning

agent is expected to select an optimal arm that satisfies

a feasibility rule. In [8], the authors consider the best

arm identification problem with linear and then monotonic

safety constraints. A method to solve a feasibilityconstrained

best-arm identification (FC-BAI) problem for

a general feasibility rule is shown in [9]. In [10], the

authors solve the FC-BAI problem with a constraint on

arm mean using the track and stop technique [1]. In [11],

the feasibility constraint is in terms of the variance of an

arm. In this problem setting, an arm is said to be feasible

if its variance is below a given threshold. We use ideas

from [11] in the design and analysis of our algorithm.

II. PROBLEM SETTING

There are N arms and [N] = {1, 2, ・ ・ ・ ,N} denotes

the set of all arms. Each arm has M attributes, denoted

by [M] = {1, 2, ・ ・ ・ ,M}. The reward corresponding to

each attribute of each arm is modeled as an independent

stochastic process. We use νij to denote the distribution of

attribute j ∈ [M] of arm i ∈ [N]. The reward of attribute j

TABLE I: Illustrative Example

Mean Reward Arm 1 Arm 2 Arm 3

Attribute 1 0.6 0.2 0.4

Attribute 2 0.4 1 0.4

Average 0.5 0.6 0.4

TABLE II: Notation

Notation Description

Xij (t) Reward for arm i, attribute j in round t

Xi(t) Reward for arm i in round t

μij Mean reward of arm i, attribute j

μi Mean reward of arm i

F Feasible Set

μTH Threshold

f Feasibility flag

i∗ Index of the best feasible arm

of arm i in round t is a stochastic random variable denoted

by Xij(t) ∼ νij . We define μij = E[Xij(t)].

The reward of arm i in round t, denoted by Xi(t) is the

average of rewards of all the attributes of that arm, i.e.,

Xi(t) =





MX

j=1

Xij(t)



/M. (1)

Further, we are given a threshold, μTH which determines

the feasibility of each arm. An arm i is said to be feasible

if and only if the mean reward of all its attributes is at

least μTH. We define the set of feasible arms, referred to

as the Feasible Set, denoted by F as follows:

F := {i ∈ [N] : min

j

μij ≥ μTH}. (2)

A problem instance is said to be feasible if F ̸= ∅ and

is called infeasible otherwise. We define the feasibility flag

f as follows

f :=

(

1 if the instance is feasible,

0 otherwise.

For a feasible instance, the best arm i∗ is defined as the

arm with the highest average mean reward in F, i.e.,

i∗ :=arg max

i∈F

μi, where, μi =





MX

j=1

μi,j



/M.

For example, consider the problem instance in Table I.

Here we have three arms, each with two attributes. Let

μTH = 0.3. In this case, the feasible arms are Arm 1 and

Arm 3 and the best feasible arm is Arm 1. Arm 2 has the

highest average mean reward, but is infeasible since the

mean reward for Attribute 1 is less than μTH.

We assume that the best arm, if it exists, is unique.

The rewards are considered to be bounded in [0, 1]. The

algorithmic challenge for the learner is to decide which

arm-attribute pairs to play in each round. The notation

used in the section is summarized in Table II.

III. OUR ALGORITHM: CONFIDENCE SET SAMPLING

We propose an LUCB-style algorithm [11], where we

sample multiple arms in each round. Note that we have an

extra degree of freedom as we can choose the attributes

to be sampled. We divide the arms and attributes into

subsets based on their potential feasibility and explore

arms for which we are uncertain about the feasibility. We

also explore the arms with assured feasibility to get tighter

confidence bounds for their average mean rewards. We

stop the algorithm when we ascertain that the average

mean reward of the current best feasible arm is greater

than that of any other feasible arm.

The algorithm starts with a uniform exploration for each

attribute of each arm. Rounds are indexed by t and the

total number of pulls till round t is denoted by k(t). Let

Jt denote the set of arm-attribute pairs pulled in round

t. We define Tij(t) as the number of samples of attribute

j of arm i taken till round t. Similarly, Ti(t) is the total

number of samples of arm i till time t. It follows that

Tij(t) :=

Xt−1

s=1

1(i,j)∈Js , Ti(t) := min

j∈[M]

Tij(t).

