Hybrid Combinatorial Multi-Armed Bandits for Discrete Resource Allocation Under Fairness Constraints

Wafik Aboualim
University of Pittsburgh
wat29@pitt.edu

Abstract

We study a sequential resource allocation problem where, in each round, a decision-maker must distribute a discrete resource under a fixed budget across multiple entities to maximize cumulative reward over a horizon T. This setting covers applications such as allocating police beats, compute resources, or humanitarian aid. Existing approaches often struggle with large combinatorial action spaces and offer limited flexibility for incorporating practical hard constraints, including minimum service levels or fairness requirements. We propose HybCLCB-DRA, a hybrid offline-to-online combinatorial multi-armed bandit (CMAB) algorithm that warm-starts from a pre-collected dataset and then adapts through online interaction. For scenarios that require explicit fairness constraints, we introduce FAIR-HybCLCB-DRA, which incorporates such constraints through a constrained allocation oracle. We evaluate both approaches in a case study on allocating police officers across Chicago's neighborhoods using historical crime and demographic data.

1 Introduction

Resource allocation is a fundamental challenge that arises in a wide range of applications, from bandwidth assignment and task scheduling to public-sector decisions such as distributing aid or allocating police units across a city. In all of these settings, a decision-maker must distribute a limited set of discrete resources across multiple competing regions, tasks, or communities, each of which exhibits heterogeneous and often uncertain reward characteristics.

A natural framework for modeling this class of problems is the *Combinatorial Multi-Armed Bandit (CMAB)*.The CMAB has become a staple reinforcement learning framework over the past decade, providing the structural setup upon which algorithms balance exploration and exploitation by searching for the super-arms with the highest rewards. It is particularly attractive for discrete resource allocation because each feasible allocation can be represented as a super-arm with rewards determined by the underlying utility of assigning resources to each component. Despite this progress, prior CMAB for resource allocation approaches (Zuo and Joe-Wong 2021; Gupta et al. 2022) face two major limitations when applied to real-world settings:

1. **Poor performance in large combinatorial action spaces.** When the number of super-arms grows combinatorially, as in city-scale resource allocation, classical optimistic algorithms such as CUCB (Zuo and Joe-Wong 2021) often over-estimate many unobserved or sparsely observed actions. This produces unstable learning curves and long exploratory phases.

2. **Inability to exploit abundant offline datasets.** Many domains, including urban policing, offer rich historical datasets that reflect how past allocations performed. Standard online CMAB algorithms require costly and sometimes ethically questionable exploration in the real environment.

Our work aims to tackle these two issues. We propose two algorithms: **HybCLCB-DRA** and its fairness-aware extension **FAIR-HybCLCB-DRA**. HybCLCB-DRA is a hybrid offline and online lower-confidence-bound algorithm for discrete resource allocation. The offline phase uses historical data to construct pessimistic confidence estimates over arm–weight pairs, and the online phase continually refines these estimates through cautious exploration, which improves performance and allows adaptation to changing reward patterns while remaining conservative in regions that lack data. FAIR-HybCLCB-DRA augments this framework with a flexible constraint-handling layer that can enforce a wide range of policy requirements, including minimum or maximum service guarantees, group parity, and other application-specific allocation rules. This allows the learner to balance reward with domain constraints in a principled way.

2 Related Work

CMAB for Resource Allocation. Combinatorial Multi-Armed Bandits (CMABs) provide a natural framework for sequential discrete resource allocation, where each feasible allocation corresponds to a super-arm that maximizes cumulative reward (Chen, Wang, and Yuan 2013). Bandits with Knapsacks (BwK) further extends this setting to budget-limited decisions (Badanidiyuru, Kleinberg, and Slivkins 2013). Zuo and Joe-Wong (Zuo and Joe-Wong 2021; Gupta

et al. 2022) studied a discrete-budget CMAB formulation for practical resource allocation tasks, providing the closest analogue to our setup.

Online CMAB. Classical online CMAB algorithms, including CUCB and its variants (Chen, Wang, and Yuan 2013), balance exploration and exploitation over combinatorial action spaces. Although theoretically strong, these methods struggle in large action spaces and do not leverage available offline datasets.

