

对抗背包问题在序列竞争资源分配中的应用

Omkar Thakoor, Rajgopal Kannan, Victor Prasanna

Abstract

本研究解决了在顺序设置中竞争性资源分配的问题，其中两名玩家将资源分配到共享兴趣的对象或位置上。与同时进行的 Colonel Blotto 游戏不同，我们的框架引入了一种顺序决策动态，玩家可以部分或完全了解之前的行动。与依赖复杂混合策略的传统方法不同，我们专注于确定性的纯策略，简化计算的同时保持战略深度。此外，我们将效用结构扩展到容纳分数分配和效用，超越二元的全有或全无范式，允许更精细的结果。

我们将此问题建模为一个对抗性背包游戏，将其表述为一个多级优化问题，整合了领导者的目标与跟随者的最佳反应。这种基于背包的方法在竞争性资源分配的背景下是新颖的，此前的工作仅部分利用它进行跟随者分析。我们的贡献包括：(1) 提出一种用于顺序资源分配问题的对抗性背包公式，(2) 开发适用于分数分配场景的有效启发式算法，以及 (3) 分析 0-1 背包情况，并提供计算难度结果及启发式解决方案。

1 介绍

战略竞争中，两名玩家分配资源到共享兴趣的对象或位置上，是一种常见的情形。Colonel Blotto 游戏是此类情形的一个流行框架。在[7]中引入的原始模型有两个玩家将资源（部队）分配给多个战场，目标是最大化其胜利数量。一个战场上获胜的标准是在该战场上拥有比对手更多的部队，最终收益则是赢得的战场数量。最初为军事策略而开发的该模型已被应用于各种竞争环境，如体育、广告和政治。

在其经典形式中，解由纳什均衡给出，其中两位玩家最佳响应对对方策略。在双人零和游戏中，纳什均衡被证明等同于确保理性玩家最坏情况下最优结果的极大极小策略。然而，在上校布洛陀游戏中

找到这些策略计算复杂度很高，因为它们和解空间可以编码在更简单的两人游戏中，但由于收益结构和战略互动的性质，需要显著更多的计算。

我们的工作三个方面与传统的 Colonel Blotto 游戏区分开来。首先，虽然 Colonel Blotto 框架在一个同时决策的环境中运作，在这个环境中玩家分配资源时不会观察对手的行为，但我们的方法考虑了一个顺序设置，引入了一种基于对之前行动的部分或完全了解来做决定的动态机制。其次，我们模型强调使用确定性的纯策略，而不是依赖涉及概率分配的复杂混合策略，这简化了策略计算，同时保持决策的有效性。第三，我们将收益结构扩展到允许分数分配和分数收益，摆脱传统的全有或全无范式，在这种范式中战场上的胜利完全是一个二元结果，从而能够更细致地评估结果。我们的方法是将其建模为一个背包游戏，特别是对手背包公式的一个双层优化问题。虽然这种视角非常适合这种情况，但以前的作品很少使用或提到它，唯一的例外是[2]仅限于跟随者最优反应的使用，并没有像我们这样扩展到领导者的决策目标。

我们的贡献如下：我们提出了一种用于序列竞争性资源分配问题的对抗背包模型。我们在分数分配设置中进行了分析，并提供了有效的启发式算法。最后，我们也分析了在 0-1 背包模型下的传统全有或全无场景，并给出了计算复杂度结果以及一个启发式解决方案。

2 相关工作

经典的老兵布洛陀问题及其绝大多数变体都考虑的是零和博弈，但其复杂性源于纯策略随部队数量和战场数量的指数增长，这使得最优策略的计算变得具有挑战性。自提出以来，[19; 4; 5; 3; 18; 17; 15; 13; 11; 14; 20] 各种方法已经解决了该问题的具体案例。许多研究放松了整数约束，探索了一种可分割部队的连续版本。例如，[6] 首次提供了三个战场的解决方案，而[12] 则将其扩展到任意数量的战场，并假设双方玩家的部队数量相等。[17] 研究了所有战场权重相同的对称博弈中的最优策略。[13] 处理

了特定案例下的离散、对称版本。最近, [1] 通过设计使用指数大小线性规划的最优策略, 并利用椭圆法在多项式时间内求解, 取得了重大进展, 引起了广泛关注。然而, 尽管具有重要的理论意义, 他们的算法在实际应用中价值不大, 因为其计算复杂度为 $O(B^{12}N^4)$, 其中 B 是玩家的资源预算。