The empirical mean of the reward of attribute j of arm

i is denoted by ˆμij(t). The empirical average reward of

arm i is denoted by ˆμi(t). Formally,

ˆμij(t) :=

1

Tij(t)

Xt−1

s=1

Xs,i(t)1i∈Js ,

ˆμi(t) :=





MX

j=1

ˆμij



/M.

(3)

We define the confidence radii for the arms and attributes

as

α(t) :=

s

1

2T(t)

ln

\_

N(M + 1)(k(t))3

2δ

\_

,

where T(t) is corresponding to the arm or attribute-arm

pair for which we are calculating the confidence radii.

We define the confidence intervals for each attribute

with the lower confidence bound (LCB, denoted by Lij(t))

and the upper confidence bound (UCB, denoted by Uij(t))

as follows

Lij(t) := ˆμij(t) − α(t, Tij(t), k(t)),

Uij(t) := ˆμij(t) + α(t, Tij(t), k(t)).

(4)

Similarly, we define the confidence interval for arm i

with the lower confidence bound (LCB, denoted by Li(t))

and the upper confidence bound (UCB, denoted by Ui(t))

as follows

Li(t) := ˆμi(t) − α(t, Ti(t), k(t)),

Ui(t) := ˆμi(t) + α(t, Ti(t), k(t)).

(5)

Based on these confidence intervals, we define the

following subsets of the set of all arm-attribute pairs:

1) Perfectly Feasible Attribute Set: the set of armattribute

pairs whose lower confidence bound is above

the threshold μTH. Formally,

FA

Pt := {i ∈ [N], j ∈ [M] : Lij(t) ≥ μTH}.

2) Almost Feasible Attribute Set: the set of arm-attribute

pairs whose lower confidence bound is less than the

threshold μTH and the upper confidence bound is

larger than the threshold μTH. In other words, the

threshold lies within the confidence interval for these

arm-attribute pairs. Formally,

∂FA

t := {i ∈ [N], j ∈ [M] : Uij(t) ≥ μTH > Lij(t)}.

3) Feasible Attribute Set: the union of the Perfectly Feasible

Attribute Set and the Almost Feasible Attribute

Set and is denoted by FA

t .

4) Infeasible Attribute Set: All the arm-attribute pairs

not in the Feasible Attribute Set, i.e., the attributes

have UCB higher than the threshold.

Based on the confidence intervals, we define the following

subsets of the set of all arms:

1) Perfectly Feasible Set: the set of arms with all the

attributes in the Perfectly Feasible Attribute Set. Formally,

FPt := {i ∈ [N] : (i, j) ∈ FA

Pt, ∀j ∈ [M]}.

2) Feasible Set: the set of arms with all the attributes in

the Feasible Attribute Set, i.e., all the attributes have

UCB higher than μTH. Formally,

Ft := {i ∈ [N] : (i, j) ∈ FA

t , ∀j ∈ [M]}.

3) Almost Feasible Set: the set of arms which are in the

Feasible Set but not in the Perfect Feasible Set. This

is denoted by ∂Ft.

4) Infeasible Set: the set of arms with at least one

attribute in the Infeasible Attribute Set, which makes

the arm infeasible.

5) Potential Set: the set of arms with UCB lower than

the LCB of the empirically best arm. Formally,

Pt :=

(

{i ∈ [N] : Li⋆t (t) ≤ Ui(t) F ̸= ϕ,

[N] F = ϕ.

(6)

These notations are summarised in Table III

TABLE III: Common notation in algorithm.

Notation Description

k(t) Total number of pulls till time t

Tij (t) Number of pulls - arm i, attribute j till time t

Ti(t) Number of pulls of arm i till time t

ˆμij (t) Empirical mean reward - arm i, attribute j

ˆμi(t) Empirical mean reward of arm i at time t

α(t) Confidence Radii

Lij (t),Uij(t) Attribute Confidence bounds

Li(t),Ui(t) Arm confidence bounds

FA

Pt Perfectly Feasible Attribute Set

∂FA

t Almost Feasible Attribute Set

FA

t Feasible Attribute Set

FPt Perfectly Feasible Arm Set

∂Ft Almost Feasible Arm Set

Ft Feasible Arm Set

Pt Potential Set

A. Stopping Criteria

The algorithm stops when there are no competitor arms

to be pulled, that is the set Ft ∩ Pt = ϕ. We then check

the feasible set, if Ft = ϕ, then the given instance is

declared to be infeasible, and the feasibility flag ˆ f is set

to 0. Otherwise, the feasibility flag is set to 1 and the arm

it is declared as the best feasible arm.