Offline CMAB. Offline RL has demonstrated the importance of pessimism for learning reliable policies from fixed datasets (Jin, Yang, and Wang 2020). In the CMAB setting, Liu et al. proposed Off-CMAB (Liu et al. 2025), the first offline algorithm for probabilistically triggered submodular CMABs, showing that effective policies can be learned without online exploration. However, offline CMAB methods have not been applied to knapsack-style resource allocation or to settings requiring constraints such as fairness or service guarantees.

Hybrid Bandits. Hybrid offline–online bandits that warm-start the learner with offline estimates and refine them online have been studied broadly (Shivaswamy and Joachims 2012; Song et al. 2023). However, hybrid CMAB methods have not been formally developed for resource allocation. Our work fills this gap by proposing hybrid pessimistic CMAB methods for knapsack-constrained allocations, with and without domain-specific constraints. The proposed HybCLCB-DRA and FAIR-HybCLCB-DRA algorithms build on CUCB-DRA (Zuo and Joe-Wong 2021) and Off-CLCB (Liu et al. 2025).

3 Problem Setting

We model the general discrete resource allocation problem as a Knapsack-Constrained Combinatorial Multi-Armed Bandit with a discrete budget space (Badanidiyuru, Kleinberg, and Slivkins 2013). At each round t, the agent selects an allocation vector

$$\mathbf{a}_t = (a_{1,t}, \ldots, a_{K,t}) \in \mathbb{N}^K$$

subject to a global budget constraint

$$\sum_{k=1}^{K} a_{k,t} \leq Q,$$

where Q is the total available resource. Each $a_{k,t}$ represents the number of resource units assigned to arm k.

Assigning $a_{k,t}$ units to arm k generates a stochastic reward modeled by a random variable $X_{k,t}$ drawn from an unknown distribution $\mathcal{D}_k$. The realized reward takes the form $f_k(a_{k,t}, X_{k,t})$, and the expected reward of assigning a units to arm k is

$$\mu_{k,a} = \mathbb{E}_{X_{k,t}\sim\mathcal{D}_k}\big[f_k(a, X_{k,t})\big].$$

Let $\boldsymbol{\mu} = (\mu_{k,a})_{(k,a)\in S}$ denote the vector of expected arm–allocation rewards. For simplicity, we assume rewards across arms are independent, and denote the joint distribution by $\mathbf{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_K)$.

Let A_t be the set of all feasible allocations at round t. The expected reward of choosing allocation $\mathbf{a}_t$ is

$$r(\mathbf{a}_t, \mathbf{D}) = \mathbb{E}\left[\sum_{k=1}^{K} f_k(a_{k,t}, X_{k,t})\right] = \sum_{k=1}^{K}\sum_{a\in A_t} \mu_{k,a}\,\mathbf{1}[a_{k,t} = a].$$

For an algorithm π, let $\mathbf{a}_t^{\pi}$ be the allocation it selects at round t. The optimal expected reward at round t is

$$\mathbf{opt}_t = \max_{\mathbf{a}_t\in A_t} r(\mathbf{a}_t, \mathbf{D}),$$

and we denote the global optimum by

$$\mathbf{opt} = \max_t \mathbf{opt}_t.$$

Following prior work, we assume access to an offline (α, β)-approximation oracle $\mathcal{O}$ that, given $\boldsymbol{\mu}$, outputs an allocation $\mathbf{a}_t^{\mathcal{O}}$ satisfying

$$\Pr\big\{r(\mathbf{a}_t^{\mathcal{O}}, \mathbf{D}) \geq \alpha \cdot \mathbf{opt}_t\big\} \geq \beta$$

(Chen, Wang, and Yuan 2013; Zuo and Joe-Wong 2021; Liu et al. 2025). In other words, the oracle provides an allocation whose reward is at least an α fraction of the optimal reward with confidence β.

The agent's objective is to learn a policy π' that selects allocations $\mathbf{a}_t^{\pi'}$ over T rounds so as to minimize the cumulative (α, β)-approximation regret:

$$Reg_{\pi}^{\alpha,\beta}(T; \mathbf{D}) = T \cdot \alpha \cdot \beta \cdot \mathbf{opt}(\mathbf{D}) - \sum_{t=1}^{T} r(\mathbf{a}_t^{\pi}, \mathbf{D}).$$

4 HybCLCB-DRA Algorithm

In this section, we introduce our hybrid algorithm, HybCLCB-DRA, which integrates offline pessimistic initialization with online refinement. The core idea is straightforward: use offline data to build conservative reward estimates that warm-start the learner, then allow the online phase to continuously update these estimates as new observations arrive. This provides good initial coverage of the arm–weight space while retaining the ability to adapt to nonstationary reward patterns.