近年来, 背包问题的双层变体得到了广泛的研究。例如, [9] 研究了一种模型, 在该模型中领导者确定背包的重量容量, 而跟随者在给定约束下选择要包含哪些项目。[16] 提出了另一种方法, 他们研究了将物品集合分配给领导和跟随者的双层背包问题, 每个玩家管理各自的子集。[10] 引入的另一种变体涉及两个玩家各自拥有一个背包的情况, 跟随者只能从领导者未装入的项目中进行选择。[8] 展示了上述三种变体均为 Σ_2^P -完全问题。我们的问题中的一个重要区别在于内层问题包含由外层和内层变量乘积组成的双线性项, 这构成了一个挑战。

3 对抗背包问题

回忆一下, 背包问题的一个实例由一组具有给定重量和利润的物品以及一个具有给定重量容量的背包组成。目标是在满足所选项目总重量必须适合放入背包这一约束条件下, 选择具有最大总利润的子集。

我们考虑两个玩家之间的游戏, 一领导者 l 和跟随者 f — 将资源分配给一组 n 项。每个项目 i 具有值 v_i , 我们将所有值的向量表示为 $\mathbf{v}$ 。领导者为每个 i 设置权重 w_i , 这代表了跟随者需要多少资源才能对项目 i 执行完全获胜操作。对于玩家的策略和收益, 我们考虑两种不同的设置。在第一种情况下, 我们考虑一个 0-1 背包问题来捕捉一种全有或全无的情景, 即如果跟随者决定赢得某个物品, 他们必须投入价值为 w_i 的资源。第二种设定是分数背包问题, 在这种情况下, 分配少于设定价格的资源将产生相应比例的分数效用。实际上, 这可以被视为期望效用, 对应着赢得比赛的可能性是所需分配的比例。由于领导者的资源有限, 我们用预算参数 W^l 来表示, 并要求 $\sum_i w_i \leq W^l$ 。跟随者有一个背包容量 W^f 代表其可用的资源。

因此, 我们的对抗背包游戏由元组 $\langle n, \mathbf{v}, W^l, W^f \rangle$ 描述。跟随者的目标是在领导者的权重分配下计算出价值最大的背包, 而领导者希望设置权重 $\mathbf{w}$ 以最小化跟随者的收益。我们的分析显示了当领导者只能设置整数权重时, 即从离散空间中选择, 与允许任何实数值时, 即从连续空间中选择, 问题的复杂性如何变化。

4 分数阶表述

在零和游戏中, 我们不需要区分强与弱的 Stackelberg 均衡, 从而导致领导者面临的以下双层优化

问题 (OP):

$$\begin{aligned} \min_{\sum w_i \leq W^l} \quad & \max_x \sum_i v_i x_i \\ \text{s.t.} \quad & \sum_i w_i x_i \leq W^f \\ & x_i \in [0, 1] \end{aligned}$$

内层是追随者的最佳反应, 即解决背包问题, 外层则是领导者对抗性地设置最优权重。

对于跟随者而言, Dantzig 的算法是一种众所周知的贪心方法, 用于实现最大背包价值。它简单地按照物品的性价比 $\frac{v_i}{w_i}$ (此后记为 BB) 从高到低顺序选择物品, 直到背包装满。

当领导者可以设置非整数权重时, 我们有如下命题给出的解:

命题 1. 领导的最优解是将每个 w_i 设为与其 v_i 成正比, 即 $w_i = \frac{v_i}{\sum_i v_i} W^l \forall i$ 。

证明. 我们注意到, 在连续背包设定中, 跟随者总是能够用尽他们的预算, 所以给定任意权重 $\mathbf{w}$, 他们的收益是 $W^f \cdot b$, 其中 b 是他们选择的物品的 BB (加权) 平均值。但由于他们是按照 BB 从高到低的顺序挑选物品, 因此必须满足 $b \geq \tilde{b}$, 其中 $\tilde{b}$ 是所有物品的 BB 的加权平均值。但由于所有项目的总价值和总重量是固定的, 因此 $\tilde{b} (= \frac{\sum_i v_i}{\sum_i w_i})$ 也是固定的, 与 $\mathbf{w}$ 无关。所以, 如果防御者能够实现 $b = \tilde{b}$, 跟随者的收益就会被最小化, 这是通过将每个项目的 BB 设置为 $\tilde{b}$ 来实现的, 即 $\frac{v_i}{w_i} = \frac{\sum_i v_i}{\sum_i w_i}$, 从而证明了结果。 $\square$

我们将该解称为值成比例 (VP) 权重。由此可知, 即使项目价值和预算都是整数, VP 权重也可能不是整数。因此, 如果 $\mathbf{w}$ 仅限于所有整数, 则上述解决方案不适用。我们将在下一节探讨这种设置。

