出价策略大体上可以分为两部分：
1）结合业务特点，将竞价优化问题建模为线性规划问题，并利用对偶优化理论，求解出最优出价的公式。这一步基本没什么难度，只要掌握一些线性规划知识，都可以求解出来
2）求解最优出价公式中的参数，以达到流量的最优分配。这一步是出价策略中最难的一部分

参数求解目前来看主要有以下三种形式：
1）利用历史请求信息求解线性规划问题，得到最优参数。这种方式适合流量比较稳定的场景，如一些站内流量。对于流量变化比较大的场景，利用历史流量求解出来的参数在未来往往无法达到很好的效果
2）利用PID控制策略进行调节。这种应该是业界应用最为广泛的方式了，逻辑及实现都较为简单，能够有效地实现调控目的。缺点是，如果需要调节的变量不止一个，单纯的PID控制策略无法达到最优，因为它无法处理多个变量之间的耦合关系
3）利用强化学习模型来求解参数。这也是目前许多paper中推崇的方式，优点是可以克服PID的缺陷，同时对多个参数进行调节，缺点是其训练起来相对困难，要想达到期望的效果，需要一定的探索成本

Optimized Cost per Click in Taobao Display Advertising(2017)

论文链接：[Optimized Cost per Click in Taobao Display Advertising]

论文主要贡献：
1）提出了ocpc投放模式下的出价策略框架，能够同时满足广告主的质量和数量需求
2）提出了在ecpm非$ctr*bid$情况下的排序策略框架

符号定义：
b_a^*：最优出价
b_a：广告主支付的点击费用
$p(c|u,a)$：用户u在点击广告a后的交易转化概率
v_a：预估的单次交易额
n_u：一个用户在一段时间内对一个广告的总点击数

广告a单次点击的期望ROI：

$$roi_{u,a} = \frac{p(c|u,a) \cdot v_a}{b_a} \quad (1)$$

广告a在不同用户和点击下的总体ROI：

$$roi_a = \frac{v_a \cdot \sum_u n_u \cdot p(c|u,a)}{b_a \cdot \sum_u n_u} = \frac{E_u[p(c|u,a)] \cdot v_a}{b_a} \quad (2)$$

由公式(2)可知，广告主的整体ROI由三个因素决定：期望转化率$E_u[p(c|u,a)]$，预估的单次交易额v_a以及出价b_a。其中，v_a是每个广告的内在属性，$E_u[p(c|u,a)]$是每次特定拍卖的静态值

等式(2)证明了roi_a与$E_u[p(c|u,a)]$之间的线性关系，为了避免ROI下降，竞价优化需要满足$\frac{b_a^*}{b_a} \leq \frac{p(c|u,a)}{E_u[p(c|u,a)]}$。考虑到广告主获取高质量流量的需求，可以在投放时遵循以下原则：当流量质量较高，即$\frac{p(c|u,a)}{E_u[p(c|u,a)]} \geq 1$时，提高出价；当流量质量偏低，即$\frac{p(c|u,a)}{E_u[p(c|u,a)]} < 1$时，降低出价。质量和数量折中的优化范围如图(1)所示，固定阈值r_a如(40%)通常是为了业务安全而设置的，用于避免广告主在优化ROI时，流量下降过大

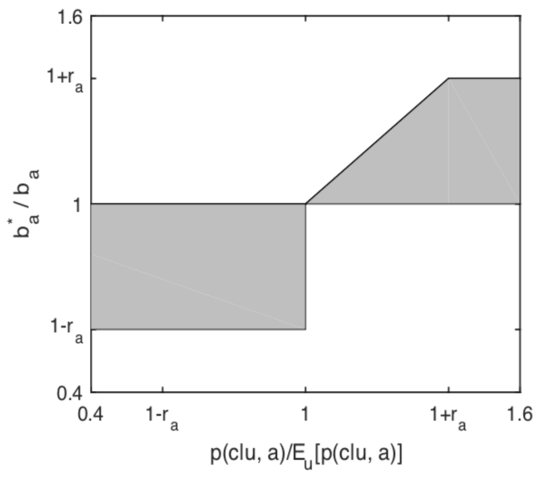


Figure 3: The bid optimization scope (the gray area) under ROI constraint.

对于图(3)中的的灰色区域，我们用$l(b_a^*)$和$u(b_a^*)$来表示上届和下界，它的计算方法如公式(3)和公式(4)所示。竞价优化目标可以推广到广告主的其它目标，而不仅限于ROI。如果部分广告主未授权进行竞价优化，则其上下限均等于b_a

$$l(b_a^*) = \begin{cases} b_a \cdot (1-r_a), & \frac{p(c|u,a)}{E_u[p(c|u,a)]} < 1 \\ b_a, & \frac{p(c|u,a)}{E_u[p(c|u,a)]} \geq 1 \end{cases} \quad (3)$$

$$u(b_a^*) = \begin{cases} b_a, & \frac{p(c|u,a)}{E_u[p(c|u,a)]} < 1 \\ b_a \cdot min(1+r_a, \frac{p(c|u,a)}{E_u[p(c|u,a)]}), & \frac{p(c|u,a)}{E_u[p(c|u,a)]} \geq 1 \end{cases} \quad (4)$$

假设我们在eCPM排序机制下展示一个广告，希望最大化以下目标：

$$\max_{b_1^*,\dots,b_n^*} f(b_k^*) \quad (5)$$
$$s.t. \quad k = arg\max_i \quad pctr_i * b_i^* \quad (6)$$
$$l(b_i^*) \leq b_i^* \leq u(b_i^*), i=1,\dots,n \quad (7)$$

其中，n是一次PV中候选广告的数量，f是一个能够表示综合指标的函数，它包含了各方面的指标需求。我们假设$f(b_i^*)$与b_i^*是单调关系，等式(6)中的条件意味着在拍卖中获得曝光机会的广告需要在eCPM排序中排在top k的位置，等式(7)意味着优化后的出价需要在指定范围内。对于等式(5)中提出的优化问题，有两层含义：一方面，我们试图找到第k个广告使得$f(b_k^*)$的值最大；另一方面，每个广告的调整后出价应该最大化第k个广告的eCPM。对于f，我们给出以下两个示例：

$$f_1(b_k^*) = pctr_k * pcvr_k * v_k$$
$$f_2(b_k^*) = pctr_k * pcvr_k * v_k + \alpha * pctr_k * b_k^*$$