Both the offline and online routines maintain empirical means $\hat{\mu}_{i,j}$ and corresponding lower confidence bounds (LCBs) for every arm–weight pair (i, j). The LCB at any point is given by

$$\ell_{i,j} = \hat{\mu}_{i,j} - \sqrt{\frac{\ln\left(\frac{4Kn}{1-\beta}\right)}{N_{i,j}}},$$

where $N_{i,j}$ is the number of times (i, j) has been observed, n is the offline dataset size, K is the number of arms, and β is the success probability of the approximation oracle. Iterating through the offline dataset produces initial estimates (ℓ, μ, N) that reflect a pessimistic view of the true reward landscape. These estimates warm-start the online learner.

In the online phase, HybCLCB-DRA repeatedly queries the (α, β) approximation oracle using the current LCB table to obtain an allocation for each round. After executing

the allocation, the learner incorporates the newly observed rewards and updates both the empirical means and the lower bounds. This structure ensures that the learner uses the offline initialization to avoid overestimation while still improving its estimates as additional data is gathered.

By combining pessimistic offline initialization with continuous online refinement, HybCLCB-DRA balances caution with adaptivity and avoids the pitfalls of purely optimistic exploration in large combinatorial spaces.

Algorithm 1 OffCLCB-DRA: Offline Warm-Start for Hybrid CLCB-DRA

Require: Budget Q, (α, β) approximation oracle $\mathcal{O}$, offline dataset $\mathcal{D} = \{(S_t, (X_{i,t})_i)\}_{t=1}^{n}$
Ensure: Initial estimates $\boldsymbol{\ell}, \boldsymbol{\mu}, \boldsymbol{N}$

Initialize $N_{i,j} \leftarrow 0$, $\hat{\mu}_{i,j} \leftarrow 0$, $\ell_{i,j} \leftarrow -\infty$ for all (i,j)
$K \leftarrow |\{i\}|$, $n \leftarrow |\mathcal{D}|$
for $(S_t, (X_{i,t})_i) \in \mathcal{D}$ **do**
  for $(i,j) \in S_t$ **do**
    $r_{i,t} \leftarrow X_{i,t}$
    $N_{i,j} \leftarrow N_{i,j} + 1$
    $\hat{\mu}_{i,j} \leftarrow \hat{\mu}_{i,j} + \dfrac{r_{i,t} - \hat{\mu}_{i,j}}{N_{i,j}}$
    $\ell_{i,j} \leftarrow \hat{\mu}_{i,j} - \sqrt{\dfrac{1}{2N_{i,j}} \ln\Big(\dfrac{4Kn}{1-\beta}\Big)}$
  end for
end for
return $\boldsymbol{\ell}, \boldsymbol{\mu}, \boldsymbol{N}$

Algorithm 2 HybCLCB-DRA: Hybrid Combinatorial Lower Confidence Bound Algorithm for Discrete Resource Allocation

Require: Budget Q, (α, β) oracle $\mathcal{O}$, warm-start estimates $(\boldsymbol{\ell}, \boldsymbol{\mu}, \boldsymbol{N})$
Ensure: Sequence of allocations $\boldsymbol{a}_t$

$\boldsymbol{a} \leftarrow \mathcal{O}(\boldsymbol{\ell}, Q)$ ▷ initial allocation from warm start
for $t = 1, 2, \ldots$ **do**
  Execute $\boldsymbol{a}$ and observe rewards $r_{i,t}$
  for each arm i with chosen weight j **do**
    $N_{i,j} \leftarrow N_{i,j} + 1$
    $\hat{\mu}_{i,j} \leftarrow \hat{\mu}_{i,j} + \dfrac{r_{i,t} - \hat{\mu}_{i,j}}{N_{i,j}}$
    $\ell_{i,j} \leftarrow \hat{\mu}_{i,j} - \sqrt{\dfrac{1}{2N_{i,j}} \ln(4K)}$
  end for
  $\boldsymbol{a} \leftarrow \mathcal{O}(\boldsymbol{\ell}, Q)$ ▷ update allocation for next round
end for

5 FAIR-HybCLCB-DRA Algorithm

In real resource-allocation settings, the decision maker often imposes explicit fairness or policy constraints on how resources may be distributed. These may include parity-style requirements across protected groups, minimum service guarantees of the form $a_i \geq m_i$, maximum exposure limits such as $a_i \leq M_i$, or broader group-level and regional fairness constraints that restrict how allocations can vary across subsets of arms.