4.1 离散资源

给定连续情况下直接的闭式解, 对于离散情况下的自然补救方法是获得 VP 权重并将它们四舍五入到最接近的整数。结果表明这并不能保持最优性。

考虑以下示例。假设我们有两个项目, 其值分别为 14, 26, 以及说 $W^l = 4$ 。VP 权重分别为 (1.4, 2.6), 四舍五入后得到 (1, 3)。对于 $W^f = 2$, 它得出 22.67 (跟随者贪婪地选取项目 1 的值 14 和权重 1, 接着选取项目 2 的 26 的 1/3, 使用剩余的权重 1)。这确实可以被认为比其他解决方案更好。然而, 对于 $W^f = 1$, 最优的是 (2, 2), 这将跟随者收益限制为 13, 相比之下, 解 (1, 3) 可以产生 14, 因此, 四舍五入的 VP 权重并非最优。从图 1 中可以得出的关键点是, 最优权重取决于 W^f 。因此, 任何从一个好的分数解找到一个好整数解的方法都不能忽视 W^f 。

对于我们的其余结果, 我们定义了一个分段线性函数 $U_{\mathbf{w}}()$, 用于任何 $\mathbf{w}$ 的情况, 该函数以跟随者分配 (这是 Dantzig 算法计算的内容) 为单位给出跟随者

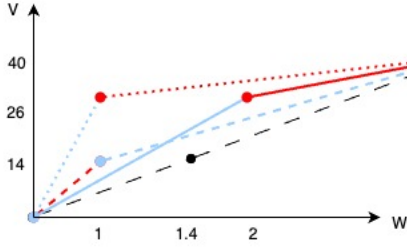


图 1: 示例在离散分数设置中显示追随者收益随追随者预算 W^f 变化, 对于不同的权重 w 。项目 1 ($v_1 = 1.4$) 和项目 2 ($v_2 = 2.6$) 分别用红色、蓝色表示。权重 (1,3), (2,2), (3,1) 通过虚线、实线和点画线分别表示。以黑色线条显示连续分配基线。

的收益。例如, 图.1 显示了四个不同的函数, VP 权重产生了线性函数。当 w 按照物品 BB 的非递增顺序排列, 并且 k 是最大的索引, 使得 $w_1 + w_2 \dots + w_k \leq W$, 我们可以写出

$$U_w(W) = \sum_{i=1}^k v_i + \frac{v_{k+1}}{w_{k+1}} \cdot (W - \sum_{i=1}^k w_i)$$

也值得注意的是, 我们可以构造出最优权重超出相应 VP 权重的地板和天花板值的例子, 这表明该问题更加困难。假设 w^{in} 表示具有整数限制权重的最优解, 而 w^{fr} 是 VP 权重。每个 w_j^{in} 相对于相应的 w_j^{fr} 要么被压缩要么被放大。我们证明了

引理 1. w^{in} 满足其中一个

- $w_j^{\text{in}} \leq w_j^{\text{fr}} \Rightarrow w_j^{\text{in}} = \lfloor w_j^{\text{fr}} \rfloor \quad \forall j$ 或者,
- $w_j^{\text{in}} \geq w_j^{\text{fr}} \Rightarrow w_j^{\text{in}} = \lceil w_j^{\text{fr}} \rceil \quad \forall j$

证明概要. 假设我们有权重 w , 使得 $\exists j, k$ 具有 $w_j < \lfloor w_j^{\text{fr}} \rfloor$ 和 $w_k > \lceil w_k^{\text{fr}} \rceil$ 。然后, 我们考虑通过改变 w 而得到的 w' 。 $w'_j = w_j + 1$ 并使用 $w'_k = w_k - 1$ 的比较值。通过比较 BB 值 $\frac{v_j}{w_j} > \frac{v_j}{w'_j} > \frac{v_k}{w_k} > \frac{v_k}{w'_k}$ 以及 $U_w()$ 和 $U_{w'}()$ 的凹性, 我们能够证明 $U_w() \geq U_{w'}()$ (详细内容省略以简化说明)。因此, 我们能够在改进解决方案的同时改变权重, 直到无法找到如上的配对 (j, k) , 从而最终的解 w^{in} 满足我们的结果中的两个条件之一。 $\square$

接下来, 我们分析权重的整数限制对防御者目标造成的最坏情况差异。

命题 2. 对于最优解 w 和 VP 权重 $\tilde{w}$, 我们有 $U_w(W^f) \leq U_{\tilde{w}}(W^f) \left(1 + \frac{n-1}{W^f}\right)$

证明. 回忆一下, $U_{\tilde{w}}$ 是一个斜率为 $\tilde{b} = \frac{\sum v_i}{W^f}$ 的线性函数。接下来, 凹函数 U_w 在 $W = 0$ 和 $W = W^f$ 处与

$U_{\tilde{w}}$ 具有相同的值。此外, 由于 U_w 是分段线性的, 函数 $(U_w - U_{\tilde{w}})$ 在 U_w 的一个断点处达到最大值, 即对于某个 $k < n$, 有 $W = \sum_{i=1}^k w_i$ 。这简化为

$$(U_w - U_{\tilde{w}}) \left(\sum_{i=1}^k w_i \right) = \tilde{b} \cdot \sum_{i=1}^k (\tilde{w}_i - w_i) \leq (n-1) \tilde{b}$$

