AAMAS2021
LONDON, UK

A Multi-Arm Bandit Approach To Subset Selection Under Constraints

Ayush Deva, Kumar Abhishek, Sujit Gujar

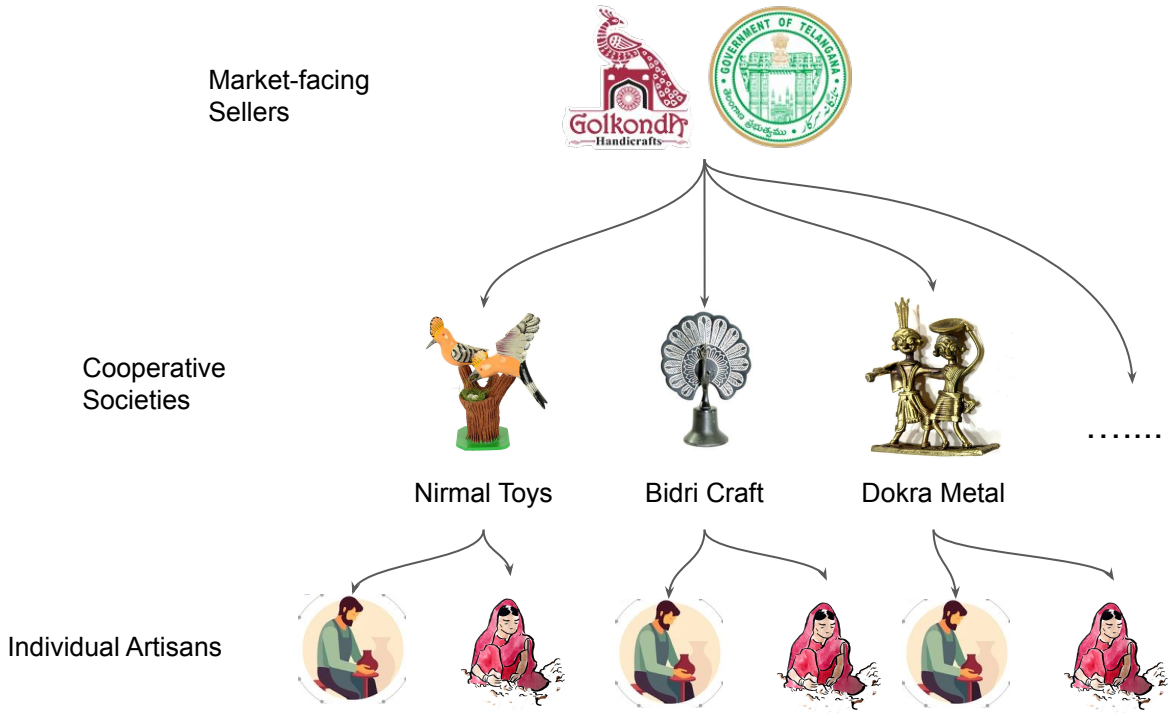
ayushdeva97@gmail.com | kumar.abhishek@research.iiit.ac.in | sujit.gujar@iiit.ac.in

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Motivation

Cooperative Societies, such as those of artisans, where the quality and perceived value of each produce may vary greatly, it becomes important to carefully manage the inventory and generate supply accordingly.

The problem faced is hence about deciding what quantity of each type of product to produce / procure so as to maximize the overall revenue of the cooperative(s).



