计算机科学
COMPUTER SCIENCE

基于预算时变的多臂赌博机模型

林宝玲, 贾日恒, 林飞龙, 郑忠龙, 李明禄

引用本文

林宝玲, 贾日恒, 林飞龙, 郑忠龙, 李明禄基于预算时变的多臂赌博机模型[J]. 计算机科学, 2022, 49(11A): 210800212-6.

LIN Bao-ling, JIA Ri-heng, LIN Fei-long, ZHENG Zhong-long, LI Ming-lu. Multi-armed Bandit Model Based on Time-variant Budgets [J]. Computer Science, 2022, 49(11A): 210800212-6.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210800212

基于预算时变的多臂赌博机模型

林宝玲　贾日恒　林飞龙　郑忠龙　李明禄
浙江师范大学数学与计算机科学学院　浙江 金华 321004
(18205776899@163.com)

摘　要　目前已有很多有关预算的多臂赌博机模型,但这些模型能解决的实际问题具有局限性,即这些问题必须都是全程受一个总预算限制。对此,文中提出基于预算时变的多臂赌博机模型,该模型能够打破这种局限性,并被用于解决其他更多的实际问题。该模型抓住了学习者每一轮的动作都受到相应这一轮预算限制的情况。更具体地说,每一轮,玩家都需要在相应这一轮预算的限制下选择拉 $L(L\geqslant 1)$ 个臂(L 不是一个固定值)。玩家的目标就是在每一轮预算的限制下,最大化总的平均奖励。根据这个模型,文中提出基于置信界的动态规划算法。该算法利用模型的特点,每一轮都以臂的经验平均奖励的置信上界为依据,然后使用动态规划算法进行拉臂操作。文中进一步引入遗憾的概念,并从理论上推导得出该算法遗憾的上界与最终预算的总和存在一定的关系。最后,通过实验,将所提算法在不同场景下和其他几个传统的预算受限的多臂赌博机算法(ε-first,KUBE,BTS)进行比较,验证了所提算法的可行性。

关键词:多臂赌博机;预算时变;经验平均奖励;动态规划;遗憾
中图法分类号　TP301

Multi-armed Bandit Model Based on Time-variant Budgets

LIN Bao-ling,JIA Ri-heng,LIN Fei-long,ZHENG Zhong-long and LI Ming-lu
College of Mathematics and Computer Science,Zhejiang Normal University,Jinhua,Zhejiang 321004,China

Abstract　Many budget-related multi-armed bandit models have been proposed,but the practical problems they can solve are limited,that is,they must all be subject to a total budget limit. Therefore,this paper proposes a multi-armed bandit model based on time-variant budgets,which can break this limitation and be used to solve other practical problems. The model captures the situation where the learner's actions for each round are limited by the corresponding round budget. More specifically,at each round,the player is required to choose to pull $L(L\geqslant 1)$ arms(L is not a fixed value) within the budget limits of that round. The player's goal is to maximize the total average reward within the budget limits of each round. According to this model,a dynamic programming algorithm based on confidence bound is proposed. The algorithm takes advantage of the characteristics of the model,takes the confidence upper bound of the empirical average reward of the arm as the basis for each round,and then uses the dynamic programming algorithm to perform the arm pull operation. In this paper,the concept of regret is introduced,and it is deduced theoretically that there is a relationship between the upper bound of regret and the sum of budget. Finally,the feasibility of the proposed algorithm is verified by comparing the proposed algorithm with other traditional budget-limited multi-armed bandit algorithms(ε-first,KUBE,BTS) under different scenarios.

Keywords　Multi-armed bandit,Time-variant budgets,Empirical average reward,Dynamic programming,Regret

1　引言

多臂赌博机(Multi-armed Bandit,MAB)问题是一个典型的决策理论问题。该问题的模型可被用于解决很多实际问题,如推荐系统[1]、在线核选择问题[2]和 RFID 多阅读器信道资源分配问题[3]等。随机多臂赌博机(Stochastic MAB,SMAB)问题又是 MAB 问题中一类经典的问题[4-5]。问题具体描述是:一共有 k 个臂,玩家每次拉臂都会获得一定的奖励。每个臂所对应的奖励是服从一定分布的,只是具体服从的分布是未知的。玩家的目标就是能够在这种情况下最终获得最大的奖励总和。

在 SMAB 问题研究的基础上,由于实际问题需要,出现了很多其他的多臂赌博机问题,比如预先观察的多臂赌博机问题(Multi-armed Bandit with Pre-observations)[6]、提供补偿的多臂赌博机问题(Multi-armed Bandit with Compensation)[7]、实现公平的多臂赌博机问题(FAIR-MAB)[8]、上下文多臂赌博机问题(Contextual Multi-armed Bandit)[9-10]等。除此之外,由于在现实生活中做出动作时(租用线上广告横幅等)往往需要付出代价,因此出现了许多有关预算的多臂赌博机问题,包括纯探索多臂赌博机问题(Pure-exploration Bandit)[11-13]、预算受限的多臂赌博机问题(Budget-limited Multi-armed Bandit)[14-15]、Multi-armed Bandit with Constraint