这个界限由引理 1 得出。由于这是最大值的一个界限, 我们特别得到了 $(U_w - U_{\tilde{w}})(W^f) \leq (n-1) \tilde{b}$ 。重写 $U_{\tilde{w}}(W^f) = \tilde{b} W^f$ 并重新排列项后给出了结果。 $\square$

启发式解决方案

在分析了最优解的界限和性质后, 我们现在提出了一些有效的启发式方法, 并在第 5 节中进行了数值评估。

我们的启发式算法通过修改 VP 权重 $U_{\tilde{w}}$, 逐步构建函数 U_w 来计算最优的 w 。第一个启发式算法 BBup, 如 Algorithm 1 所示, 工作方式如下。我们迭代地将项目视为候选的第一部分 U_w (第 3 行) —, 对于项目 i , 我们设置 $w_i = \lfloor \tilde{w}_i \rfloor$ 。解决方案 $U_{\tilde{w}}(W^f)$ 在最终设定 w_i 后的质量是通过将其视为一个部分来 (低) 估的, 即: 以剩余项目的平均 BB (因此启发式名称中的 'BB') 为结果, 导致跟随者的预算比例回报 (第 4 行)。具有最好这种估计值的项目 (第 6 行) 被选作第一部分, 然后从考虑中移除 (第 7-8 行)。然后对残差问题重复该过程 (第 9 行)。当分配的权重超过 W^f 时停止该程序, 并且剩余的部分以任意顺序构建 (出于简明省略了详细信息)。

虽然先前的方法构建了 U_w 上升, 我们类似地考虑 BBdown 来构建部分下降, 而 BB+ 则视为考虑两者中更好的一个。我们讨论这样做的优势以及我们的数值结果。很容易看出这些具有 $O(n^2)$ 的运行时间复杂度, 因为每个需要 $O(n)$ 时间的 $O(n)$ 递归调用。

Algorithm 1: BBup: 一个用于离散分数设置的有效启发式算法

```

1 BBup ( $n, v, W^1, W^f$ ):
2    $\tilde{w} = \text{VP weights}$ 
3   for  $i = 1, 2, \dots, n$  do
4      $est[i] = v_i + \frac{W^f - \lfloor \tilde{w}_i \rfloor}{W^1 - \lfloor \tilde{w}_i \rfloor} \sum_{j \neq i} v_j$ 
5   end
6    $minI = \text{argmin}(est)$ 
7    $W^f -= \lfloor \tilde{w}_{minI} \rfloor, W^1 -= \lfloor \tilde{w}_{minI} \rfloor$ 
8    $v.remove(minI), n -= 1$ 
9    $bbEstRound(n, v, W^1, W^f)$ 
```

我们的下一个方法 GD-f2c 如算法 2 所示, 利用了引理 1。其中的第一个条件意味着没有权重 w_i 小于 $\lfloor \tilde{w}_i \rfloor$ 。因此, 我们将所有 w_i 初始化为所述的基值

(第 2 行), 并表示 s 为待分配的剩余权重 (第 3 行)。然后通过贪婪地考虑使用 Dantzig 算法计算出全局目标值来找到吸收一个单位权重的最佳项目 (第 5-6 行)。一旦找到这样的项目并增加其权重 (第 10-11 行), 我们就重复该过程直到所有剩余权重分配完毕。

Algorithm 2: GD-f2c: 一种高效的启发式算法用于离散分数设置

```

1  $\tilde{w} = \text{VP 权重}; w = (\lfloor \tilde{w}_1 \rfloor, \lfloor \tilde{w}_2 \rfloor, \dots, \lfloor \tilde{w}_n \rfloor);$ 
   $s = W^l - \sum_i w_i; \text{ for } j = 1, 2, \dots, s \text{ do}$ 
2    $\text{for } i = 1, 2, \dots, n \text{ do}$ 
3      $w' = (w_1, w_2, \dots, w_i + 1, \dots, w_n)$ 
4     Sort  $w'$  in non-ascending BB
5      $est[i] = U_{w'}(W^f)$ 
6   end
7    $minI = \text{argmin}(est)$ 
8    $w_{minI} += 1$ 
9 end
```

虽然先前的方法 gd 分配盈余时采用了来自 floor 值到 ceiling 的权重 (因此得名 GD-f2c), 我们类似地考虑了 GD-c2f, 该方法初始化权重为天花板值并逐渐向流量值通缩, 反映了引理 1 的第二个条件。由于这两个中只有一个被保证, 我们让 GD+ 成为考虑两者中较好的一个的算法。通过按 BB 排序顺序维护项目, 我们可以计算 Dantzig 的算法在 $O(n)$ 中, 并且由于内部和外部循环分别运行 $O(n)$ 和 $O(s)$ 次, 这些的运行时间复杂度是 $O(n^2s)$ 。请注意, 虽然 s 依赖于项目值, 但它最多为 $O(n)$ 。