其中，f_1试图最大化淘宝平台的整体GMV，即所有广告主的收入。f_2是平台GMV和广告收入的综合，α是二者的平衡系数，不同的α会导致不同的优化目标

排序任务的剩余工作是通过为每个广告a找到最佳b_a^*来最大化等式(5)中的目标。类推到出价边界，有：

$$l(s_a^*) = pctr_a * l(b_a^*)$$
$$u(s_a^*) = pctr_a * u(b_a^*)$$

用于表示排序得分的上下界。为了优化目标，只需要根据$f(u(b_i^*))$对广告进行排序(需要注意的是，使用的是出价上届$u(b_i^*)$，因为我们假设$f(b_i^*)$与b_i^*是单调关系)，然后从上到下选择一个广告，这个广告的$u(s_k^*)$不小于其它任何广告的$l(s_i^*)$，最终的出价$b_k^* = u(b_k^*)$。最后，更新其它候选广告在其可行域内的竞价，以确保该广告拥有最大的eCPM

在实际场景中，每个PV可能有多个广告展示位置，我们在算法1中提出了贪心算法，并做如下简单说明。首先，按照一个广告位算法选出广告k放到广告位1，然后调整剩余广告eCPM分数的上届，以保证广告k的eCPM是最大的。接着，重复这个步骤，直到挑选出k个广告。最后，将最终出价设置为$u(b_i^*)$

Algorithm 1: Ranking Algorithm

Input: Ad list A, corresponding boundaries of bid price
Output: Optimized bid prices b_a^* for $\forall a \in A$

```
Winning set 𝒜 = ∅;
repeat
    Sort ads in A in descending order of f(u(b_i^*));
    t ← the largest l(s_a^*) for ∀a ∈ A;
    Find the first ad k from A that u(s_k^*) ≥ t;
    𝒜 = 𝒜 ∪ {k};
    A = A\{k};
    for i ∈ A do
        u(s_i^*) = min(u(s_i^*), u(s_k^*));
        u(b_i^*) = min(u(b_i^*), u(s_i^*)/pctr_i);
    end
until ||𝒜|| == N or A == ∅;
for i ∈ 𝒜 ∪ A do
    b_i^* = u(s_i^*)/pctr_i;
end
Return b_a^* for each ad in 𝒜 ∪ A;
```

Bid Optimization by Multivariable Control in Display Advertising(2019)

论文链接：[Bid Optimization by Multivariable Control in Display Advertising]

论文主要贡献：
1）提出了利用最优化理论推导出价公式的方法
2）提出了利用双PID控制方法求解参数的方法

paper给出的示例是在预算约束和点击成本约束下，最大化广告主收益。但其实整体推导过程也可适用于其它的广告场景，具体问题定义如下：

$$\begin{aligned} &\max_{x_i} \sum_{i=1,\dots,N} x_i * CTR_i * CVR_i \quad (LP1) \\ &s.t. \sum_{i=1,\dots,N} x_i * wp_i \le B \quad (1) \\ &\frac{\sum_{i=1,\dots,N} x_i * wp_i}{\sum_{i=1,\dots,N} x_i * CTR_i} \le C \quad (2) \\ &where\ 0 \le x_i \le 1, \forall i \end{aligned}$$

上述公式中各符号的定义如下：
N：广告计划可参与的总拍卖次数
x_i：第i次拍卖获胜的概率
wp_i：第i次拍卖的赢价，即bid_price要大于这个值才能赢得拍卖机会
B：广告计划的总预算
C：广告计划设置的点击成本

由于我们研究的并不是广告分配问题，而是出价策略。因此，我们不需要直接求出上述优化问题的最优解，只需要求出取值为最优时的解形式，作为我们的出价公式即可

首先根据对偶理论将原问题转化为对偶问题，关于对偶理论，可参考前面的相关文章。对偶问题如下：

$$\begin{aligned} &\min_{p,q,r_i} B * q + \sum_{i=1,\dots,N} r_i \quad (LP2) \\ &s.t. \quad wp_i * p + (wp_i - CTR_i * C)q + r_i \ge v_i, \forall i \quad (3) \\ &where\ p \ge 0 \\ &\qquad q \ge 0 \\ &\qquad r_i \ge 0, \forall i \\ &\qquad v_i = CTR_i * CVR_i, \forall i \end{aligned}$$

上式中的p, q, r_i都是对偶变量，对应于原问题中的三类约束：预算、成本及对变量x_i的约束。根据互补松弛定理，有下面两个公式：

$$\begin{aligned} x_i^* * (v_i - wp_i * p - (wp_i - CTR_i * C)q - r_i) = 0, \forall i \quad (4) \\ (x_i^* - 1) * r_i^* = 0, \forall i \quad (5) \end{aligned}$$

其中，x_i^*和r_i^*分别是原问题和对偶问题的最优解。下面我们对出价公式进行推导，首先，写出拉格朗日函数：

$$L(x,p,q) = \sum_{i=1,\dots,N} x_i CTR_i CVR_i + p(\sum_{i=1,\dots,N} x_i wp_i - B) + q(\sum_{i=1,\dots,N} x_i (wp_i - CTR_i C)) \quad (6)$$

这里我们忽略了自变量x_i的约束条件，后面会对其进行讨论。由于我们不需要确切的解，只需要最优解的表达式，因此可令$\frac{\partial L(x,p,q)}{\partial x} = 0$，求解可得：

$$bid_i^* = wp_i = \frac{1}{p+q} \times CTR_i \times CVR_i + \frac{q}{p+q} \times C \times CTR_i \quad (7)$$

由式(7)可得：

$$v_i = (p+q) bid_i - q \times C \times CTR_i \quad (8)$$

将式(8)代入式(4)可得：

$$x_i^* \cdot ((bid_i^* - wp_i)(p^* + q^*) - r_i^*) = 0 \quad (9)$$

根据等式(9):