These constraints do not change the bandit learning rule. Instead, they restrict which allocations the agent is allowed to select.

5.1 Fair Oracle for HybCLCB-DRA

Let $\mathcal{A}(Q)$ denote the set of all feasible allocations under a fixed budget Q across K base arms:

$$\mathcal{A}(Q) = \big\{\boldsymbol{a} = (a_1, \ldots, a_K) \in \{0{:}Q\}^K : \sum_{i=1}^{K} a_i = Q\big\}.$$

In many practical resource-allocation settings, the decision maker imposes additional structure or policy constraints on how resources may be distributed. These may include per-arm lower and upper bounds ($m_i \leq a_i \leq M_i$), parity relationships such as $a_i = a_j$, group-level fairness constraints, exposure or coverage limits, or other linear, convex, or discrete restrictions.

To describe these constraints in a unified way, let Θ denote any problem-specific parameters and define the induced feasible set

$$\mathcal{F}(\Theta) = \big\{\boldsymbol{a} \in \{0{:}Q\}^K : \sum_{i} a_i = Q,\ g_k(\boldsymbol{a}; \Theta) \leq 0,\ h_\ell(\boldsymbol{a}; \Theta) = 0\big\},$$

where the functions g_k and h_ℓ capture general inequality and equality conditions determined by Θ. The agent is therefore restricted to choosing allocations in $\mathcal{F}(\Theta)$.

Given the LCB table $\boldsymbol{\ell} = (\ell_{i,j})_{i,j}$, the fair oracle selects a feasible allocation that maximizes the total lower confidence bound:

$$\mathcal{O}_\Theta(\boldsymbol{\ell}, Q) \in \arg\max_{\boldsymbol{a} \in \mathcal{F}(\Theta)} \sum_{i=1}^{K} \ell_{i,a_i}.$$

This is the optimization subroutine solved at every round of HybCLCB-DRA. All fairness, structural, and policy requirements are absorbed into $\mathcal{F}(\Theta)$, while the algorithm itself only updates the LCB estimates that guide the oracle.

Algorithm 3 FAIR-HybCLCB-DRA: Hybrid CMAB with General Constrained Oracle

Require: Budget Q, constraint parameters θ, constrained oracle $\mathcal{O}_\theta$, initial estimates $\{N_{i,j}, \hat{\mu}_{i,j}\}$ from warm start
Ensure: Continually updated estimates and constrained allocations

$\boldsymbol{a} \leftarrow \mathcal{O}_\theta(\boldsymbol{\ell}, Q)$
for $t = 1, 2, \ldots$ **do**
  Allocate $\boldsymbol{a}$ and observe realized rewards $r_{i,t}$
  for each arm i with allocated weight j **do**
    $N_{i,j} \leftarrow N_{i,j} + 1$
    $\hat{\mu}_{i,j} \leftarrow \hat{\mu}_{i,j} + \dfrac{r_{i,t} - \hat{\mu}_{i,j}}{N_{i,j}}$
    $\ell_{i,j} \leftarrow \hat{\mu}_{i,j} - \sqrt{\dfrac{1}{2N_{i,j}} \ln(4K)}$
  end for
  $\boldsymbol{a} \leftarrow \mathcal{O}_\theta(\boldsymbol{\ell}, Q)$ ▷ new fair allocation for next round
end for

6 Experiments

6.1 Learning to Allocate Resources in a Synthetic Environment

To evaluate our algorithms in a structured and heterogeneous setting, we construct a synthetic resource-allocation environment derived from publicly available datasets from the City of Chicago. Although our theoretical results apply generally, the Chicago data provide a realistic template for modeling variation across arms and for generating a controlled benchmark.

6.2 Datasets

We use two sources of information to build the synthetic environment: (1) historical crime data (City of Chicago 2025b), and (2) ACS 5-year demographic statistics (City of Chicago 2025a).