我们现在考虑 0-1 设置, 即全有或全无的情景。

5 0-1 形式化

类似于分数设置, 我们有以下领导者的双层优化问题 (0-1 模型):

$$\begin{aligned}
& \min_{\sum w_i \leq W^l} \max_x \sum_i v_i x_i \\
& \text{s.t. } \sum_i w_i x_i \leq W^f \\
& x_i \in \{0, 1\}
\end{aligned}$$

在分数模型中, 最后一个约束条件是 $x_i \in [0, 1]$ 。回想一下, 在分数背包公式中, 命题 1 表明价值比例权重是最优的。我们可以证明这在 0/1 公式中不一定最优, 如下所示。

反例

假设我们有两个物品, 其价值分别为 50 和 100。假设 $W^l = 15$ 和 $W^f = 12$ 。将权重设置为 5 和 10 分别导致领导者效用 100, 但最优值是通过将权重设为 2 和 13 获得的 50。

另一个最优解不完全将一个成本设置得比 W^f 更高的情况: 假设我们有三个物品, 其效用为 10, 另外三个物品的效用为 3。假设 $W^l = 390$ 和 $W^f = 235$ 。将两种商品类型的权重分别设置为 100 和 30, 得到的效用是 23 (如 $100 + 100 + 30 \leq 235$ 所示)。但是最优值是通过将两种商品类型的成本分别设置为 120 和 10 而达到的 19。

目标的下界

请注意, 无论权重如何设置, 最小的 k 个权重之和总是可以装入背包中的, 其中 $k = \lfloor \frac{W^f}{W^l} n \rfloor$, 导致目标值为最小的 k 个值的总和。这个界对于几种情况是紧的; 特别是, 我们可以概述以下命题:

命题 3. 上述下界在

- $\frac{W^f}{W^l} < \frac{1}{n-1}$ 通过将所有 $i \geq 2$ 设置为 $w_i > W^f$ 获得。
- $\geq 1 - \frac{1}{n-1}$ 通过设置 $w_n > W^f$ 获得。

时是紧的。由此可知, 对于 $n = 2$ 它总是紧的。

表 1 显示了预算比率某些特殊情况下的闭式解。 $n \leq 4$ 的值是通过逐案分析获得的, 而对于小预算值的情况, 则使用归纳论证将其扩展到任意的 n 。从第一列可以看出 VP 权重的次优性没有界限: 如果 $\frac{v_1}{V} \leq \frac{W^f}{W^l} < \frac{1}{n-1}$, 最优值为 0, 而 VP 权重则会产生非零结果, 因为跟随者可以至少将 v_1 放入背包中。

除了上述下界外, 我们还可以将上界设为最大的 k 权重之和, 因为将所有项目分配相等的权重允许最多放入 k 个项目 (跟随者选择最大的 k), 因此最优解不会更差。

命题 4. 存在一个最优解满足 $\forall i, j$:

$$v_i < v_j \Rightarrow w_i < w_j \quad \& \quad v_i = v_j \Rightarrow w_i = w_j.$$

这一结果特别有利于通过增加上述约束来加速任何 (双层) LP 求解器。

离散分配的计算复杂性

我们现在讨论该问题的难度及其在某个复杂性类中的包含关系。

我们回忆一下, 复杂性类 Σ_2^P 包含所有可以写成形式 $\exists x \forall y P(x, y)$ 的判定问题; 也就是说, 以存在量词开始, 接着是全称量词, 最后是一个可以在多项式时间内评估的布尔谓词 $P(x, y)$; 例如, 请参见 Papadimitriou 的书 [23] 第 17 章。与 bilevel 问题 DeRi [5], MACH, DNeg [6] 的判定版本类似, 我们的问题是询问是否存在一种方法来确定领导者控制的变量, 使得跟随者控制的所有可能变量设置对领导者产生一个良好的目标值。由于这个问题恰好具有形式 $\exists x \forall y P(x, y)$, 我们得出结论它包含在 Σ_2^P 中。