Subset Selection and Its Applications

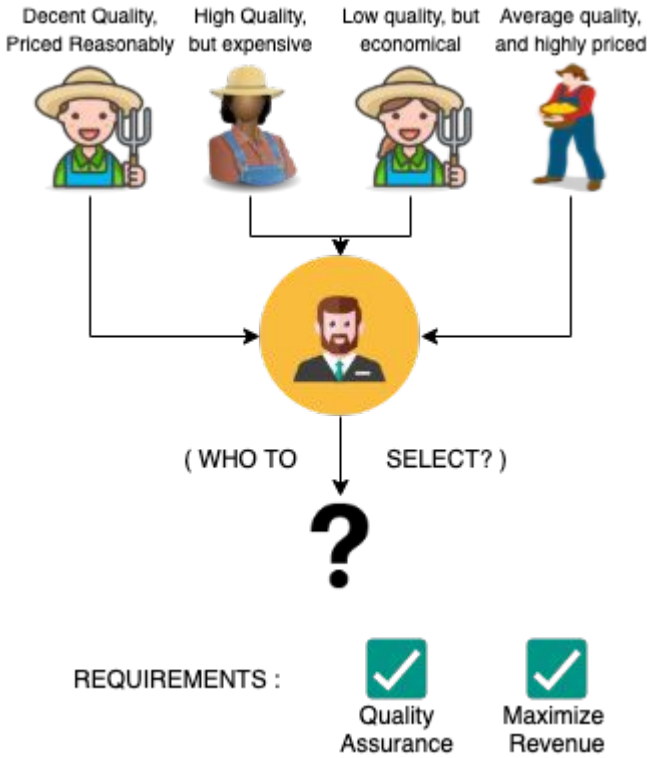
Such problems fall under the wider category of **‘Subset Selection’** and is faced not only by cooperative societies but also in other familiar marketplaces such as E-commerce Platforms and Supermarkets etc.

In all scenarios, the broader goal is to maximize revenue whilst ensuring certain quality standards.

amazon → Which seller’s products to display to a user for a particular searched keyword.

coop → How many apples to procure from each of the regional farmers.

Problem Formulation



N agents, each associated with
capacity k_i,
cost c_i, and
quality q_i.

Expected utility to planner upon selecting an agent is equal to $Rq_i - c_i$

Average Quality Threshold α

$$
\begin{aligned}
\max_{x_i} \quad & \sum_{i \in N} (Rq_i - c_i)x_i \\
\text{s.t. } & q_{av} = \frac{\sum_{i \in N} q_i x_i}{\sum_{i \in N} x_i} \\
& q_{av} \geq \alpha \\
& 0 \leq x_i \leq k_i \qquad \forall i \in N \\
& x_i \in \mathbb{Z} \qquad \forall i \in N
\end{aligned}
\tag{1}
$$

Dynamic Programming Based Solution

Agents are segregated into four categories :

- High Quality, Low Cost (S_1) ✅
- Low Quality, High Cost (S_4) ❌
- High Quality, High Cost (S_3) ❓
- Low Quality, Low Cost (S_2) ❓

We formulate it as a dynamic programming problem where selecting each unit brings in a revenue of $R^*(q_i - c_i)$ at a cost of $(\alpha - q_i)$, which is the loss in quality.

Hence, the problem is to maximize the revenue while making sure that the total cost is within *budget*.

Algorithm 1 DPSS

Inputs: N, α, R, costs $\boldsymbol{c} = \{c_i\}_{i \in N}$, qualities $\boldsymbol{q} = \{q_i\}_{i \in N}$
Output: Quantities procured $\boldsymbol{x} = (x_1, \ldots, x_n)$
Initialization: $\forall i \in N, r_i = Rq_i - c_i, z = 0$
Segregate S_1,S_2,S_3,S_4 as described in Section 3.4
$\forall i \in S_1, x_i = 1; z = z + r_i; d = \sum_{i \in S_1}(q_i - \alpha)$
$\forall i \in S_4, x_i = 0$
$G = S_2 \cup S_3$; $\forall i \in G, d_i = q_i - \alpha$
function DP($i, d^{te}, x^{te}, x^\star, z^{te}, z^\star$)
    if $i ==$ $|G|$ and $d^{te} < 0$ **then return** $x^\star, z^\star$
    if $i ==$ $|G|$ and $d^{te} \geq 0$ **then**
        if $z^{te} > z^\star$ **then**
            $z^\star = z^{te}; x^\star = x^{te}$
        return $x^\star, z^\star$
    $x^\star, z^\star = DP(i+1, d^{te}, [x^{te}, 0], x^\star, z^{te}, z^\star)$
    $x^\star, z^\star = DP(i+1, d^{te} + d_i, [x^{te}, 1], x^\star, z^{te} + r_i, z^\star)$
    return $x^\star, z^\star$
x^G, z^G = DP(0,d,[ ],[ ],0,0)
$\forall i \in G, x_i = x_i^G$
return x

What if qualities are unknown?