1）如果广告活动赢得了展示机会，意味着$x_i^* > 0$，同时，有$r_i^* \geq 0$。因此，$bid_i^* \geq wp_i$

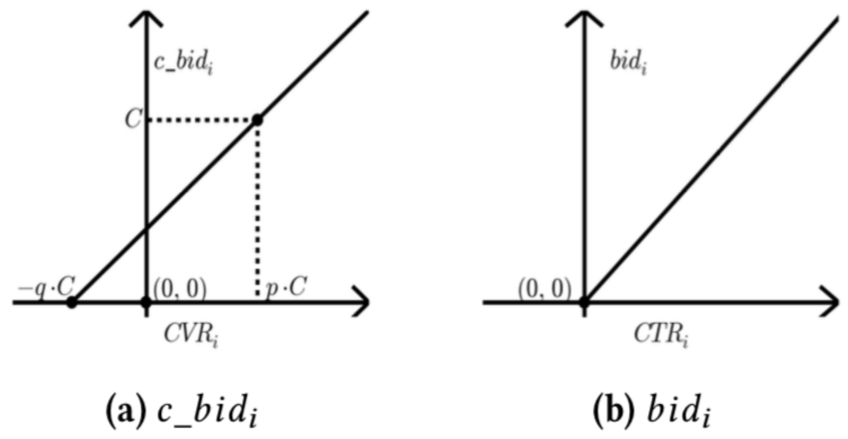
2）如果广告活动没有赢得展示机会，意味着$x_i = 0$，则由公式(5)可得，$r_i^* = 0$。最后，根据不等式(3)可得$bid_i^* \leq wp_i$

从以上两个讨论中可知，无论最优解x^*是赢得这次竞价，还是输掉这次竞价，按照公式(7)进行出价时，总能保证解是最优的

回到公式(7)的最优出价公式，我们将其写成$c_bid * ctr$的形式，如下：

$$c\_bid_i = \frac{1}{p+q} \cdot CVR_i + \frac{q}{p+q} \cdot C \quad (10)$$
$$bid_i = c\_bid_i \cdot CTR_i \quad (11)$$

可以更直观地画出如下所示的图，从图中可知，在CVR为0的情况下，bid也不一定为0，这跟常见的$ecpm = bid \times pctr \times pcvr$不太一样，这可以理解为一些cvr低但是ctr高的流量也是可以拿的

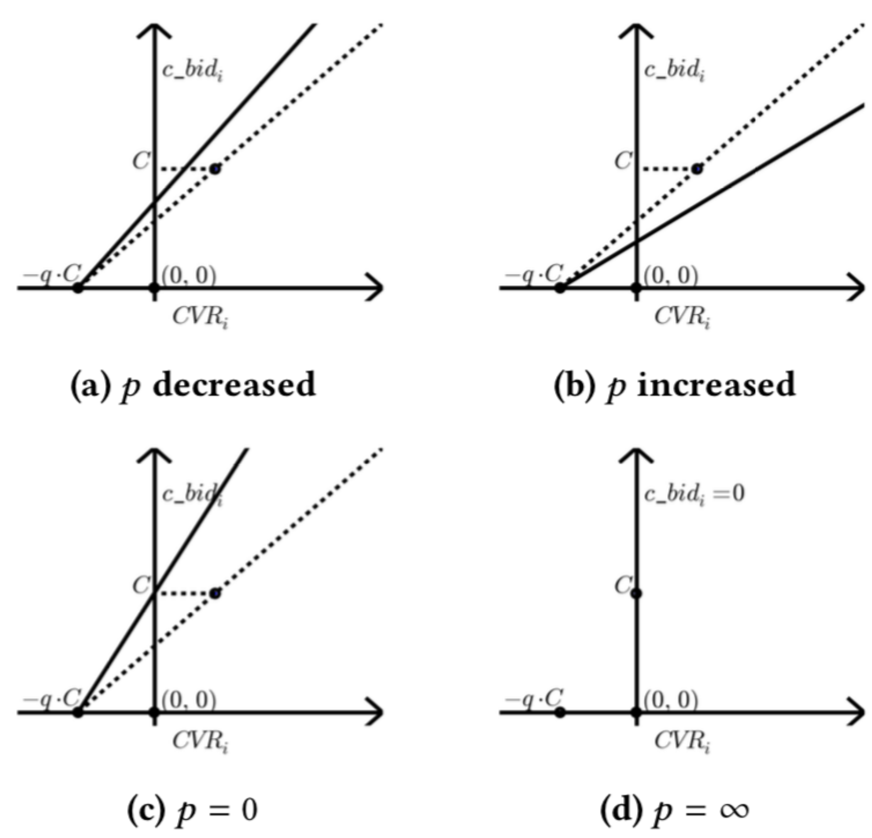


(a) c_bid_i (b) bid_i

Figure 2: Optimal bidding strategy

前面提到，最优出价公式中的p和q的最优解需要拿到参竞后的后验数据，但是bid是要在参竞的时候给出来，这就是一个先有鸡还是先有蛋的问题。针对这个问题，一个最直观的想法是，可不可以利用历史数据来求出最优的p和q，并应用到出价中？答案是no。因为这个方法假设了参竞流量的分布是基本不变的，但是竞价环境是一个受多个因素影响的动态变化环境(包括参竞流量、ctr、cvr等)，即历史的最优不会是未来的最优