Algorithm 1 Confidence Set Sampling

1: Sample each of the N arms once

2: Set FN = [N]

3: for time steps t > M × N do

4: Calculate ˆμij , ˆμi ∀i, j using (3)

5: Calculate confidence bounds using (4) and (5)

6: Update ∂FA

t , FA

t , FPt, ∂Ft, Ft according to III

7: Find i∗t := arg max{ˆμi(t) : i ∈ FPt}

8: Update Pt according to (6)

9: Set it := arg max{ˆμi(t) : i ∈ Ft}

10: Set competitor arm

ct := arg max{Ui(t) : i ∈ Ft, i ̸= it}

11: if Ft ∩ Pt = ϕ then

12: if Ft ̸= ϕ then, Set iout = it, ˆ f = 1

13: else Set ˆ f = 0

14: end if

15: break

16: end if

17: if |Ft| = 1 then

18: Pull (it, j) such that (it, j) ∈ ∂FA

t

19: If no such j, pull all attributes of it

20: else

21: Find it and ct

22: Pull (i, j) such that i ∈ {it, ct}, (i, j) ∈ ∂FA

t

23: If no such j, pull all attributes of it and ct

24: end if

25: end for

B. Sampling Criteria

We consider two cases. The first case is when there is

a single arm in the Feasible set. In this case, we find it

defined as the arm with the highest empirical average mean

reward from the Almost Feasible Set. We then check the

arm-attribute pairs in the Possibly Feasible Attribute Set

and pull all the attributes of arm it in this set. If no such

arm-attribute pair exists, we pull all the attributes of the

arm it once each.

In the second case, i.e., when there are more than one

arms in the Feasible Set, it is defined as the best arm

from the Feasible Set, and ct is the potentially competitor

arm, that is an arm other than it from the feasible set,

which has the highest UCB. We then check the Possibly

Feasible Attribute Set and pull all the arm-attribute pairs

corresponding to arms it or ct. If no such pairs are found,

we pull all the attributes of the arms it and ct.

The pseudo-code of the above algorithm is given in 1.

IV. ANALYTICAL PERFORMANCE GUARANTEES

In this section, we provide analytical guarantees for the

performance of our algorithm. We define the following

sets based on the ground truth:

1) Suboptimal Set: the set of arms with average mean

reward less than that of the best feasible arm. Formally,

S :=

(

{i ∈ [N] : μi < μ⋆} F ̸= ϕ

ϕ F = ϕ.

2) Risky Set: the set of arms with average mean reward

more than that of the best feasible arm. Note that by

definition, all these arms are infeasible. Formally,

R := [N] \ S.

We define i⋆⋆ := arg max{μi : i ∈ S}, and for all

i ∈ S,Δi := μi⋆ − μi if F ̸= ϕ. Further,

Δi⋆ := μi⋆ − μi⋆⋆ . (7)

Similarly, Δattr

ij := μij − μTH. We also define Δattr

i :=

| min

j

μij − μTH|.

Further, the separator ￣μ is defined as follows:

￣μ :=

(

(μi⋆ + μi⋆⋆ )/2 F ̸= ϕ and S ̸= ϕ,

−∞ otherwise.

(8)

We also have Empirically Suboptimal Set, Risky Set,

and Neutral Set depending on the separator value and are

defined according to the equations given below:

St := {i : Ui(t) < ￣μ}

Rt := {i : Li(t) > ￣μ}

Nt := [N] \ (St ∪ Rt) = {i : Li(t) ≤ ￣μ ≤ Ui(t)}.

Next, we define the hardness index of a problem instance,

denoted by Hid, as

Hid :=

1

min{Δi∗

2 , 2Δattr

i∗ }2

+

X

i∈F∩S

1

􀀀Δi

2

\_2

+

X

i∈F￣C∩R

1

\_

Δattr

i

2

\_2 +

X

i∈F￣C∩S

1

max{Δi

2 , 2Δattr

i }2

.