基金项目:国家自然科学基金(61902358)
This work was supported by the National Natural Science Foundation of China(61902358).
通信作者:贾日恒(rihengjia@zjnu.edu.cn)

Budget and Vaiable Costs[16-17] 和 Budget-constraint Multi-armed Bandit with Multiple Plays[18-19]。

上述有关预算的多臂赌博机模型能够解决现实生活中的很多问题，但是存在局限性。更具体地说，那些能被解决的实际问题必须都是全程只受一个总预算限制。为了打破这种局限性，文中提出基于预算时变的多臂赌博机模型。该模型抓住了学习者每一轮的动作都受到相应这一轮预算限制的情况，而不再是全程只受一个总预算限制。也就是说，每一轮，玩家都需要在相应这一轮预算的限制下选择拉 $L(L\geqslant 1)$ 个臂（L 不是一个固定值）。玩家的目标就是在每一轮预算的限制下，最大化总的平均奖励。值得注意的是，各轮之间是独立的，即使某一轮的预算没有被用完，也不能累加到对应的下一轮的预算之中。

根据这个模型，文中提出基于置信界的动态规划算法。具体地，文中将问题分为两种情况考虑。首先考虑每个臂的奖励分布是已知的情况。在这种情况下，由于每一轮臂的选择问题可以被看成一个 0-1 背包问题，故文中选择采用动态规划算法。其中，0-1 背包问题被具体描述为：给定 n 个重量分别为 $w_1, w_2, \cdots, w_n$，价值分别为 $v_1, v_2, \cdots, v_n$ 的物品和容量为 C 的背包，求这个物品中一个最有价值的子集，使得在满足背包容量的前提下，包内的总价值最大。很明显地，模型中每一轮受到的限制就相当于 0-1 背包问题里的容量 C，每个臂所对应的固定的代价及已知的平均奖励大小就相当于 0-1 背包问题里物品的重量及价值。因此，0-1 背包问题里所求的最有价值的物品子集就相当于每一轮里要拉的最理想的臂的组合。然后再考虑每个臂的奖励分布是未知的情况。在这种情况下，一味地利用目前有限次拉臂所获得的臂的信息会导致结果不好。为了防止该情况发生，文中引入了臂的经验平均奖励（Empirical Average Reward）的置信上界值这个概念并以此为拉臂的依据。也就是说，每个臂的经验平均奖励（Empirical Average Reward）的置信上界值就代表了此时每个臂被认为的好坏程度。拥有了这个值，问题就可以被转化成第一种情况来考虑。这样，就有了基于置信界的动态规划算法。

文中进一步引入遗憾的概念，并从理论上推导得出该算法遗憾的上界与最终预算的总和存在一定的关系。对于遗憾，算法产生遗憾的大小是衡量一个多臂赌博机算法性能的重要指标。更具体地，算法产生的遗憾越大，则证明该算法性能越不好。最后，文中通过实验验证了所提算法的可行性。本篇文章的主要贡献在两方面：

(1)区别于已有的大多数相关研究文献，文中创新性地提出了基于预算受限的多臂赌博机模型，该模型能够解决现有有关预算的多臂赌博机模型没能解决的实际问题；

(2)文中针对基于预算时变的多臂赌博机模型提出了一个算法，即基于置信界的动态规划算法，而且通过实验验证了算法的可行性。

2 基于预算时变的多臂赌博机模型的建立

本节描述基于预算时变的多臂赌博机模型。更具体地说，文中考虑的是一个拥有 $K(K\geqslant 2)$ 个臂的多臂赌博机问题。在每一轮，玩家都会先获得相应这一轮的预算值，文中定义第 t 轮获得的预算值为 B_t。其中，B_t 来自于一个分布。然后，在每一轮，玩家需要选择拉 $L(L\geqslant 1)$ 个臂。这里需要注意的是，L 并不是固定的一个值。这是因为每一轮获得的预算值很可能是不一样的，而不同的预算值又会使得每一轮拉的臂数存在差异。文中定义第 t 轮拉的臂的组合为 I_t。当玩家在第 t 轮拉了臂 $i\in[K]$（$[K]$ 被定义为集合 $\{1,2,\cdots,K\}$），则需要付出一个固定的代价值 $c_i(c_i>1)$，即第 i 个臂对应的固定的代价值。同时，玩家也会获得一个随机的奖励值 $r_i(t)(r_i(t)\in[0,1])$。其中，$r_i(t)$ 来自臂 i 的奖励分布。对于预算值 B_t，很明显地，第 t 轮拉的臂的代价总和是不允许超出这个值的。此外，每一轮的预算值是独立于其他任何轮的预算值的，即使某一轮的预算值没有被消耗完，也是不允许累加到相应下一轮的预算值中去。然后，文中定义一共有 T 轮，那么总的预算值就可以表示为 $\sum_{t=1}^{T}B_t$。因为最开始，玩家并不知道每个臂的真实平均奖励（文中定义第 i 个臂的真实平均奖励为 u_i），所以玩家必须去学习臂的情况。玩家的目标就是在满足每一轮拉的臂的代价总和不超过每一轮对应的预算值的条件下，找到最优的拉臂的序列，最大化最终获得的奖励总和。