- Quality of an individual item of produce is tough to measure.

- Quality of an agent is defined as the expected quality of its produce.

- The planner needs to carefully learn the qualities by procuring units from the agents across multiple round in a way that minimizes the loss of revenue.

This is an example of Exploration vs Exploitation trade-off.

- This type of sequential learning problem is often formulated as Multi-Armed Bandit (MAB) problem.

- Here, each agent is modelled as an arm associated with an unknown stochastic parameter : quality.

- Since the planner can select multiple arms in each round (say, a day), and gets to observe the sale of all the procured products, this problem falls under the category of Combinatorial MAB (CMAB) with semi-bandit feedback.

Isn’t there something that already addresses this?

Combinatorial Multi-Armed Bandit with General Reward Functions

Wei Chen* Wei Hu† Fu Li‡ Jian Li§ Yu Liu¶ Pinyan Lu‖

- Assumes the availability of a feasible set of subsets (which they refer to as ‘super arm’) to select from
- In our setting, the feasible set (that satisfies the Quality Constraint) is unknown and needs to be learnt as well.

A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing

Shweta Jain[a,*], Sujit Gujar[b,1], Satyanath Bhat[c,2], Onno Zoeter[d,3], Y. Narahari[a]

[a] Indian Institute of Science, Bangalore, India
[b] International Institute of Information Technology, Hyderabad, India
[c] Institute of Operations Research and Analytics, NUS Business School, Singapore, Singapore
[d] Booking.com, Amsterdam, Netherlands

- The objective function depends only on the agents’ cost (which are unknown) and not the qualities of the agents.
- In our setting, both the constraint and the objective function depends on the unknown parameter.

(DP)SS - UCB

We propose a UCB-based framework that assumes the availability of an offline Subset Selection Algorithm (SSA) which outputs the optimal solution for the qualities and cost of agents, and the quality threshold provided to it as input. When DPSS is used as the SSA, we refer to it as DPSS-UCB.

We prove that our framework achieves the following two properties:

1. Constraint Satisfaction
2. Sublinear Regret

Algorithm 2 SS-UCB

Inputs: N, α, ϵ_2, R, costs $\boldsymbol{c} = \{c_i\}_{i \in N}$
For each agent i, maintain: w_i^t, q_i^t, $(\hat{q}_i^t)^+$
$\tau \leftarrow \frac{3 \ln T}{2\epsilon_2^2}$; $t = 0$
while $t \leq \tau$ **(Explore Phase) do**
    Play a super-arm $S^t = N$
    Observe qualities $X_i^j, \forall i \in S^t$ and update w_i^t, $\hat{q}_i^t$
    $t \leftarrow t + 1$
while $t \leq T$ **(Explore-Exploit Phase) do**
    For each agent i, set $(\hat{q}_i^t)^+ = \hat{q}_i^t + \sqrt{\frac{3 \ln t}{2 w_i^t}}$
    $S^t = \text{SSA}\left(\{(\hat{q}_i^t)^+\}_{i \in N}, c, \alpha + \epsilon_2, \text{R}\right)$
    Observe qualities $X_i^j, \forall i \in S^t$ and update w_i^t, $\hat{q}_i^t$
    $t \leftarrow t + 1$

Ensuring Quality Constraints

Ensuring Qualities during exploration is difficult and counter-productive.

