Cover Ratio Maximization

Problem Definition:

Given 2 product datasets $D = \{d_1,\ d_2,\ \ldots, d_m\}$ and $P = \{p_1,\ p_2,\ \ldots, p_n\}$, product creation budget B, a positive integer k, a creation cost function C, a user preference tuple set $W = \{w_1, w_2, \ldots, w_r\}$, a store's($s$) score for a user w_i is defined as $w_i \cdot s$, find a new product p such that satisfies the constrain $C(p) \leq B$ and maximize $cp(p,\ P, k) = \frac{|\{w|\forall w \in W, \{P \cup \{p\}\} \cap TopK(w) \neq \emptyset\}|}{|W|}$, which $C(p) = c \cdot p$; $TopK(w)$ means the stores whose scores rank top k with respect to user w.

problem example:

There is a fast food shop call McDonald's with a lot of chain stores $P = \{p_1,\ p_2,\ \ldots, p_n\}$. McDonald's also has a lot of competitor $D = \{d_1,\ d_2,\ \ldots, d_m\}$, such as KFC and Burger King. These stores compete each other at geographical location, price, taste and other aspects. Even the stores with the same brand will compete each other to attract(cover) customers. For now, McDonald's is to open a new store p. But the problem is where p should be so that it and the formal $P = \{p_1,\ p_2,\ \ldots, p_n\}$ together can cover as more as customers(A store's score rank top k for a customer will cover this customer). It is of course that the cost of this new McDonald's store p should not exceed B.

Solution:

1. We notice that

   For any solution $\boldsymbol{p_1}$ on $\boldsymbol{C(p) < c \cdot p}$, there will always be a solution $\boldsymbol{p_2}$ on $\boldsymbol{C(p) = c \cdot p}$ such that $\boldsymbol{score(p_2) > score(p_1)}$.

   See proof 1.

   It is obvious and easy to proof that the greater the score of store, the more user it could cover. Therefore, we only need to focus on $C(p) = c \cdot p$.
2. **There exists a $\boldsymbol{w_i}$ in $\boldsymbol{W}$ such that one of the solutions on the intersection between $\boldsymbol{w_i \cdot p = Score_{ik}}$ and $\boldsymbol{C(p) = c \cdot p}$ maximize $\boldsymbol{cp(p,\ P, k)}$, $\boldsymbol{Score_{ik}}$ means the k-th ranking score respecting to $\boldsymbol{w_i}$.**

   See proof 2.
3. Each intersection divide space $\boldsymbol{C(p) = c \cdot p}$ into two part, one is $\boldsymbol{w_i \cdot p > Score_{ik}}$, another is $\boldsymbol{w_i \cdot p < Score_{ik}}$, if we use method CTA in $kSPR$ to count the half-space number we can find the optimal region on $\boldsymbol{C(p) = c \cdot p}$ that maximize $\boldsymbol{cp(p,\ P, k)}$

   If dimension $n = 2$, the intersections are points;
   $n = 3$, the intersections are lines;
   …

   The time complexity is $O(m^{d-1})$, where m is the number of w_i that the intersection part between $\boldsymbol{w_i \cdot p = Score_{ik}}$ **and** $\boldsymbol{C(p) = c \cdot p}$ is not empty and w_i should not be covered by any option in P (this means we don't need all w). Notice that time complexity of CTA in $kSPR$ is $O(n^d)$, where n is the number of options.
4. We can also consider apply $P - CTA$ and $LP - CTA$

Proof

1. For any solution p_1 on $C(p) < c \cdot p$, there will always be a solution p_2 on $C(p) = c \cdot p$ such that $score(p_2) > score(p_1)$

   For any p_1, $p_2 = p_1 \cdot \frac{B}{p_1 \cdot c}$ is on $C(p) = c \cdot p$
   Then $w \cdot p_2 = w \cdot p_1 \cdot \frac{B}{p_1 \cdot c} > w \cdot p_1 \cdot 1$
   Which means $score(p_2) > score(p_1)$

2. Suppose the optimal solution p_{opt} is not on any intersection between $w_i \cdot p = Score_{ik}$ and $C(p) = c \cdot p$, and for any $w\ in\ the\ set\ W' = \{w_1{}', w_2{}', \ldots, w_i{}'\} \subset W$, we have $w \cdot p_{opt} > Score_k$.
   Because of the continuous of solution space, there must be a solution p' such that for any w in $W'' = \{w_1'', w_2'', \ldots, w_j''\} \subset W'$, we have $w \cdot p' > Score_k$; for any w in $W' \Delta W''$, we have $w \cdot p' \geq Score_k$. This means p' is also an optimal solution. And p' is at least on the at least one of the intersections, so proof ends.