根据前面的描述，在第 t 轮，则有：

$$\sum_{i\in I_t}c_i\leqslant B_t$$

定义 R_t 为第 t 轮拉臂的组合 I_t 获得的总的奖励，那么 R_t 的期望就表示为：

$$E[R_t]=\sum_{i\in I_t}u_i$$

然后，I_t^* 被定义为第 t 轮最理想的臂的组合，它能够最大化第 t 轮所获得的平均奖励。此时将第 t 轮拉 I_t^* 这个臂的组合获得的奖励定义为 R_t^*，那么为了去确定 I_t^*，玩家必须提前知道每个臂的 u_i。而实际上，玩家不可能会提前知道每个臂的奖励分布情况。因此，I_t^* 只是理论上的一个最理想的臂的组合，在实际上是无法获得的。

文中引入遗憾的概念，其中，第 t 轮获得的遗憾的含义指的是第 t 轮根据算法拉 I_t 这个臂的组合获得的平均奖励与第 t 轮拉 I_t^* 这个臂的组合获得的平均奖励的差别。从遗憾的含义中可看出，遗憾代表着算法在执行过程中的损失。算法产生的遗憾越小，损失则越小，算法性能就比较好。故算法产生遗憾的大小是衡量多臂赌博机算法性能的重要指标。第 t 轮获得的遗憾在文中被定义为 $regret(t)$，则有：

$$regret(t)=E[R_t^*]-E[R_t]$$

此外，文中定义 T 轮获得的总的遗憾为 regret，则：

$$regret=\sum_{t=1}^{T}regret(t)=\sum_{t=1}^{T}(E[R_t^*]-E[R_t])$$

因此，玩家的目标也可以被认为是找到好的拉臂序列，最小化 T 轮产生的总的遗憾。

3 算法描述

根据基于预算时变的多臂赌博机模型，文中提出基于置信界的动态规划算法。在每一轮，算法首先计算出每个臂的经验平均奖励的置信上界值；然后依据这个值，用动态规划算法选出这一轮所要拉的臂的组合。接下来，首先具体考虑当所有臂的奖励分布情况已知时拉臂的策略，即要找到每一轮的 I_t^* 做出的拉臂的策略，再具体考虑当所有臂的奖励分布未知时的拉臂策略。

3.1 臂的奖励分布已知时的算法

此时，臂 i 的平均奖励 u_i 和其固定的代价 c_i，$i\in[K]$ 是已知的。在每一轮，玩家的动作都是会受到相应这一轮的预算值

的限制；并且每一轮的预算值之间是独立的，并不存在某一轮的剩余预算可以累加到下一轮的情况。很明显地，此时每一轮的拉臂问题就相当于一个 0-1 背包问题。那么，在第 t（$t=1,2,\cdots,T$）轮，为了获得 I_t^*，文中采用动态规划算法。具体算法如算法 1 所示。

算法 1 奖励分布已知时的动态规划算法

输入：u_i，c_i，$i\in[K]$，B_t，$t\in[T]$

1. for t→1：T do
2. valueExcel[i][j]=0，∀i∈{0,1,…,K}，j∈{0,1,…,B_t}；
3. for i→1：K do
4. for j→1：B_t do
5. valueExcel[i][j]=valueExcel[i−1][j]；
6. if j>=c_i：
7. valueExcel[i][j]=max(valueExcel[i−1][j−c_i]+u_i，valueExcel[i][j])；
8. end if
9. end for
10. end for
11. 根据求出的二维数组 valueExcel 获得I_t^*；
12. 在第 t 轮拉臂的组合I_t^*；
13. end for

算法 1 中的 $valueExcel[i][j]$ 表示在允许拉的臂的臂号范围是 $[1,i]$ 且预算值为 j 的情况下，能获得的最大奖励值（最大奖励值是由臂的已知的平均奖励组成）。因此，对于 $t\in[T]$，有：

$$valueExcel[K][B_t]=\sum_{i\in I_t^*}u_i$$

此外，算法 1 中的步骤 11，要想获得 I_t^*，即在预算受限的情况下，得出是哪个臂的组合达到了最大奖励值 $valueExcel[K][B_t]$，只需将步骤 2－10 求得的二维数组 $valueExcel$ 反向推导即可。具体求解算法如算法 2 所示。最后，因为算法 1 中每一轮求出二维数组 $valueExcel$ 都需要一个双重 for 循环，故算法中每一轮的时间复杂度为 $O(KB_t)$。

算法 2 依据二维数组 $valueExcel$ 获得I_t^*

1. i=K，j=B_t；
2. while i>0 and j>0：
3. if valueExcel[i][j]=valueExcel[i−1][j−c_i]+u_i：
4. 输出 i；j=j−c_i；
5. end if
6. i=i−1；
7. end while