定理 1. 问题属于 NP 难问题。

证明. 我们将问题归约为已知的 NP 难问题——数划分问题 (NPP)。假设我们得到一个 NPP 实例

n=2	$\begin{array}{c} [0, 1/2) \\ 0 \end{array}$				$\begin{array}{c} [1/2, 1) \\ v_1 \end{array}$	
n=3	$\begin{array}{c} [0, 1/3) \\ 0 \end{array}$	$\begin{array}{c} [1/3, 1/2) \\ v_1 \end{array}$			$\begin{array}{c} [1/2, 2/3) \\ \min\{v_3, v_1 + v_2\} \end{array}$	$\begin{array}{c} [2/3, 1) \\ v_1 + v_2 \end{array}$
n=4	$\begin{array}{c} [0, 1/4) \\ 0 \end{array}$	$\begin{array}{c} [1/4, 1/3) \\ v_1 \end{array}$	$\begin{array}{c} [1/3, 2/5) \\ \min\{v_3, v_1 + v_2\} \end{array}$	$\begin{array}{c} [2/5, 1/2) \\ \min\{v_4, v_1 + v_2\} \end{array}$	$\dots$	$\begin{array}{c} [3/4, 1) \\ v_1 + v_2 + v_3 \end{array}$
n	$\begin{array}{c} [0, 1/n) \\ 0 \end{array}$	$\begin{array}{c} [1/n, 1/(n-1)) \\ v_1 \end{array}$	$\begin{array}{c} [1/(n-1), 2/(2n-3)) \\ \min\{v_3, v_1 + v_2\} \end{array}$	$\begin{array}{c} [2/(2n-3), 1/(n-2)) \\ \min\{v_4, v_1 + v_2\} \end{array}$	$\dots$	$\begin{array}{c} [1-1/n, 1) \\ v_1 + \dots + v_{n-1} \end{array}$

表 1: 表展示了针对不同值的 n (指定在第 1 列) 的各种区间范围内的 $\frac{W^f}{W^l}$ (每个单元格中的顶行) 的最佳目标的闭式表达式。

($v_1, v_2, \dots, v_n$)。我们构造一个包含 n 个物品、这些物品的价值为 $W^l = \sum_i v_i$ 和 $W^f = W^l/2$ 的背包游戏实例。然后, 我们声称最优划分具有 k 的差异当且仅当背包游戏具有最优目标值 $W^f - k/2$ 。

为了证明这一点, 假设最优划分是 (S, S') , 其权重为 $v(S) = v(S') - k$ 。请注意, 无论领导者设置的权重如何, S 和 S' 不可能同时不可承担, 因为它们的权重之和必须等于 W^l , 而跟随者的预算是 $W^f = W^l/2$ 。因此, 最优的权重分配不可能达到优于 $W^f - k/2$ 的效果。

另一方面, 价值成比例的权重分配实现了 $W^f - k/2$, 因为 S 必须是最大的可负担子集 (更大的可负担子集将为 NPP 产生更小的差异)。因此, 最优权重分配不会差于 $W^f - k/2$ 。

因此, 在这种情况下最优值正是 $W^f - k/2$ 。 $\square$

最小化局部最大值

我们利用命题 3 的分析推断出存在一个大小为 k 的背包问题解, 对于跟随者来说是几乎最优的。因此, 我们考虑跟随者的局部最优 k 大小解并通过以下 MILP 进行最小化:

$$\min_{\sum w_i \leq W^l, V} V \quad (1)$$

$$\text{s.t. } \sum_i w_i x_i \leq W^f \quad (2)$$

$$\sum_i v_i x_i \leq V \quad (3)$$

$$y_{jkj} = 0, y_{jkk} = 1 \quad \forall j \neq k \quad (4)$$

$$y_{jki} = x_i \quad \forall i \neq j, k \quad \forall j \neq k \quad (5)$$

$$\sum_i w_i y_{jki} > W^f - M_1 a_{jk} \quad \forall j \neq k \quad (6)$$

$$\sum_i v_i y_{jki} \leq V + M_2 b_{jk} \quad \forall j \neq k \quad (7)$$

$$a_{jk} + b_{jk} \leq 1 \quad \forall j \neq k \quad (8)$$

$$\sum_i x_i = k \quad (9)$$

$$x_i, y_{jki}, a_{jk}, b_{jk} \in \{0, 1\} \quad \forall i, j, k \quad (10)$$

这里, x 表示我们正在寻找的局部最优解, 由约束条件 2,3 特征化。每个 y_{jk} 是 x 的一个邻居, 在其中项目 j 可能与项目 k 交换位置, 如约束条件 4,5 所述。约束条件 6 和 7 确保如果 y_{jk} 符合容量要求, 那么 $a_{jk} = 1$ 也成立; 如果 y_{jk} 为跟随者提供更高的价值, 则 $b_{jk} = 1$ 也成立。为了使 x 局部最优, 约束条件 8 确保没有既可负担又能提供更多价值的邻居。