由于竞价环境是实时变化的，因此需要动态调价，在调价策略中，需要先明确两个点：**调控的目标和调控的变量**。下面首先分析控制变量p和q分别影响哪些控制目标



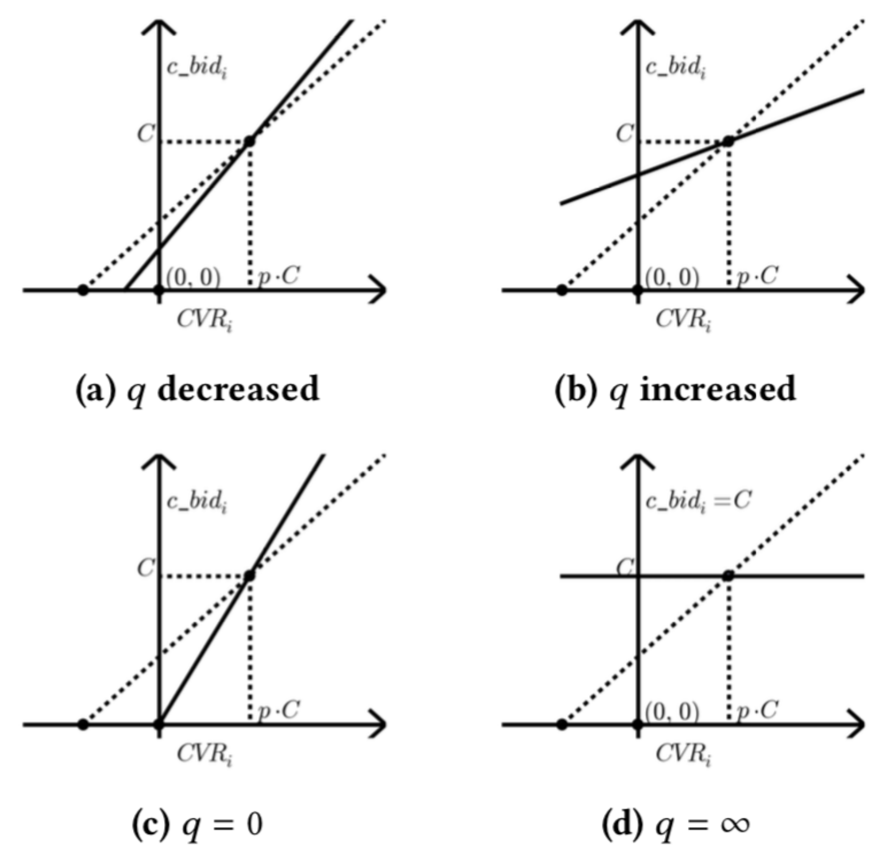
(a) p decreased (b) p increased

(c) $p = 0$ (d) $p = \infty$

Figure 4: Bidding strategies with q fixed and p respectively decreased, increased, equal to 0, and equal to ∞. The dashed line is illustrated as a reference to the original function.

如上图所示，固定q，改变p时，c_bid的变化。从图中可知：

- 出价的直线始终通过(-qC, 0)这个点
- 随着p的减小，出价直线的斜率逐渐增大，表示出价更高，同时消耗的budget也会更多；而随着p增大导致的结果则是刚好相反
- 当p取最小值即0时，表示没有budget的限制，出价公式退化为$C \times CTR + \frac{1}{q} CTR \times CVR$，公式第一项可以认为是只按照点击出价来保点击成本$C$，第二项则是为了达到$max\ v = CTR \times CVR$的目标



(a) q decreased (b) q increased

(c) $q = 0$ (d) $q = \infty$

Figure 5: Bidding strategies with p fixed and q respectively decreased, increased, equal to 0 and equal to ∞. The dashed line is illustrated as a reference to the original function.

上图是固定p改变q时，c_bid的变化。从图中可知：

- **出价的直线始终通过(pC, C)这个点**
- **随着q的减少，出价直线的斜率逐渐增大，表示对于CVR比pC更高的流量出价更高，CVR比pC更低的流量出价更低，而随着q增大导致的结果则刚好相反**
- **当q取最小值，即0时，表示没有点击成本的限制，此时的出价公式退化为$\frac{1}{p}CTR \times CVR$，代表出价成本的符号$C$没有出现在出价公式中，表示总体要达到$max\ v = CTR \times CVR$的目标，同时通过$p$来控制预算**

通过上面的分析可知，参数p可以用来控制预算的使用，参数q可以用来控制点击成本，这与我们推导最优出价公式时对应的约束条件是一致的。因此，一种最简单的策略是两个独立的PID来分别调控变量p和q，调控的目标则是预算和成本，如下图所示：

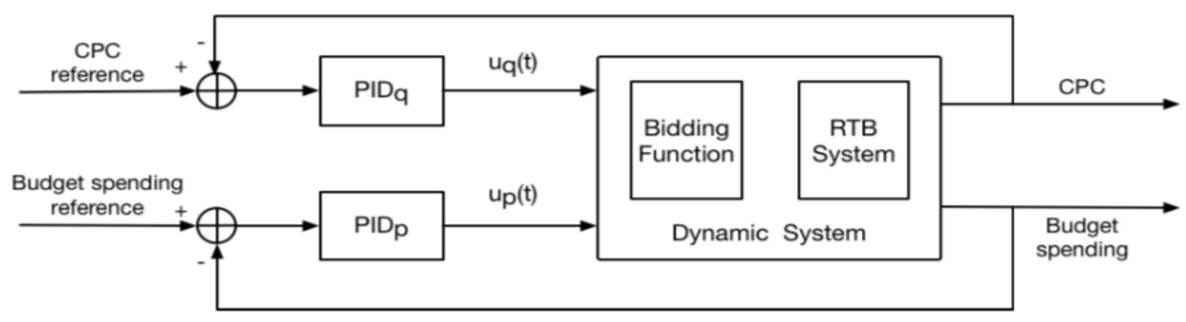


Figure 6: Independent PID control system

但是，这两者并不是完全独立的，比如说为了保点击成本进行提价或降价也会影响到预算的使用，反之亦然。而关于这个问题的研究，paper并没有直接采用这个方法，而是通过一个线性模型去拟合。个人认为其可行的原因是，其调控往往会分为多个时间片，然后在每个时间片内进行调控，而在每个时间片内用直线去拟合，理论上只要把时间片切得足够小，最终总体上也能拟合出非线性的曲线

主要建模思想是通过线性模型直接建模变量p和q和目标$cost$、CPC的关系，具体做法如下：

$$\begin{bmatrix} cost \\ CPC \end{bmatrix} = [\text{X b}] \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} \Delta cost \\ \Delta CPC \end{bmatrix} = [\text{X}] \begin{bmatrix} \Delta p \\ \Delta q \end{bmatrix} \quad (17)$$

$$\begin{bmatrix} \Delta p \\ \Delta q \end{bmatrix} = [\text{X}]^{-1} \begin{bmatrix} \Delta cost \\ \Delta CPC \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} u_p^{`}(t) \\ u_q^{`}(t) \end{bmatrix} = \begin{bmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{bmatrix} \begin{bmatrix} u_p(t) \\ u_q(t) \end{bmatrix} \quad (19)$$