The crime dataset contains incident-level records across all community areas. We convert each incident into a severity score using a fixed weighting scheme As shown in Table 1, and compute, for each community area, the empirical mean and variance of its aggregated daily crime score. These summary statistics form the basis for generating synthetic rewards.

The ACS data are used only to define optional fairness groupings in the constrained oracle (for example, grouping arms by demographic or socioeconomic similarity). They are not used to estimate reward distributions.

6.3 Synthetic Reward Generation and TabNet Regression Model

To create a realistic but controllable environment, we use the historical mean and variance of each community's crime score to build a synthetic dataset of allocation–reward pairs. For each allocation vector $\mathbf{a}$, we generate a corresponding total crime score by sampling from the community-specific distributions implied by the summary statistics. This yields a large supervised dataset of the form

$$(\mathbf{a},\ r(\mathbf{a})),$$

where $r(\mathbf{a})$ is a synthetic reward representing the expected crime score under allocation $\mathbf{a}$.

We then train a TabNet regression model on this dataset. The purpose of TabNet is to learn a smooth approximation of the expected reward as a function of the allocation:

$$\widehat{r}(\mathbf{a}) \approx \mathbb{E}[r(\mathbf{a})].$$

The learned TabNet model serves as the reward oracle inside our `Gymnasium` environment. When an algorithm proposes an allocation $\mathbf{a}$, the environment queries the model and returns a stochastic reward by sampling around TabNet's prediction. This produces an environment that is both data-informed and computationally efficient, while avoiding the need to simulate raw crime incidents.

6.4 Virtual Environment Using Gymnasium

The environment is implemented in `Gymnasium`. Each community area i is associated with a synthetic reward distribution parameterized by the TabNet prediction and an estimated variance. When an allocation $\mathbf{a}$ is submitted, the environment draws a reward by sampling once per arm and summing the results. This produces a stochastic reward landscape with realistic heterogeneity across arms.

The reward-generation mechanism is summarized below:

Algorithm 4 Compute Synthetic Reward for Allocation $\mathbf{a}$

Require: TabNet predictor f_{tab}, allocation $\mathbf{a}$
Ensure: Reward $R(\mathbf{a})$
1: $\mu \leftarrow f_{\text{tab}}(\mathbf{a})$
2: Draw $\epsilon \sim \mathcal{N}(0, \sigma^2(\mathbf{a}))$ ▷ σ^2 estimated from historical variance
3: **return** $R(\mathbf{a}) = \mu + \epsilon$

This design enables consistent evaluation of offline, online, and hybrid CMAB algorithms under the same data-driven reward structure.

6.5 Approximation Oracle

The constrained allocation problem reduces to a multiple-choice knapsack instance, which can be solved exactly in polynomial time. In all experiments, we therefore treat the oracle as exact, corresponding to $\alpha = \beta = 1$.

6.6 Results

We compare three methods, each using the same oracle: (1) a random allocation baseline, (2) CUCB-DRA (Zuo and Joe-Wong 2021), and (3) our novel HybCLCB-DRA algorithm.

Across 20000 simulated episodes, the offline-initialized method achieves a higher total reward and converges more rapidly, and has a lower cumilative regret. This highlights the benefit of pessimistic warm-starting in large combinatorial spaces. Results are shown in Figure 1, Figure 2

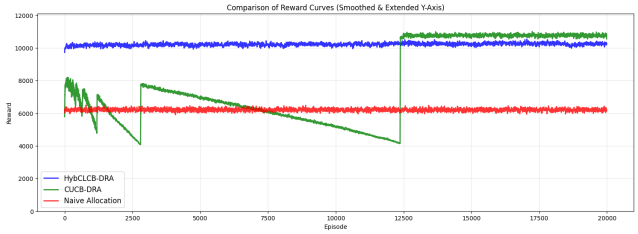

Figure 1: Reward curves for the random baseline, CUCB-DRA, and our HybCLCB-DRA algorithm.

Code Availability

The full implementation of HybCLCB-DRA and FAIR-HybCLCB-DRA, along with the synthetic data generator and experimental code, is available at:
`https://github.com/Wafik20/Offline-RL-for-Resource-Allocation`

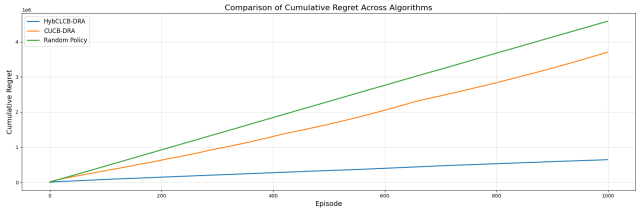

Figure 2: Cumilative regret curves for the random baseline, CUCB-DRA, and our HybCLCB-DRA algorithm across 1000 episodes.