We now state an upper bound on the number of samples

required by CSS-LUCB in the following theorem.

Theorem 1 (Upper bound). Given an instance and a

confidence parameter δ, with probability at least 1 − δ,

the CSS-LUCB algorithm succeeds and terminates in

O

􀀀

Hid ln Hid

δ

\_

samples.

The proof outline is as follows:

– We first define an event E where the means of all

arms and all attributes lie within their confidence

intervals for all rounds t ≥ 2. We prove that if the

algorithm terminates, this event occurs with “high

probability”.

– Next, we prove that if event E occurs, the algorithm

will give the correct answer.

– We then prove that if the algorithm does not terminate,

one of the two following conditions is satisfied:

• it ∈ (∂Ft\St) ∪ (Ft ∩ Nt)

• ct ∈ (∂Ft\St) ∪ (Ft ∩ Nt)

We then show that these conditions occur with “low

probability”.

– Finally, we prove that for some t > K ×

Hid ln(Hid/δ), for a constant K, the algorithm terminates

with “high probability”.

These steps combined show that the algorithm terminates

and identifies the best arm with probability at least 1−δ.

V. NUMERICAL RESULTS

In this section, we present our numerical results. We

compare the performance of our policy with two suitably

adapted variants of the widely studied action elimination

algorithm [3]. More specifically, along with the usual

elimination of arms whose averaged mean reward is low,

we also eliminate arms that have one or more attributes

whose UCB is less than the given threshold (μTH) . In

addition, we also simulate an algorithm that divides the

problem into two sub-tasks. The first task is to eliminate

arms that are infeasible, following which, the second task

is to identify the best arm in the set of feasible arms. We

use action elimination for the second task. We refer to this

approach as Feasibility then BAI.

TABLE IV: Mean Rewards

Experiment 1: x varies from 0.6 to 0.9, μTH = 0.3

Arm Attribute 1 Attribute 2 Remarks

1 x 0.6 Best arm

2 0.5 0.6 Feasible

3 0.2 0.8 Feasible

4 0.4 0.5 Feasible

5 0.5 0.3 Feasible

Experiment 2

Experiment 2: x varies from 0.6 to 0.9, μTH = 0.4

Arm Attribute 1 Attribute 2 Remarks

1 x 0.8 Best arm

2 0.3 1 Infeasible

3 0.5 0.6 Feasible

4 0.4 0.5 Feasible

5 0.1 0.5 Infeasible

Experiment 2

Experiment 3: x varies from 0.35 to 0.45, μTH = 0.5

Arm Attribute 1 Attribute 2 Remarks

1 0.6 0.7 Best feasible arm

2 x 0.9 Infeasible

3 0.3 0.55 Infeasible

4 0.55 0.55 Feasible

5 0.2 0.4 Infeasible

For the results presented in this section, the reward of

each attribute of each arm is an independent stochastic

process with the Beta distribution [12]. We perform three

experiments. In the first experiment, we consider the case

where all arms are feasible. In the second experiment,

some sub-optimal arms are infeasible. The arm with the

highest mean remains feasible. Finally, in the third experiment,

the arm with the highest average mean reward is

infeasible, i.e., it has an attribute with mean reward below

the threshold. We set δ = 0.1, N = 5, and M = 2. The

values of various parameters are given in Table IV. The

number of samples required by each algorithm is plotted

against the hardness of the problem given by the reciprocal

of Δi∗ which is calculated using (2). The results are shown

in the Fig. 1. We observe that our algorithm outperforms

the standard algorithms in all our experiments.

Next, we vary the number of arms and attributes,

comparing the results of the CSS-LUCB algorithm in the

case where the arm with the highest mean was infeasible

and the results are shown in Fig. 2. In Fig 2a, we compare

the sample complexity for varying number of arms keeping

number of attributes constant. For N ∈ {4, 5, 6}, we

consider arms 1, 2, ・ ・ ・N, shown in Table V. In Fig 2b,

6 8 10 12

Hardness

10000

20000

30000

40000

50000

60000

70000

80000

Number of Samples

Action Elimination

Feasibility then BAI

CSS\_LUCB

(a) Experiment 1

5 6 7 8 9 10 11 12

Hardness

5000

10000

15000

20000

25000

30000

Number of Samples

Action Elimination

Feasibility then BAI

CSS\_LUCB

(b) Experiment 2

8 10 12 14 16 18 20

Hardness

20000

30000

40000

50000

60000

70000

Number of Samples

Action Elimination

Feasibility then BAI

CSS\_LUCB

(c) Experiment 3

Fig. 1: Sample-complexity as a function of the hardness

of problem

we vary number of attributes for a fixed number of arms.