公式(16)里的X和b分别表示2×2和2×1的矩阵，展开后其实就是两个线性回归模型。进一步地，公式(17)表示的是给定需要控制的$\Delta cost$和ΔCPC(调价是分时间片进行调控的，在每次调控前都可以根据当前累积消耗和成本等后验数据，进而计算当前时间片需要调控的$\Delta cost$和ΔCPC)，可以对p和q分别进行Δp和Δq的调控达到目标

其实到了公式(17)已经可以进行调控了，只是调控的方式和paper中的不太一样：首先需要获取公式(17)中的X，而X中的参数其实是可以通过训练数据获取的，训练的数据集从当前时间往前的若干时间片内的$(\Delta p, \Delta q, \Delta p, \Delta q)$，然后$X$就可以通过常规的训练方式获取。这样在每个时间片进行调价时，只需要计算好的X和下一时间片的调控目标：$\Delta cost$、ΔCPC，就能够得到最优的Δp和Δq

公式(18)是在公式(17)的基础上乘上矩阵X的逆得到的，公式(19)则是paper提到的调控方式：首先通过PID调控方式将公式(18)中的$\Delta cost$和ΔCPC变为$u_p(t)$和$u_q(t)$，同时只用两个变量α和β来近似矩阵X的逆，并认为p和q的控制信号$u_p^{'}(t)$和$u_q^{'}(t)$ 是$u_p(t)$和$u_q(t)$的线性组合。总体的调控系统如下图所示：

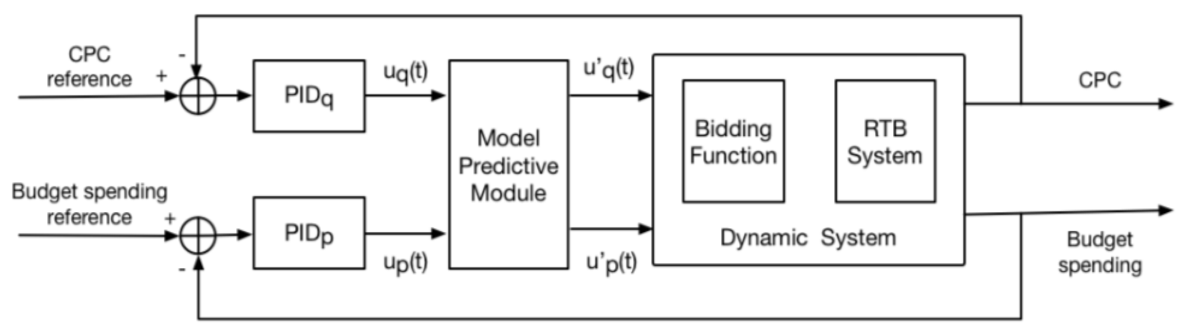


Figure 7: Model predictive PID control system

Optimized Cost per Mille in Feeds Advertising（2020）

论文链接：[Optimized Cost per Mille in Feeds Advertising]

论文主要贡献：
1）对OCPM竞价优化问题进行了较为深入的分析
2）将OCPM竞价优化稳定抽象为一个强化学习问题，通过经典的强化学习算法来求解

拍卖中有三种传统的定价方法，CPC、CPA和CPM。更具体地说，CPM更适合品牌推广和保持品牌知名度，CPC和CPA更适合即时销售增长。最近，为了更好地满足不同的商业目标，出现了更多的定价方法，如ECPC、OCPC以及OCPM等。与传统方法相比，这些新的定价方法都在试图优化转化成本

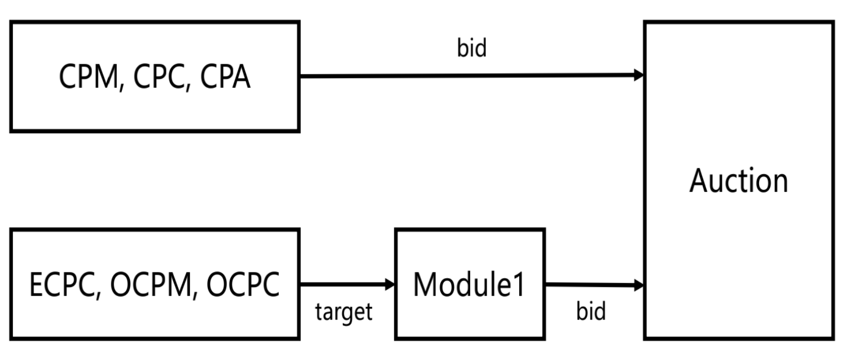


Figure 1: Different pricing methods in advertising system

CPC、CPA和CPM的广告主需要依据他们的目标手动地设置出价，ECPC、OCPC以及OCPM的广告主只需要设置目标成本即可。虽然他们必须按照点击或曝光付费，但平台会自动出价，以满足他们的目标成本。与传统方法相比，无论是从广告主还是从平台的角度来看，这些方法都具有很多优点：

- **对于广告主来说，这些定价方法使得竞价优化更加方便。平台必须对他们的收入、转化率负责，并实现竞价和流量质量在PV粒度的更好匹配**

- 对于平台而言，这些定价方法可以将转化率的不确定性风险转移给广告主。在CPA中，广告主优化广告上下文的动机较小，因为他们只需要在发生转换时进行付费。然而，在这些新的定价方法中，平台会自动降低转化率较低的广告出价，以满足他们的平均转化成本目标。因此，如果广告主不提供具有吸引力的广告，那他们赢得的拍卖和转化就会很少

本文主要针对OCPM竞价优化进行研究。首先，进行问题定义：

$$\begin{aligned} &max \quad \sum_{l=1}^{|\tau_i|} \alpha_i^l \\ &s.t. \quad \frac{\sum_{l=1}^{|\tau_i|} p_i^l}{\sum_{l=1}^{|\tau_i|} \alpha_i^l} \le v_i \end{aligned} \quad (1)$$

其中，$\alpha_i^l \in (0,1)$是一个二进制变量，表示广告主i在拍卖l中的转化数量，p_i^l表示广告在拍卖l中的消耗，v_i表示广告的目标转化成本。问题中唯一的约束条件是平均转化成本应不大于目标转化成本v_i。ROI的表示如公式(2)所示：

$$roi_i^l = \frac{\sum_{l'=1}^{l} p_i^{l'}}{v_i \sum_{l'}^{l} \alpha_i^{l'}} \quad (2)$$