可以看出，算法 2 中步骤 4 输出的 i 构成了I_t^*。

3.2 基于置信界的动态规划算法

此时，每个臂的奖励分布情况是未知的。只有在某个臂被拉之后，才能观察到拉这个臂所产生的奖励大小。在这种情况下，文中提出的算法是直接明了的：在每一轮，首先借助这一轮之前拉臂观察到的臂的奖励值对每个臂的真实平均奖励值进行估计。估计得出来的每个臂的真实平均奖励值是这一轮拉臂的重要依据，它的作用就如同臂的奖励分布已知时的每个臂的真实平均奖励值一样。因此，当每一轮拥有了这个值时，每轮的拉臂问题就可以被看成一个 0-1 背包问题并可通过动态规划算法求解。具体做法为：在每一轮，用估计得到的第 i 个臂的真实平均奖励值替换算法 1 中的 u_i，$i\in[K]$；然后执行算法 1 中的步骤 2－11，即算法 1 中的第一个 for 循环只执行一次，这样就能得到算法认为的这一轮最好的臂的组合 I_t。

现在来定义一些新的变量。对任何臂 $i\in[K]$，定义 $T_i(t)$ 为臂 i 到第 t 轮时被拉的次数总和，$u_i(t)$ 为臂 i 到第 t 轮时的经验平均奖励（Empirical Average Reward），$\theta_{i,t}^k$ 为臂 i 在第 t 时刻的置信项。具体地，则有：

$$T_i(t)=\sum_{n=1}^{t}\mathbb{I}\{i\in I_n\}$$

$$u_i(t)=\frac{1}{T_i(t)}\sum_{n=1}^{t}r_i(n)\mathbb{I}\{i\in I_n\},\theta_{i,t}^k=\sqrt{\frac{k\lg(t-1)}{T_i(t-1)}}$$

其中，k 是一个正的参数，它让算法变得更灵活。$\mathbb{I}\{\cdot\}$ 是一个指示函数，只有当事件 E 是真时，$\mathbb{I}\{E\}=1$，否则 $\mathbb{I}\{E\}=0$。

在每一轮，当要估计每个臂的真实平均奖励值时，文中并没有让它直接等于每个臂在这一时刻的经验平均奖励，而是加入了探索的元素，即$\theta_{i,t}^k$。文中定义 $\widetilde{u}_i(t)$ 为经验平均奖励的置信上界值。对于 $\widetilde{u}_i(t)$，其具体意思为第 t 轮所估计的第 i 个臂的真实平均奖励值，且有：

$$\widetilde{u}_i(t)=\min\{u_i(t-1)+\theta_{i,t}^k,1\}$$

考虑到这里，在臂的奖励分布未知时提出的拉臂策略，即基于置信界的动态规划算法，其具体步骤如算法 3 所示。

算法 3 基于置信界的动态规划算法

输入：c_i，$i\in[K]$，B_t，$t\in\{K+1,K+2,\cdots,K+T\}$

1. for t→1：K do
2. pull arm t；
3. end for
4. for t→K+1：T+K do
5. calculate the $T_i(t-1)$，$u_i(t-1)$，$\theta_{i,t}^k$，$\widetilde{u}_i(t)$ for any i，$i\in[K]$；
6. valueExcel[i][j]=0，∀i∈{0,1,…,K}，j∈{0,1,…,B_t}；
7. for i→1：K do
8. for j→1：B_t do
9. valueExcel[i][j]=valueExcel[i−1][j]；
10. if j>=c_i：
11. valueExcel[i][j]=max(valueExcel[i−1][j−c_i]+$\widetilde{u}_i(t)$，valueExcel[i][j])；
12. end if
13. end for
14. end for
15. 根据求出的二维数组 valueExcel 获得 I_t；
16. 在第 t 轮拉臂的组合 I_t；
17. end for

注意，算法 3 中的步骤 1－3 其实是一个初始化的过程。在这个过程中，拉臂并不需要付出代价。只有在步骤 4－17 的时候，每一轮才有预算值，拉臂才需要付出代价。因为真正玩的轮数是 T 轮，因此在步骤 4 中设置了 $t\to K+1:T+K$，即循环执行了 T 次。然后，执行算法 3 中的步骤 4－17 的时候，每一轮获得的预算值与算法 1 中每一轮的预算值是一一对应的。此外，算法 3 中的步骤 15 的具体过程可参考算法 2，这里不再具体表示。

4 算法遗憾上界及证明

前面已经提到，遗憾代表着算法在执行过程中的损失。算法产生的遗憾越小，损失则越小，算法性能就比较好。故算法产生遗憾的大小是衡量多臂赌博机算法性能的重要指标。那么，本节将推导出基于置信界的动态规划算法的遗憾的上界，这是从理论上初步表明该算法性能的一种重要方式。

同时，本节也会清楚地展示出推导的思路。

首先，定义几个新的符号，这些符号在推导的过程中都需要被使用到：

(1) $S_t=\{\{j:x_j=1\}\mid\sum_{j=1}^{K}c_j\mathbb{I}\{x_j=1\}\leqslant B_t;x_j\in\{0,1\};\forall j\in[K]\}$