We provide PAC bounds on DPSS satisfying QC after 'τ' rounds.

LEMMA 2. *The difference between the average of $(\hat{q}_i^t)^+$ and the average of $\hat{q}_i^t$ over the agents i in S^t is less than ϵ_2, $\forall t > \tau$.*

PROOF. We have,

$$\frac{1}{s^t}\sum_{i\in S^t}((\hat{q}_i^t)^+ - \hat{q}_i^t) = \frac{1}{s^t}\sum_{i\in S^t}\frac{\sqrt{3\ln t}}{\sqrt{2w_i^t}} \leq \frac{\sqrt{3\ln t}}{\sqrt{2w_{min}^t}}.$$

where $w_{min}^t = \min_i w_i^t$. Since, for $t < \tau$, we are exploring all the agents, thus, $w_i^\tau = \tau$. Now, since $w_i^t \geq w_i^\tau, \forall t > \tau$, thus, we claim that $w_{min}^t \geq \tau$ for $t > \tau$. Hence,

$$\frac{\sqrt{3\ln t}}{\sqrt{2w_{min}^t}} \leq \frac{\sqrt{3\ln T}}{\sqrt{2\tau}}.$$

For $\tau = \frac{3\ln T}{2\epsilon_2^2}$, we have,

$$\frac{1}{s^t}\sum_{i\in S^t}((\hat{q}_i^t)^+ - \hat{q}_i^t) \leq \epsilon_2.$$

□

LEMMA 3. $\forall t > \tau$

$$\mathcal{P}\left(q_{av}^t < \alpha - \epsilon_1 \mid \frac{1}{s^t}\left(\sum_{i\in S^t}(\hat{q}_i^t) \geq \alpha\right)\right) \leq exp(-\epsilon_1^2 t).$$

PROOF. Let $Y^t = \frac{1}{s^t}\sum_{i\in S^t}\hat{q}_i^t$. Since $\mathrm{E}[\hat{q}_i^t] = \mathrm{E}[X_i^j] = q_i$, $\mathrm{E}[Y^t] = q_{av}^t$. Hence, we have,

$$\mathcal{P}(E[Y^t] < \alpha - \epsilon_1) \mid Y^t \geq \alpha) \leq \mathcal{P}(Y^t \geq E[Y^t] + \epsilon_1) \leq exp(-\epsilon_1^2 w^t).$$

where $w^t = \sum_{i\in S^t} w_i^t$, i.e., total number of agents selected till round t. Since we pull atleast one agent in each round, we can say that, $w^t \geq t$. Thus, $\forall t > \tau$

$$\mathcal{P}\left(q_{av}^t < \alpha - \epsilon_1 \mid \frac{1}{s^t}\left(\sum_{i\in S^t}(\hat{q}_i^t) \geq \alpha\right)\right) \leq exp(-\epsilon_1^2 t).$$

□

Regret Analysis

Refer to paper

https://arxiv.org/pdf/2102.04824.pdf

Greedy Approach

DPSS is $O(2^n)$ which makes it difficult to scale when *n* is large.

We propose a greedy based algorithm, GSS, that runs in polynomial time, O(nlogn), and provides an approximate solution to our ILP.

GSS can also be used as a SSA in the SS-UCB framework, which we refer to as GSS-UCB.