第 6 个严格不等式可以通过引入一个小常数 ϵ 来处理。第 2、6 中的双线性项是二进制变量和连续变量的乘积, 已知这些是可以线性化的。

6 数值结果

我们现在展示数值结果来评估第 4 节中的启发式方法。我们通过采样生成实例, 物品数量在范围 $[10, 20]$ 内, 项目价值作为总价值 V 的分区。我们将领导者预算设置为 $W^l = 100$, 并将分数 VP 权重设置为 $V = 100W^l$ 以获得小数权重。我们通过改变 $\{0.1, 0.2, \dots, 0.9\}$ 中的比率 $\frac{W^f}{W^l}$ 来变化追随者预算, 并如上所述为每种情况选择 100 个实例。我们将我们的启发式方法与基线启发式方法全率进行比较, 该方法 roundsVP 权重 randomly。

我们的结果总结在图 2 中。每个图显示了由于我们的启发式算法相对于 RR 的改进 (用绿色表示)、与 RR 相同输出 (用黄色表示) 以及比 RR 更差的输出 (用红色表示) 的实例数量。我们分析了 x 轴上不同值的比率 $\frac{W^f}{W^l}$ 的性能。

图 2a, 2b, 2c 展示了 BBup、BBdown 和 BB+ 的性能, 分别与 RR 对比。我们看到 BBup 和 BBdown 在预算比例不接近 0.5 时效果显著, 但在接近 0.5 时则表现不佳。这可能是因为我们在方法中使用的分段线性函数的线性估计自然在预算比例更接近 0 或 1 时更加准确。BB+ 被证明是解决这个问题的有效补救措施, 因为 BBup 和 BBdown 显示出了互补的效果。在其最差的情况下 (预算比率为 0.5), BB+ 在 24% 的实例中表现不如 RR, 但在 46% 的实例中提高了输出。在最佳情况下 (预算比例为 0.1 或 0.9), 这些数字分别改善到了 1% 和 90%。

尽管 GD-f2c、GD-c2f 和 GD+ 具有稍高的运行

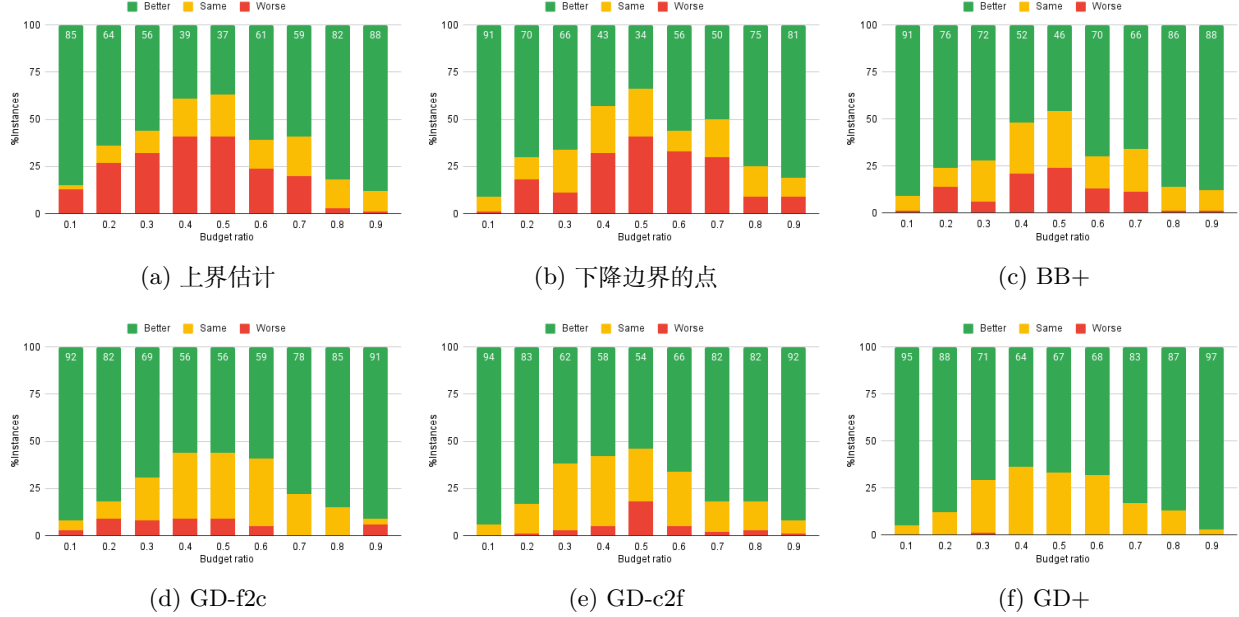


图 2: 与 RR 基线相比的启发式算法比较。每个图显示了在预算比例变化时, 我们的启发式算法表现优于 RR (绿色)、相同 (黄色) 和较差 (红色) 的实例百分比。

时复杂度, 它们的整体输出表现明显更好。前两者分别优于 BB+, 而 GD+ 则在集成学习的优势下表现得更加出色。在所有 1000 个实例中, GD+ 仅在一个实例上劣于 RR, 在最坏的情况下 (预算比率为 0.5 左右) 约有 65% 的实例表现出严格改进, 在最好情况下 (预算比率为 0.1 或 0.9) 则达到 97%。