(2) $C_{s,t}=S_t\backslash\{I_t^*\}$

(3) $e_t^*=\sum_{i\in I_t^*}u_i$

(4) $\frac{u_{imax}}{c_{imax}}=\max_{i\in[K]}\frac{u_i}{c_i}$

(5) $B=\sum_{t=1}^{T}B_t$

(6) $B_{tmin}=\min_{t\in[T]}B_t$

(7) $e_{tmin}^*=\min_{t\in T}\sum_{i\in I_t^*}u_i$

其中，S_t 可以被看成第 t 轮时所有满足预算 B_t 限制的臂的组合的集合。然后，$\frac{u_{imax}}{c_{imax}}$ 这个值所对应的臂是理论上最好的臂，即在消耗相同预算的时候，该臂能产生比其他任何臂都大的平均奖励。B 指 T 轮的总预算值。又因为每一轮的预算值不可能都是一样的，因此就存在最小值，于是文中用符号 B_{tmin} 来表示这个最小值。同样地，定义 e_t^* ($t\in[T]$) 为这 T 个值里的最小值为 e_{tmin}^*。

本文第 2 节已经提出算法的遗憾表示为：

$$regret=\sum_{t=1}^{T}regret(t)=\sum_{t=1}^{T}(E[R_t^*]-E[R_t])$$

为了更好地得出 $regret$ 的上界，文中将 $regret$ 分成两部分来分析：$\sum_{t=1}^{T}E[R_t^*]$ 和 $\sum_{t=1}^{T}E[R_t]$。

首先，对于第一部分 $\sum_{t=1}^{T}E[R_t^*]$，它指的是在 T 轮里，当拉臂的序列是 $\{I_1^*,I_2^*,\cdots,I_T^*\}$ 的时候所获得的总的平均奖励值。此外，由于新定义的符号 e_t^* 更能形象具体地表示出在第 t 轮的时候，当所有臂的奖励分布情况已知时所获得的最大平均奖励值，因此根据这个新符号，再加上一些推导，可将 $\sum_{t=1}^{T}E[R_t^*]$ 换成另外一种表示，即为 $\sum_{t=1}^{T}e_t^*$。

证明：$\sum_{t=1}^{T}E[R_t^*]=E[\sum_{t=1}^{T}\sum_{I\in S_t}\sum_{i\in I}r_i(t)\mathbb{I}\{I_t^*=I\}]=E[\sum_{t=1}^{T}\sum_{I\in S_t}\sum_{i\in I_t^*}r_i(t)\mathbb{I}\{I_t^*=I\}]=\sum_{t=1}^{T}\sum_{I\in S_t}E[\sum_{i\in I_t^*}r_i(t)\mathbb{I}\{I_t^*=I\}]=\sum_{t=1}^{T}\sum_{I\in S_t}\sum_{i\in I_t^*}u_iP\{I_t^*=I\}=\sum_{t=1}^{T}\sum_{i\in I_t^*}u_i=\sum_{t=1}^{T}e_t^*$

然后，对于第二部分 $\sum_{t=1}^{T}E[R_t]$，它指的是在 T 轮里，当拉臂的序列是 $\{I_1,I_2,\cdots,I_T\}$ 时最终获得的总的平均奖励值。同样使用一些推导方法，最后得出了第二部分的下界为：

$$e_{tmin}^*-E[\sum_{t=1}^{T}e_t^*\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$$

证明：$\sum_{t=1}^{T}E[R_t]=E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_t\}]+E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_t\}]\geqslant E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]+E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]=E[\sum_{t=1}^{T}\sum_{i\in I_t^*}r_i(t)\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]+E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]=E[\sum_{t=1}^{T}\sum_{i\in I_t^*}u_i\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]+E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]-E[\sum_{t=1}^{T}\sum_{i\in I_t^*}u_i\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]+E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]+E[\sum_{t=1}^{T}\sum_{i\in I_t}u_i\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]=E[\sum_{t=1}^{T}e_t^*\mathbb{I}\{\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]-E[\sum_{t=1}^{T}(e_t^*-\sum_{i\in I_t}u_i)\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]\geqslant e_{tmin}^*-E[\sum_{t=1}^{T}e_t^*\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$

其中，$E[\sum_{t=1}^{T}\sum_{i\in I_t^*}u_i\mathbb{I}\{I_t=I_t^*,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$ 可以被获得，是因为：

$$E[\sum_{t=1}^{T}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]=E[\sum_{t=1}^{T}\sum_{I\in C_{s,t}}\sum_{i\in I_t}r_i(t)\mathbb{I}\{I_t=I,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]=\sum_{t=1}^{T}\sum_{I\in C_{s,t}}E[\sum_{i\in I_t}r_i(t)]P\{I_t=I,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}=\sum_{t=1}^{T}\sum_{I\in C_{s,t}}\sum_{i\in I_t}u_iP\{I_t=I,\sum_{i\in I_t}c_i\leqslant B_{tmin}\}=E[\sum_{t=1}^{T}\sum_{i\in I_t}u_i\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$$

同理，$E[\sum_{t=1}^{T}\sum_{i\in I_t}u_i\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$ 也可以用上述推导方法获得。此外，$e_{tmin}^*-E[\sum_{t=1}^{T}e_t^*\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$ 中能得出 e_{tmin}^*，是因为 T 轮中至少有一轮会满足 $\sum_{i\in I_t}c_i\leqslant B_{tmin}$。

最后，有了上述的结论，$regret$ 的上界就可以被简单表示出来：

$$\sum_{t=1}^{T}e_t^*-e_{tmin}^*+E[\sum\nolimits_{t=1}^{T}e_t^*\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$$

