Discrete Optimization: Homework #8, Ex. #1

Denis Steffen, Yann Eberhard & Gaëtan Bossy

April 20, 2018

We use basic binary search on each of the dimension. For every $1 \leq i \leq n$, we use the oracle algorithm at most $log_2(B)$ times and find the i-th coordinate of the red point. By doing this n times, we can find every coordinate of the unique red point. Thus our main algorithm will call the secondary algorithm at most n times, and the secondary algorithm will call the oracle algorithm $O(log(b))$ times, which means the main algorithm will call the oracle algorithm $O(n \cdot log(B))$ times.

Main algorithm: $main(B)$, with $B \in \mathbb{N}$
For $i = 1 \to n$:

$x_i^* = alg(i, 0, B)$

return x^*

Secondary algorithm: $alg(i, a, b)$, with i an index, $a, b \in \mathbb{N}$
If $a = b$

return a

else

if $oracle(i, a + floor(b - a/2)) = true$

return $alg(i, a, a + floor(b - a/2))$

else

return $alg(i, a + floor(b - a/2), b)$