竞价机制包含分配和计费两部分。以VCG拍卖机制为例，它由两个函数$M=(\sigma,p)$构成。VCG拍卖机制目的是找到social welfare最大化的分配方案，计费则按照广告参加竞价给其它广告带来的损失进行：

- **分配函数** 为σ：$R^n \to R^n$，输入广告主的出价，输出n维向量表示slot分配结果。其中$\sigma_i = j \le m$表示将slot j分配给广告主i，$\sigma_i = m+1$表示广告主i未竞得任何slot
- **计费函数** 为p：$R^n \to R^n$，输入广告主的出价，输出n维向量表示对每个广告主的计费，即如果广告主i竞得slot σ_i，需要支付费用p_i

在auction l中，存在四种不同类型的广告主，分别是CPC、CPA、CPM和OCPM，分配规则由公式(3)决定：

$$\sigma^l = arg\max_{\sigma^l} \sum_{i=1}^{n} \beta_{i,\sigma_i^l}^l \cdot b_i^l \quad (3)$$

其中，$\beta_{i,\sigma_i^l}^l$在不同的定价方法中是不同的：

- 对于CPC广告主：$\beta_{i,\sigma_i^l}^l = pctr_{i,\sigma_i^l}^l$
- 针对CPM广告主：$\beta_{i,\sigma_i^l}^l = 1$
- 针对CPA或OCPM广告主：$\beta_{i,\sigma_i^l}^l = pctr_{i,\sigma_i^l}^l \cdot pcvr_{i,\sigma_i^l}^l$

论文把竞价优化建模为一个强化学习问题，针对auction l，有下面的相关定义：

- **State** s_i^l：对于广告主i，$s_i^l = < v_i, t, roi_i^l, \vec{auct} >$，其中，$t$表示当前时间，$\vec{auct}$表示我们可以从广告环境中获取的与拍卖相关的特征向量
- **Action** a_i^l：出价
- **Reward** $r_i(s_i^l, a_i^l)$：在状态s_i^l下通过动作a_i^l获得的收益
- **Policy** $\pi(s_i^l)$：在状态s_i^l下执行动作$\pi(s_i^l)$
- **Episode ep**：本文将一天作为一个episode

问题的目标是在状态s_i^l寻找$\pi(.)$使得期望累计回报(reward)最大化：

$$\sum_{l=1}^{|\tau_i|} \gamma^{l-1} r_i(s_i^l, a_i^l)$$

OCPM的目标是在平均转化成本约束下最大化转化数量，然而，直接使用转化数来设计reward函数有以下两个问题：

- 转化行为稀疏。实际中转化行为较少，不同的出价可能会导致相同的结果(例如均无转化)，只能为训练提供有限的信息。文中给出的定理指出，广告主消耗越大，则期望转化数越大，因此，在reward函数采用消耗而不是实际转化数
- 转化成本限制。指在reward函数中需要对超出转化成本的情况进行惩罚

因此，奖励函数设计为：

$$r_i(s_i^l, a_i^l) = p_i^l - max\{\lambda(p_i^l - \beta_{i,\sigma_i^l}^l \cdot v_i, 0)\} \quad (4)$$

λ是一个大于0的参数，如果挑选合适，则可以在成本约束被打破的情况下时，reward总是负的。λ的挑选在后面的篇幅中会有介绍

得到a_i^l后，可生成出价：

$$b_i^l = v_i \cdot (1 + \alpha_i^l) \quad (5)$$

其中，v_i为基础项，a_i^l调整项。没有直接产生出价b_i^l的原因是不同行业的广告主出价范围可能相差很大，真正重要的是出价相对转化成本v_i的比例

文中提到，模型预估经常会有高估&低估的问题，一方面用户真实行为受众多因素影响，如信息流中上下文item的影响；另一方面，真实行为α_i^l可以认为服从一个二项分布(参数为$\beta_{i,\sigma_i^l}^l$)，即使$\beta_{i,\sigma_i^l}^l$预估得足够准也可能存在方差。如果$\beta_{i,\sigma_i^l}^l$存在高估问题，则很容易出现转化成本超额的问题

因此，本文提出了一个基于IQN的ROI-sensitive算法。用$Q_\tau(s,a)$表示变量$Q(s,a)$在分布$\tau \sim U([0,1])$的分位数函数，用$p:[0,1] \to [0,1$ 表示distortion risk measure。基于此，在$p(.)$下的期望$Q(s,a)$可以用公式(6)表示：

$$Q_p(s,a) = E_{\tau \sim u([0,1])}[Q_{p(\tau)}(s,a)] \quad (6)$$

对应的策略函数为：

$$\pi_p(s) = arg\max_a Q_p(s,a) \quad (7)$$

随着roi_i^l的改变，可以使用不同的$p(.)$。例如，如果roi_i^l偏高，$p(.)$可以给与$Q(s,a)$的低分位数更多的权重。$p(.)$的形式如公式(12)所示：

$$p(\tau) = \begin{cases} \tau \ \ if \ roi_i^l \le \theta \\ min\{\tau, \hat{\tau} \sim U([0,1])\} \ \ otherwise \end{cases}$$

其中，θ是一个预选定义的常量。当roi_i^l比θ高时，agent将会采取低回报的风险规避策略。根据定理3.1，此时agent会给出相对较低的出价。根据这个方法可以在RSDRL中建模ROI敏感的agent

算法的整体框架如下图所示，内层循环中，agent根据ϵ贪心策略选择并执行action，然后，为OCPM广告主生成竞价。基于VCG拍卖中的分配和付费规则，可以得到reward和下一个阶段的state。当广告被展现给用户时，可以获取α_i^l，用于更新roi_i^l。最后，网络根据IQN loss来执行梯度下降进而获得更新

Algorithm 1 RSDRL

Randomly initialize weights μ for network Q
Randomly initialize weights $\mu' = \mu$ for target network Q'
Initialize replay memory D
Initialize $roi_i^l = 0$

for episode = 1 to K **do**
  for l=1 to $|\mathcal{I}_i|$ **do**
    Get ρ based on Equation (12)
    With probability ϵ select a random action a_i^l
    Otherwise get action a_i^l according to Equation (11)
    Bid with $v_i \cdot (1 + a_i^l)$
    Get reward r_i^l
    Observe next state s_i^{l+1}
    Store transition $(s_i^l, s_i^{l+1}, a_i^l, r_i^l)$ in D
    Update roi_i^l
    Sample random mini-batch of transitions from D
    Perform a gradient descent step on IQN loss with respect to the μ
    Every C steps reset $Q' = Q$
  end for
end for