7 结论

这项工作探讨了顺序设置中的竞争性资源分配, 与传统的同时进行的上校布洛特框架有所不同。我们关注一个决策制定场景, 限制使用确定性的纯策略, 这提供了计算上的简单性, 同时保持战略有效性。此外, 我们将收益结构扩展到包含分数分配和收益, 使结果更加细腻和成比例, 相比传统二元框架更为复杂。我们通过一种新颖的对抗背包问题表述来实现这一点, 将问题构架为一个双层优化问题, 整合了领导者和跟随者的双重目标。对于分数分配设置, 我们展示了有效的启发式方法, 其中 BB+ 具有 $O(n^2)$ 的时间复杂度优势, 但在最坏情况下只实现了相对较小的改进。启发式 GD+ 即使在最坏情况下也显示出显著改进, 其时间复杂度略高为 $O(n^3)$ 。我们还提供了对经典 0-1 背包问题的分析, 显示它是一个 NP 难问题, 填补了文献中的一个重要持久空白, 同时也提供了一个 MILP 启发式方法来对抗局部最大值。

参考文献

- [1] AmirMahdi Ahmadinejad, Sina Dehghani, MohammadTaghi Hajiaghayi, Brendan Lucier, Hamid Mahini, and Saeed Seddighin. From duels to battlefields: Computing equilibria of blotto and other games. *Mathematics of Operations Research*, 44(4):1304–1325, 2019.
- [2] Soheil Behnezhad, Avrim Blum, Mahsa De-rakhshan, MohammadTaghi HajiAghayi, Mohammad Mahdian, Christos H Papadimitriou, Ronald L Rivest, Saeed Seddighin, and Philip B Stark. From battlefields to elections: Winning strategies of blotto and auditing games. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2291–2310. SIAM, 2018.
- [3] Richard Bellman. On “colonel blotto” and analogous games. *Siam Review*, 11(1):66–68, 1969.
- [4] Donald W Blackett. Some blotto games. *Naval Research Logistics Quarterly*, 1(1):55–60, 1954.
- [5] Donald W Blackett. Pure strategy solutions of blotto games. *Naval Research Logistics Quarterly*, 5(2):107–109, 1958.
- [6] E Borel and J Ville. *Application de la théorie des probabilités aux jeux de hasard*, original edition by gauthier-villars, paris, 1938; reprinted at the end of *théorie mathématique*

du bridg a la port e de tous, by e. Borel & A. Ch eron, Editions Jacques Gabay, Paris, 1991.

- [7] Emile Borel. The theory of play and integral equations with skew symmetric kernels. *Econometrica: journal of the Econometric Society*, pages 97–100, 1953.
- [8] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J Woeginger. A study on the computational complexity of the bilevel knapsack problem. *SIAM Journal on Optimization*, 24(2):823–838, 2014.
- [9] Stephan Dempe and Klaus Richter. Bilevel programming with knapsack constraints. Citeseer, 2000.
- [10] Scott DeNegre. Interdiction and discrete bilevel linear programming. Lehigh University, 2011.
- [11] Russell Golman and Scott E Page. General blotto: games of allocative strategic mismatch. *Public Choice*, 138:279–299, 2009.
- [12] Oliver Alfred Gross and R. A. Wagner. A Continuous Colonel Blotto Game. RAND Corporation, Santa Monica, CA, 1950.
- [13] Sergiu Hart. Discrete colonel blotto and general lotto games. *International Journal of Game Theory*, 36(3-4):441–460, 2008.
- [14] Dan Kovenock and Brian Roberson. Coalitional colonel blotto games with application to the economics of alliances. *Journal of Public Economic Theory*, 14(4):653–676, 2012.
- [15] Dmitriy Kvasov. Contests with limited resources. *Journal of Economic Theory*, 136(1):738–748, 2007.
- [16] Raid Mansi, Claudio Alves, JM Val rio de Carvalho, and Sa d Hanafi. An exact algorithm for bilevel 0-1 knapsack problems. *Mathematical Problems in Engineering*, 2012(1):504713, 2012.
- [17] Brian Roberson. The colonel blotto game. *Economic Theory*, 29(1):1–24, 2006.
- [18] Martin Shubik and Robert James Weber. Systems defense games: Colonel blotto, command and control. *Naval Research Logistics Quarterly*, 28(2):281–287, 1981.
- [19] John W Tukey. A problem of strategy. *Econometrica*, 17(1):73, 1949.
- [20] Jonathan Weinstein. Two notes on the blotto game. *The BE Journal of Theoretical Economics*, 12(1):0000101515193517041893, 2012.