References

Badanidiyuru, A.; Kleinberg, R.; and Slivkins, A. 2013. Bandits with knapsacks. *CoRR* abs/1305.2545.

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, PMLR, 151–159.

City of Chicago. 2025a. Acs 5-year data by community area (selected acs 2023 variables). Metadata updated June 7, 2025; Accessed: 2025-09-17.

City of Chicago. 2025b. Crimes – 2001 to present. Accessed: 2025-09-17.

Gupta, S.; Zuo, J.; Joe-Wong, C.; Joshi, G.; and Yağan, O. 2022. Correlated combinatorial bandits for online resource allocation. In *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '22, 91–100. New York, NY, USA: Association for Computing Machinery.

Jin, Y.; Yang, Z.; and Wang, Z. 2020. Is pessimism provably efficient for offline rl? *CoRR* abs/2012.15085.

Liu, X.; Dai, X.; Zuo, J.; Wang, S.; Joe-Wong, C.; Lui, J. C. S.; and Chen, W. 2025. Offline learning for combinatorial multi-armed bandits. *arXiv preprint arXiv:2501.19300*.

Shivaswamy, P., and Joachims, T. 2012. Multi-armed bandit problems with history. In Lawrence, N. D., and Girolami, M., eds., *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, 1046–1054. La Palma, Canary Islands: PMLR.

Song, Y.; Zhou, Y.; Sekhari, A.; Bagnell, J. A.; Krishnamurthy, A.; and Sun, W. 2023. Hybrid rl: Using both offline and online data can make rl efficient.

Zuo, J., and Joe-Wong, C. 2021. Combinatorial multi-armed bandits for resource allocation. *CoRR* abs/2105.04373.

Table 1: Crime severity weights based on FBI codes. Higher weights indicate more severe crimes.

FBI Code	Description	Weight
01A	Homicide	1000
02	Criminal Sexual Assault	800
03	Robbery	250
04A	Aggravated Assault	200
04B	Aggravated Battery	200
09	Arson	150
05	Burglary	75
15	Weapons Violation	60
07	Motor Vehicle Theft	50
18	Narcotics	40
06	Larceny/Theft	20
08A	Simple Assault	15
08B	Simple Battery	15
17	Sex Offenses	15
14	Vandalism	5
11	Fraud	5
13	Stolen Property	5
24	Disorderly Conduct	1
26	All Other Offenses	1

Algorithm 5 Optimal Beat Allocation Oracle

Require: Reward table ℓ of size $n \times (Q+1)$, where $\ell[i][j]$ is the reward for assigning j beats to community i; total beats Q

Ensure: Allocation vector $\mathbf{a}$ and maximum total reward $R_{\max}$

Initialize DP array $DP[0 \dots Q] \leftarrow -\infty$, set $DP[0] \leftarrow 0$
Initialize Choice table $Choice[0 \dots n-1][0 \dots Q] \leftarrow 0$
for $i = 0$ **to** $n-1$ **do**
  $DP_{\text{new}}[0 \dots Q] \leftarrow -\infty$
  for $t = 0$ **to** Q **do**
    for $j = 0$ **to** $\min(t, \text{length}(\ell[i]) - 1)$ **do**
      $val \leftarrow DP[t-j] + \ell[i][j]$
      if $val > DP_{\text{new}}[t]$ **then**
        $DP_{\text{new}}[t] \leftarrow val$
        $Choice[i][t] \leftarrow j$
      end if
    end for
  end for
  $DP \leftarrow DP_{\text{new}}$
end for
Recover allocation $\mathbf{a} \leftarrow [0, \dots, 0]$, remaining beats $r \leftarrow Q$
for $i = n-1$ **downto** 0 **do**
  $\mathbf{a}[i] \leftarrow Choice[i][r]$
  $r \leftarrow r - \mathbf{a}[i]$
end for
$R_{\max} \leftarrow DP[Q]$
return $\mathbf{a}, R_{\max}$