对于上界中的 $E[\sum_{t=1}^{T}e_t^*\mathbb{I}\{I_t\in C_{s,t},\sum_{i\in I_t}c_i\leqslant B_{tmin}\}]$，假设让 $\mathbb{I}\{\cdot\}$ 的值都取 1，那么此时这个式子的值将会达到最大。因此，这个式子的上界值就可以被得出，即为 $\sum_{t=1}^{T}e_t^*$。考虑到这里，算法遗憾的上界又可以被表示为：

$$regret\leqslant 2\sum_{t=1}^{T}e_t^*-e_{tmin}^* \tag{1}$$

从式(1)中看出，如果要让算法遗憾的上界能够更加清晰，$\sum_{t=1}^{T}e_t^*$ 需要被进一步推导。最终根据一些推导，得到如下结论：

$$\sum_{t=1}^{T}e_t^*\leqslant B\frac{u_{imax}}{c_{imax}} \tag{2}$$

证明：$\sum_{t=1}^{T}e_t^*=\sum_{t=1}^{T}\sum_{i\in I_t^*}u_i=\sum_{t=1}^{T}\sum_{i\in I_t^*}c_i\frac{u_i}{c_i}\leqslant\frac{u_{imax}}{c_{imax}}\sum_{t=1}^{T}\sum_{i\in I_t^*}c_i\leqslant\frac{u_{imax}}{c_{imax}}\sum_{t=1}^{T}B_t=B\frac{u_{imax}}{c_{imax}}$

其中，$\sum_{t=1}^{T}\sum_{i\in I_t^*}c_i$ 指的是在理想情况下，即在所有臂的奖励分布情况已知时，拉臂序列为 $\{I_1^*,I_2^*,\cdots,I_T^*\}$ 时产生的总的代价。考虑到这里，通过结合式(1)和式(2)，所提算法的遗憾的上界最终就可以得出来，具体见定理 1。

定理 1 基于置信界的动态规划算法的遗憾的上界为：

$$2B\frac{u_{imax}}{c_{imax}}-e_{tmin}^*$$

注意，这里所提的遗憾指 T 轮里总的遗憾。然后，从定理 1 中可以确切地得出结论，即算法遗憾的上界与最终预算的总和是存在一定的关系的。

5 仿真实验及结果分析

算法产生遗憾的大小是衡量一个多臂赌博机算法性能的重要指标。遗憾指的是算法产生的平均奖励大小与理想情况下产生的平均奖励大小的差别。本文除了会通过遗憾这个指标去衡量基于置信界的动态规划算法的性能，还会把这个算法和其他 3 个传统的预算受限的多臂赌博机算法进行比较，进而验证所提算法的可行性。

具体地，被用于比较的 3 种预算受限的多臂赌博机算法分别是 ε-first 算法、KUBE 算法和 BTS 算法。这 3 种算法虽然不是在基于预算时变的多臂赌博机模型下被提出来的，但也是在有关预算的多臂赌博机模型下被提出来的。因此，通过与 3 种算法的比较进而验证所提算法的可行性是合理的。此外，为了让这 3 种算法能够在新的多臂赌博机模型下发挥更好的性能，使得算法的比较能够更加科学，实验中会把这 3 种算法进行简单调整。

虽然这 3 种算法都被简单调整过，但是并不改变算法最初的思想。首先，对于 ε-first 算法，它将全程的一个总预算分为两部分。其中一部分预算用于探索，剩下的预算则用于利用，即一直拉探索阶段选出来的最好的臂，直到预算不足。那么，算法被调整过后，只是变为在前几轮中一直探索，然后就一直根据探索阶段获得的臂的信息进行拉臂操作。其次，对于 KUBE 算法的调整，实验中保持了该算法在每一轮依靠贪婪算法得出最佳拉臂策略，并根据这个策略来实际赋予每个臂在这一轮被拉的概率的做法。唯一调整的是，根据基于预算时变的多臂赌博机模型的特点，算法在每一轮得出的最佳拉臂策略不再是对应于全局的，而只是对应于该轮的。最后，对于 BTS 算法，在每一轮，是先从每个臂所对应的 beta 分布中抽样出奖励和代价，然后在这一轮中选择拉奖励和代价比值最大的那个臂，最后再根据规则更新每个臂对应的 beta 分布的参数。这里进行调整的是将从 beta 分布中抽样出代价这个步骤取消，因为基于预算时变的多臂赌博机模型中每个臂的代价都是固定的值。