For M ∈ {2, 3, 4}, we consider the first two, three, and

four attributes respectively from Table V. As expected,

we see that sample complexity increases as the number of

arms or the number of attributes increases.

TABLE V: Mean Rewards

Varying N, fixed M = 2, μTH = 0.5

Arm Attribute 1 Attribute 2

1 x 0.9

2 0.3 0.55

3 0.55 0.55

4 0.2 0.4

5 0.6 0.7

6 0.55 0.6

Experiment 2

Varying M, fixed N = 5, μTH = 0.5

Arm Attribute 1 Attribute 2 Attribute 3 Attribute 4

1 x 0.9 0.7 0.8

2 0.3 0.55 0.4 0.6

3 0.55 0.55 0.55 0.55

4 0.2 0.4 0.3 0.55

5 0.55 0.6 0.65 0.7

7.5 10.0 12.5 15.0 17.5 20.0 22.5 25.0

Hardness

10000

15000

20000

25000

30000

35000

40000

45000

50000

Number of Samples

N = 4

N = 5

N = 6

(a) Varying number of arms

7.5 10.0 12.5 15.0 17.5 20.0 22.5 25.0

Hardness

20000

25000

30000

35000

40000

45000

50000

Number of Samples

M = 2

M = 3

M = 4

(b) Varying number of attributes

Fig. 2: Sample-complexity of CSS-LUCB as a function of

the number of arms and attributes

REFERENCES

[1] Aur´elien Garivier and Emilie Kaufmann. Optimal best arm identification

with fixed confidence. In Vitaly Feldman, Alexander

Rakhlin, and Ohad Shamir, editors, 29th Annual Conference on

Learning Theory, volume 49 of Proceedings of Machine Learning

Research, pages 998–1027, Columbia University, New York, New

York, USA, 23–26 Jun 2016. PMLR.

[2] Chao Qin. Open problem: Optimal best arm identification with

fixed-budget. In Conference on Learning Theory, pages 5650–5654.

PMLR, 2022.

[3] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms

for multi-armed bandits in the fixed confidence setting. In

2014 48th Annual Conference on Information Sciences and Systems

(CISS), pages 1–6, 2014.

[4] Zhenlin Wang and Jonathan Scarlett. Max-min grouped bandits.

In Proceedings of the AAAI Conference on Artificial Intelligence,

volume 36, pages 8603–8611, 2022.

[5] Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric,

and S´ebastien Bubeck. Multi-bandit best arm identification. Advances

in Neural Information Processing Systems, 24, 2011.

[6] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter

Stone. Pac subset selection in stochastic multi-armed bandits. In

ICML, volume 12, pages 655–662, 2012.

[7] Matthieu Jedor, Vianney Perchet, and Jonathan Louedec. Categorized

bandits. Advances in Neural Information Processing Systems,

32, 2019.

[8] Zhenlin Wang, Andrew Wagenmaker, and Kevin G. Jamieson. Best

arm identification with safety constraints. CoRR, abs/2111.12151,

2021.

[9] Julian Katz-Samuels and Clayton Scott. Top feasible arm identification.

In Kamalika Chaudhuri and Masashi Sugiyama, editors,

Proceedings of the Twenty-Second International Conference on

Artificial Intelligence and Statistics, volume 89 of Proceedings of

Machine Learning Research, pages 1593–1601. PMLR, 16–18 Apr

2019.

[10] Yuhang Wu, Zeyu Zheng, and Tingyu Zhu. Best arm identification

with fairness constraints on subpopulations, 2023.

[11] Yunlong Hou, Vincent YF Tan, and Zixin Zhong. Almost optimal

variance-constrained best arm identification. IEEE Transactions on

Information Theory, 69(4):2603–2634, 2022.

[12] Wikipedia contributors. Beta distribution — Wikipedia, the free

encyclopedia, 2024. [Online; accessed 10-January-2024].