Algorithm 3 GSS

Inputs: N, α, R, costs $\boldsymbol{c} = [c_i]$, qualities $\boldsymbol{q} = [q_i]$
Output: Quantities procured $\boldsymbol{x} = (x_1, \ldots, x_n)$
Initialization: $\forall i \in N$, $r_i = Rq_i - c_i$
Segregate S_1,S_2,S_3,S_4 as described in Section 3.4
$\forall i \in S_1$, $x_i = 1$; $d = \sum_{i \in S_1}(q_i - \alpha)$
$\forall i \in S_4$, $x_i = 0$
$L_2 = sort(S_2)$ on decreasing order of $\frac{r_i}{\alpha - q_i}$
$L_3 = sort(S_3)$ on increasing order of $\frac{r_i}{\alpha - q_i}$
$p = 0$, $q = 0$
while $d > 0$ and $p < |S_2|$ **do**
    $i = L_2[p]$;
    if $\alpha - q_i \leq d$ **then** $x_i = 1$, $d = d - \alpha - q_i$, $p+ = 1$
    else $x_i = \frac{d}{\alpha - q_i}$, $d = 0$
while $p < |S_2|$ and $q < |S_3|$ **do**
    $i = L_2[p]$, $j = L_3[q]$
    $a = \frac{r_i}{\alpha - q_i}$, $b = \frac{r_j}{\alpha - q_j}$
    if $a \leq b$ **then break**;
    $w_1 = \min((1 - x_i)(\alpha - q_i), (1 - x_j)(q_j - \alpha))$
    $x_i+ = \frac{w_1}{\alpha - q_i}$, $x_j+ = \frac{w_1}{q_j - \alpha}$
    if $x_i == 0$ **then** $p++$;
    if $x_j == 0$ **then** $q++$;
if $0 < x_j < 1$ **then** $x_j = 1$
return $\lfloor \boldsymbol{x} \rfloor$

How Fast and How Approximate?

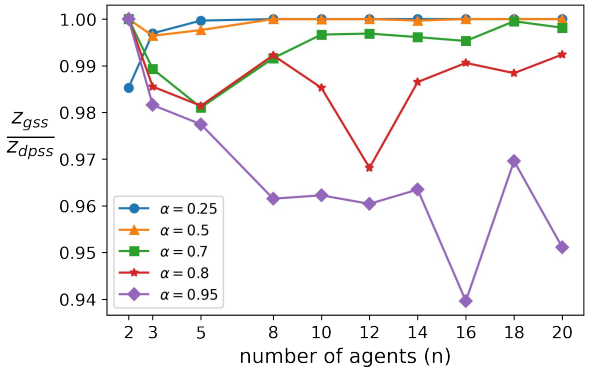
We compare the empirical performance of GSS with DPSS and an ILP solver - the COIN-OR Branch and Cut Solver (CBC).

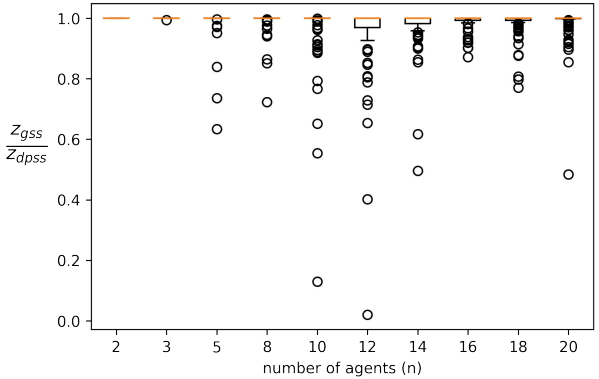
The theoretical approximation limit can be arbitrarily small but in practice it gives an average approximation ratio greater than 0.9

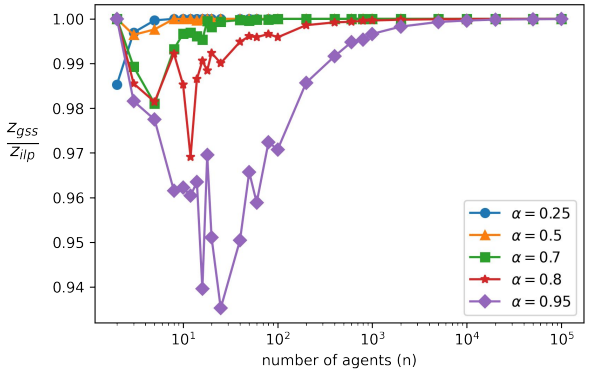
n	$t_{dpss} : t_{gss}$	$t_{ilp} : t_{gss}$
2	5.5	70
5	15.7	64
8	32.6	63.7
10	54.3	58.6
12	106.3	67.6
14	284.4	65.3
16	897.1	60.2
18	3109.7	63.1
20	11360.6	68.1

n	$t_{ilp} : t_{gss}$
25	66.7
50	58.3
100	52.7
400	43.1
1000	31.8
5000	31.6
10000	34.5
50000	45
100000	56.8