对于实验中要实现算法的比较，具体做法是让调整过后的这 3 种算法与基于置信界的动态规划算法在相同条件下执行，得出图像结果，最后通过图像来比较 4 种算法的性能。

5.1 仿真实验设计

实验设计基于置信界的动态规划算法和其他 3 种传统的预算受限的多臂赌博机算法的比较是在以下 4 种场景进行的：1）臂的奖励服从多项式分布，$K=10$；2）臂的奖励服从多项式分布，$K=30$；3）臂的奖励服从 beta 分布，$K=10$；4）臂的奖励服从 beta 分布，$K=30$。因为在基于预算时变的多臂赌博机模型下预算并不会影响玩的轮数，每一轮的预算只会影响该轮拉臂的臂数，因此玩的总轮数需要被事先设置好。又由于不同场景下的 K 值是不一样的，为了不影响 4 种算法的性能，实验设置不同的玩的轮数序列，具体设置为：在第一种场景和第三种场景下，玩的轮数序列为 {15，30，50，70，90，110，130}；在第二种场景和第四种场景下，玩的轮数序列则为 {30，50，70，90，110，130，150}。然后，为了让实验结果更加准确科学，4 种算法在每种场景下的每个轮数设置中都被运行 100 遍，最后再将结果取平均值并得出最终的图像结果。此外，所提算法里面有一个正的参数 k，根据实验，最终将它设置为 10^{-2}。最后，由于每一轮获得的预算值是来自于一个分布，因此在实验中设置任何场景下每一轮获得的预算是服从于正态分布的（注意，不同场景下预算服从的正态分布的参数是不一样的）。

5.2 仿真结果分析

实验结果如图 1－图 4 所示。图的横坐标为预算大小，纵坐标指在具体预算下某个算法产生的遗憾大小。

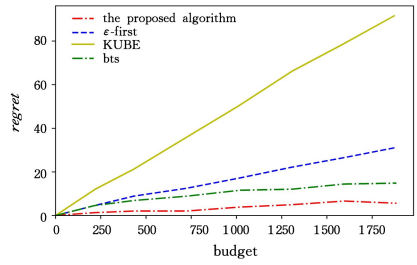

图 1 基于置信界的动态规划算法和其他 3 种算法的比较结果，臂的奖励服从多项式分布，臂的数目为 10

Fig. 1 Comparative result of the proposed algorithm and other three algorithms, multinomial distribution, $K=10$

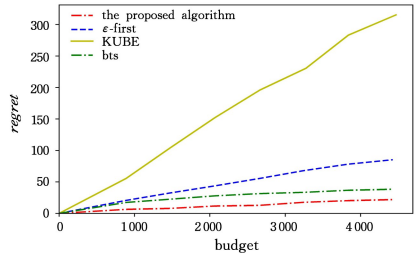

图 2 基于置信界的动态规划算法和其他 3 种算法的比较结果，臂的奖励服从多项式分布，臂的数目为 30

Fig. 2 Comparative result of the proposed algorithm and other three algorithms, multinomial distribution, $K=30$

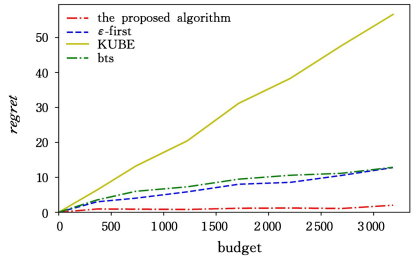

图 3 基于置信界的动态规划算法和其他 3 种算法的比较结果，臂的奖励服从 beta 分布，臂的数目为 10

Fig. 3 Comparative result of the proposed algorithm and other three algorithms, beta distribution, $K=10$

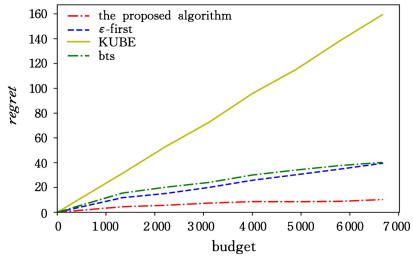

图 4 基于置信界的动态规划算法和其他 3 种算法的比较结果，臂的奖励服从 beta 分布，臂的数目为 30

Fig. 4 Comparative result of the proposed algorithm and other three algorithms, beta distribution, $K=30$

对4个图像进行整体分析，可以得出以下结论。1)不论在哪种场景下，文中所提算法都是优于其他3种算法的。2)不论在哪种场景下，KUBE算法都是4种算法里性能最差的。3)ε-first算法和BTS算法的性能不分上下。具体地，ε-first算法在臂的奖励服从多项式分布时表现得比BTS算法好，但当臂的奖励服从beta分布时表现得却不如BTS算法。对于结论1)和结论2)，因为基于置信界的动态规划算法以臂的经验平均奖励的置信上界值为依据，很好地解决了多臂赌博机模型中要平衡探索和利用的问题，同时，这个算法的随机性不强，故该算法的性能比较好。然而相比之下，KUBE算法的随机性很强，故而会产生更大的遗憾。

另外，不管在哪种场景下，所提算法产生的遗憾都是很小的，这也充分说明基于置信界的动态规划算法的可行性。最后，当把臂的奖励分布固定成某一分布，即比较图1和图2或者是图3和图4时，可以很明显地发现不管是哪一种算法，在$K=30$时产生的遗憾都会比在$K=10$时产生的遗憾更大。这个结果是符合理论的。因为在相同条件下，当臂数更多时，算法就需要对更多的臂进行学习，随机性变强，自然就加大了算法的学习难度，从而导致产生更大的遗憾。

结束语 区别于已有的大多数相关研究文献，本文创新性地提出了基于预算受限的多臂赌博机模型。这个模型最大的特点就是它不是全程只受一个总预算的限制，而是在每一轮都受到这一轮对应的预算限制。玩家的目标就是在每一轮预算的限制下，最大化最终的奖励总和。然后，文中提出了一个基于置信界的动态规划算法。具体地说，这个算法利用基于预算时变的多臂赌博机模型的特点，加入了动态规划操作。同时，这个算法不仅有效地利用了臂的经验平均奖励，而且很好地解决了平衡探索和利用的问题。最后，文中推导出所提算法遗憾的上界，并通过实验验证了所提算法的可行性。

文中提出的基于预算时变的多臂赌博机模型中每个臂对应的代价都是固定的。未来拟在此基础上研究每个臂的代价都是服从于概率分布的情况，进而能让该模型被用于解决其他更多的实际问题。

参 考 文 献

[1] CHEN K. Research on Recommendation System Based on Multi-armed Bandit Algorithm[J]. Journal of Changjiang Information and Communication, 2021, 34(3): 43-46.