文中还对几个参数进行了讨论，这里列一下结论：
1）在拍卖l中，应设置$\lambda \geq \frac{p_i^l}{p_i^l - \beta_{i,a_i^l}^l}$，此时，任何成本约束条件的破坏都会得到一个负的reward
2）调控因子$a_i^l \geq 0$
3）对于roi_i^l的计算，存在以下两个问题：i）在刚开始时并没有很好地定义：假设第一个转化发生在拍卖l中，那么$\sum_{\hat{l}=1}^{l=1} \alpha_i^{\hat{l}} = 0$，这使得$roi_i^{l-1}$没有意义；ii）它对于某些广告的转化成本是不敏感的：假设有一个广告，$roi_i^l = 1$且$\sum_{\hat{l}=1}^{l} \to \infty$。由于其分母太大，对于下一次的转化，其消耗相对于分母来说比较小，因此其ROI总是等于1，即使后面这些转化的成本比较低
为了解决这两个问题，在实现时使用了$k-ROI$敏感的agent。如公式(8)所示：

$$roi_i^l = \frac{\sum_{\hat{l}=t_k^l}^{l} p_i^{\hat{l}}}{k * v_i} \quad (13)$$

其中:

$$t_k^l = min\{\tilde{l} | \lfloor \frac{\sum_{\hat{l}=1}^{\tilde{l}} a_i^{\hat{l}}}{k} \rfloor + k > \lfloor \frac{\sum_{\hat{l}=1}^{l} a_i^{\hat{l}}}{k} \}$$

A Unified Solution to Constrained Bidding in Online Display Advertising(2021)

论文链接：[A Unified Solution to Constrained Bidding in Online Display Advertising]

论文主要贡献：

- **针对所有出价场景，提出并证明了一种统一的出价策略**
- **提出了一种降低强化学习模型学习复杂度的方**

在广告拍卖期间，广告活动的共同目标是最大化曝光价值，即$max \sum_i v_i x_i$，其中，v_i是曝光价值，x_i是表示广告是否赢得曝光i的二元变量。预算约束可以表示为$\sum_i c_i x_i \leq B$，KPI约束比较复杂，它可以分为两类：第一类是与消耗相关的约束(CR)，它主要对特定广告事件的单位成本做限制，如CPC和CPA；第二类是与消耗无关的约束(NCR)，它主要对广告的平均影响力进行约束，如CTR、CPI等。KPI约束的统一表示如公式(1)所示：

$$\frac{\sum_i C_{ij} x_i}{\sum_i P_{ij} x_i} \leq k_j \quad (1)$$

其中，k_j是约束j的上届，它由广告主提供。p_{ij}可以是任何的效果指标或者常量，$C_{ij} = c_i ICR_j + q_{ij}(1 - ICR_j)$，其中，$q_{ij}$可以是任何的效果指标或常量，$ICR_j$表示约束$j$是否是CR

总之，考虑到广告主的价值需求、预算和KPI约束，可以将广告活动的优化问题定义如下：

$$\begin{aligned} \max_{x_i} \ & \sum_i v_i x_i \\ s.t. \ & \sum_i c_i x_i \leq B \\ & \frac{\sum_i C_{ij} x_i}{\sum_i P_{ij} x_i} \leq k_j, \ \forall j \\ & x_i \leq 1, \ \forall i \\ & x_i \geq 0, \ \forall i \end{aligned} \quad (LP1)$$

LP1的对偶问题定义如下：

$$\begin{aligned} \min_{\alpha, \beta_j, \gamma_j} \ & \beta\alpha + \sum_i r_i \\ s.t. \ & c_i \alpha + \sum_j (C_{ij} - k_j P_{ij}) \beta_j + r_i \geq v_i, \ \forall i \\ & \alpha \geq 0 \\ & \beta_j \geq 0, \ \forall i \\ & r_i \geq 0, \ \forall i \end{aligned} \quad (LP2)$$

根据C_{ij}的定义，我们可以重写LP2中的第一个等式：

$$\underbrace{(v_i - \Sigma_j \beta_j (\mathbb{q}_{ij}(1 - \mathbb{1}_{CR_j}) - \mathbb{k}_j \mathbb{p}_{ij}))}_{P_{NCR}} - \underbrace{(\alpha + \Sigma_j \beta_j \mathbb{1}_{CR_j})}_{P_{CR}} c_i - r_i \leq 0 \quad (3)$$

其中，P_{NCR}表示消耗无关的因子，P_{CR}表示消耗相关的因子

用x_i^*表示原问题LP1的最优解，α^*, r_i^*和β_j^*表示最优问题$LP2$的最优解。根据互补松弛定理，可以得到：

$$x_i^*(P_{NCR}^* - P_{CR}^* c_i - r_i) = 0,\ \forall i \ \ (4)$$
$$(x_i^* - 1)r_i^* = 0,\ \forall i \ \ (5)$$

可以将曝光i的出价设置为$b_i = P_{NCR}^*/P_{CR}^*$，然后，可以分别将公式(3)转化为公式(6)，公式(4)转化为公式(7)：

$$(b_i^* - c_i)P_{CR}^* - r_i^* \le 0,\ \forall i \ \ (6)$$
$$x_i^*[(b_i^* - c_i)P_{CR}^* - r_i] = 0,\ \forall i \ \ (7)$$

由此可以推断：

- 如果一个广告活动赢得了曝光i，这意味着$x_i^* > 0$。根据公式(7)，$(b_i^* - c_i)P_{CR}^* - r_i^* = 0$。同时，由于$r_i^* \ge 0, P_{CR}^* \ge 0$，因此，可以推断出$b_i^* \ge c_i$
- 如果一个广告活动没有赢得曝光i，这意味着$x_i^* = 0$，从等式(5)可以推断，$r_i^* = 0$。由于$P_{CR}^* \ge 0$，根据公式(6)可得$b_i^* \le c_i$