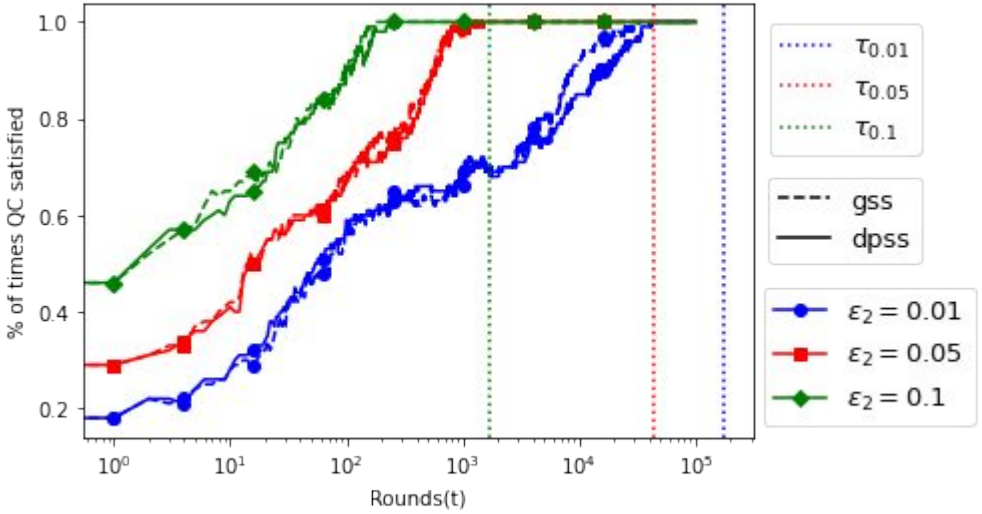
Table 1: Computational Performance of GSS w.r.t. to DPSS and ILP



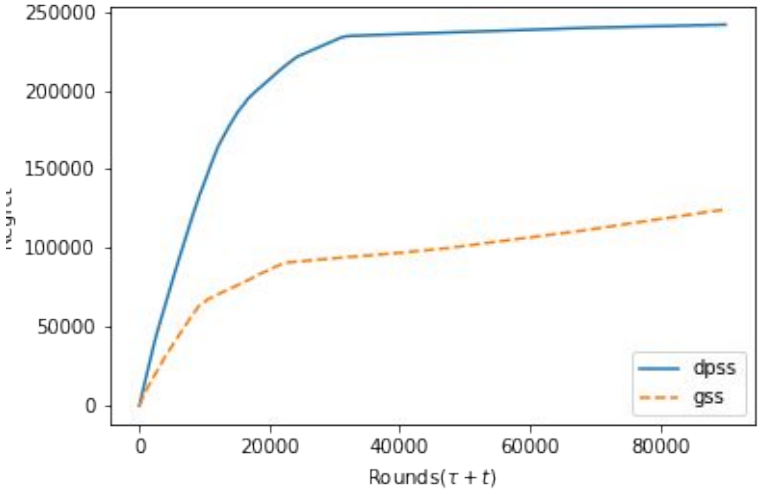




Experimental Results



Constraint Satisfaction



Regret Incurred

Possible Extensions To This Research

1. Dynamic Pool of Agents

2. An agent selected in a particular round is not available for the next few round possibly due to the lead time in procuring the units. *Sleeping Bandits*.

3. Inclusion of strategic agents where a planner also needs to elicit the cost of production truthfully. *Mechanism Design*


Thank you