[2] LI J F, LIAO S Z. Adversarial Multi-armed Bandit Model with Online Kernel Selection [J]. Computer Science, 2019, 46(1): 57-63.

[3] SHI J, ZHENG J L, YUAN Y, et al. RFID Multi-reader Channel Resources Allocation Algorithm Based on Whittle Index[J]. Computer Science, 2019, 46(10): 122-127.

[4] ERIC S, NICHOLAS T, SAMUEL J. Finding Structure in Multi-armed Bandits[J]. Cognitive Psychology, 2020, 119: 1-35.

[5] ZHANG X F, ZHOU Q, LIANG B, et al. An Adaptive Algorithm in Multi-armed Bandit Problem[J]. Journal of Computer Research and Development, 2019, 56(3): 643-654.

[6] ZUO J H, ZHANG X X, JOE-WONG C. Observe Before Play: Multi-armed Bandit with Pre-observations[C]// Proc of the AAAI Conf on Artificial Intelligence. New York: AAAI, 2020: 7023-7030.

[7] WANG S W, HUANG L B. Multi-armed Bandits with Compensation[C]// Proc of the 32th Conf on Neural Information Processing Systems. montreal: NeurIPS, 2018: 5114-5122.

[8] PATIL V, GHALME G, NAIR V, et al. Achieving Fairness in the Stochastic Multi-armed Bandit Problem[C]// Proc of the AAAI Conf on Artificial Intelligence. New York: AAAI, 2020: 5379-5386.

[9] GUTOWSKI N, AMGHAR T, CAMP O, et al. Context Enhancement for Linear Contextual Multi-armed Bandits[C]// Proc of IEEE Int Conf on Tools with Artificial Intelligence. Volos: IEEE, 2018: 1048-1055.

[10] MANICKAM I, LAN A S, BARANIUK R G. Contextual Multi-armed Bandit Algorithms for Personalized Learning Action Selection[C]// Proc of IEEE Int Conf on Acoustics. New Orleans: IEEE, 2017: 6344-6348.

[11] AZIZM, ANDERTON J, KAUFMANN E, et al. Pure Exploration in Infinitely-armed Bandit Models with Fixed-confidence [C]// Proc of Algorithmic Learning Theory Int Conf. Lanzarote: ALT, 2018: 3-24.

[12] BUBECK S, MUNOS R, STOLTZ G. Pure Exploration in Multi-armed Bandits Problems[C]// Proc of Algorithmic Learning Theory Int Conf. Porto: ALT, 2009: 23-37.

[13] XUE Y, ZHOU P, MAO S W, et al. Pure-exploration Bandits for Channel Selection in Mission-critical Wireless Communications [J]. IEEE Transactions on Vehicular Technology, 2018, 67(11): 10995-11007.

[14] LONG T T, CHAPMAN A, ROGERS A, et al. Knapsack Based Optimal Policies for Budget-limited Multi-armed Bandits[C]// Proc of the AAAI Conf on Artificial Intelligence. Toronto: AAAI, 2012: 1134-1140.

[15] LONG T T, CHAPMAN A, ROGERS A, et al. Epsilon-first Policies for Budget-limited Multi-armed Bandits[C]// Proc of the AAAI Conf on Artificial Intelligence. Atlanta: AAAI, 2010: 1211-1216.

[16] XIA Y C, LI H F, QIN T, et al. Thompson Sampling for Budgeted Multi-armed Bandits[C]// Proc of the 24th International Joint Conf on Artificial Intelligence. Buenos Aires: IJCAI, 2015: 3960-3966.

[17] DING W K, QIN T, ZHANG X D, et al. Multi-armed Bandit with Budget Constraint and Variable Costs[C]// Proc of the AAAI Conf on Artificial Intelligence. Washington: AAAI, 2013: 232-238.

[18] XIA Y C, QIN T, MA W D, et al. Budgeted Multi-armed Bandits with Multipleplays[C]// Proc of the 25th International Joint Conf on Artificial Intelligence. New York: IJCAI, 2016: 2210-2216.

[19] ZHOU D P, TOMLIN C J. Budget-constrained Multi-armed Bandits with Multiple Plays[C]// Proc of the AAAI Conf on Artificial Intelligence. New Orleans: AAAI, 2018: 4572-4579.

LIN Bao-ling, born in 1998, postgraduate. Her main research interests include reinforcement learning and deep reinforcement learning.

JIA Ri-heng, born in 1989, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include wireless networks, machine learning, and mobile charging.