总结来说，对于曝光i，出价b_i^*都会产生最优分配x_i^*。因此，最优出价是b_i^*，同时，为了更简洁明了，我们将b_i^*写成了如下形式

$$b_i^* = \frac{P_{NCR}^*}{P_{CR}^*} = \frac{v_i - \Sigma_j \beta_j^* (\mathbb{q}_{ij}(1 - \mathbb{1}_{CR_j}) - \mathbb{k}_j \mathbb{p}_{ij})}{\alpha^* + \Sigma_j \beta_j^* \mathbb{1}_{CR_j}} \qquad (8)$$
$$= \underbrace{(\frac{1}{\alpha^* + \Sigma_j \beta_j^* \mathbb{1}_{CR_j}})}_{w_0^*} v_i \qquad (9)$$
$$- \Sigma_j \underbrace{(\frac{\beta_j^*}{\alpha^* + \Sigma_j \beta_j^* \mathbb{1}_{CR_j}})}_{w_j^*} (\mathbb{q}_{ij}(1 - \mathbb{1}_{CR_j}) - \mathbb{k}_j \mathbb{p}_{ij}) \qquad (10)$$
$$= w_0^* v_i - \Sigma_j w_j^* (\mathbb{q}_{ij}(1 - \mathbb{1}_{CR_j}) - \mathbb{k}_j \mathbb{p}_{ij}) \qquad (11)$$

对于一个有M个约束条件且希望最大化曝光价值的广告活动来说，最优出价bid_i^*是由M+1个参数决定的$w_k^*, k \in [0, \ldots, M]$。使用这个最优出价函数，可以通过学习M+1个参数去解决带有约束的最优出价问题

将参数调节问题定义为马尔科夫决策过程，这个马尔科夫决策过程是由一系列描述广告状态的state S构成的。agent的参数调整action空间$A = A_0 \times A_1 \times \ldots \times A_M \in R^{M+1}$。在每一个时间步$t$，agent会基于当前状态$s_t \in S$，依据它的policy $\pi : S \to A$，执行一系列动作$a_{0t}, a_{1t}, \ldots, a_{Mt} \in A$去修改参数$w_{kt}, k \in [0, \ldots, M]$；然后，根据状态转换过程：$\gamma : S \times A \to \Omega(S)$，state将会转换到下一个state，其中，$\Omega(S)$是$S$上的概率分布集合；环境将会基于一个当前状态state和action的函数$r_t : S \times A \to R \subseteq \mathrm{R}$返回一个即时reward。agent的目标是最大化总期望回报$R = \sum_{t=1}^{T} \gamma^{t-1} r_t$，其中$\gamma$是折扣因子，$T$是时间范围。建模详细描述如下：

- S：state是描述广告状态的信息集合，这些信息应该主要反映时间、预算消耗以及KPI约束满足情况，如剩余时间、剩余预算、预算、预算消耗速度以及约束j的当前KPI ratio
- A：在每个时间步t，每个agent将会执行一个$M+1$维的action向量$\vec{a} = (a_{0t}, \ldots, a_{Mt})$去修改$M+1$维的参数向量$\vec{w}_t = (w_{0t}, \ldots, w_{Mt})$，形如：$\vec{w}_{t+1} = \vec{t}_t(1 + \vec{a}_t)$，其中，$a_{kt} \in (-1.0, +\infty), \forall k \in [0, M]$
- r_t：在step t，用O表示在step t和step t+1之间的曝光集合，因此，$r_t = \sum_{i \in O} x_i v_i$是从$O$中赢得曝光的总价值
- Γ：我们使用model-free的方法来解决我们的问题，因此，不需要显示地对动态转换建模
- γ：我们设置reward的折扣率为$\gamma = 1$，因为每个广告活动的有效性都需要从一个日常角度来评估

文中证明了：**对于在每个step t上的子问题，最优的action序列是将当前的$\vec{w}_t$修改为$\vec{w}_t^*$，并在接下来的step中固定不变**

文章利用DDPG作为强化学习算法的实现，并基于上述证明，大大简化了强化学习模型的学习复杂度

Algorithm 1: Unified Solution to Constrained Bidding

Initialize a $M+1$ dimensional random process $\mathcal{E}$;
Initialize replay memory $\mathcal{M}$ with capacity N;
Initialize actor π_θ with weights θ;
Initialize critic Q_η with weights η;
Set Batch Size as BS;
Let $\vec{w}^*$ be the optimal parameter vector $(w_0^*, \ldots, w_M^*)$;
Let R^* be the theoretically optimal result;
while *not convergent* **do**
    Randomly choose an ad campaign and simulate SPA;
    Set $\vec{w}_1 = \vec{w}^* + \mathcal{E}$;
    Bid with $\vec{w}_1$ at time-step 1 and get reward r_1;
    Set $R = r_1$;
    for $t = 2$ **to** T **do**
        Observe state s_t;
        Get action vector $\vec{a}_t = \pi_\theta(s_t) + \mathcal{E}$;
        Set $\vec{w}_t = \vec{w}_{t-1} \cdot (1 + \vec{a}_t)$;
        Bid with $\vec{w}_t$ at time-step t and get reward r_t;
        Set $R = R + r_t$ and $V = 0$;
        for $\tau = t+1$ **to** T **do**
            Bid with $\vec{w}_t$ at time-step τ and get reward r_τ;
            Set $V = V + r_\tau$;
        end
        Calculate penalty p_j for KPI constraint j;
        Set $\mathcal{G} = \min\{(R+V)/R^*, 1.0\} - \Sigma_j p_j$;
        Store $(s_t, \vec{a}_t, \mathcal{G})$ in $\mathcal{M}$;
        Sample BS $(s^k, \vec{a}^k, \mathcal{G}^k)$ tuples from $\mathcal{M}$;
        Update Critic Q_η by minimizing the loss
        $\mathcal{L}(\eta) = \frac{1}{\mathrm{BS}} \sum_k (\mathcal{G}^k - Q_\eta(s^k, \vec{a}^k))^2$;
        Update Actor π_θ by policy gradient:
        $\nabla_\theta J \approx \frac{1}{\mathrm{BS}} \sum_k \nabla_\theta \pi(s^k) \nabla_{\vec{a}} Q^\pi(s^k, \vec{a})|_{\vec{a}=\pi(s^k)}$;
    end
end