# APPENDIX

## Proof of Theorem 5.1

THEOREM 5.1. *Assume $\epsilon = B(\frac{1}{2})^i$, where $i \in Z^+$. If $\epsilon \geq d_{min}$, in the worst case, GBQ uses the same number of queries as linear search; however, if $\epsilon < d_{min}$, GBQ always uses fewer queries.*

PROOF. First, we notice that for a given $\epsilon$ that satisfies $\epsilon = B(\frac{1}{2})^i$, the linear search with $B/\epsilon$ queries can be reimplemented as a naive binary search as follows: first, we query $r(B)$; next, $r(B/2)$; then, $r(B/4)$ and $r(3B/4)$, and so on. In total, we need $i+1$ iterations to achieve $\epsilon$. In terms of queries, we need: $1 + 1 + 2 + \ldots + 2^{(i-1)} = 2^i$, which is exactly $B/\epsilon$.

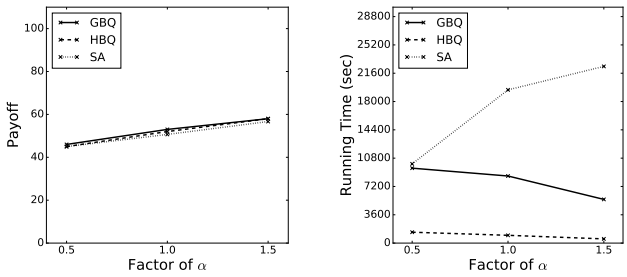
When $\epsilon \geq d_{min}$, it is possible that each query will find a new interval, and thus $2^i$ queries may be required to obtain intervals of width $\epsilon$ in the worst case. However, when $\epsilon < d_{min}$, we are guaranteed that GBQ has captured all thresholds at some iteration $\hat{i}+1$ before the iteration $i+1$. Otherwise, there must be two thresholds that are in the same interval, therefore, $d_{min} < \epsilon$, which is a contradiction. First, we compare total number of queries used by GBQ and the naive binary search up to iteration $i+1$. Among these iterations, we are not guaranteed to capture all thresholds, so GBQ will need the same number of $(2^{\hat{i}})$ queries as the naive binary search in the worst case. Second, we compare the two algorithms for iterations after $\hat{i}+1$. Starting from iteration $(\hat{i}+2)$, GBQ only sends a fixed number of queries (equal to the actual number of thresholds, denoted by $J$), as the generalized binary search will skip intervals that do not have thresholds. The total number of queries used by GBQ after $(\hat{i}+1)$ is $J(i-\hat{i})$. By contrast, the naive binary search (equivalent to linear search as we have shown) has to explore all intervals available for each iteration. After iteration $\hat{i}+1$, it will send $(2^i - 2^{\hat{i}})$ queries, which is *strictly* larger than $J(i-\hat{i})$. Notice that the number of queries used at iteration $\hat{i}+1$ by the naive binary search satisfies: $2^{\hat{i}-1} \geq J$, when all threshold are captured. In addition, the number of queries used by the naive binary search at iteration $(\hat{i}+2)$ is $2^{\hat{i}}(> 2^{\hat{i}-1})$, which implies that $2^{\hat{i}} > J$. Thus, we have $2^i - 2^{\hat{i}} = 2^{\hat{i}}(2^{i-\hat{i}} - 1) \geq 2^{\hat{i}}(i-\hat{i}) > J(i-\hat{i})$. Therefore, if $\epsilon < d_{min}$, GBQ always uses fewer queries. □

## Additional Experiments

Apart from the experimental results presented in Section 6.2, we also conducted a series of experiments regarding other parameters of the marketing simulator. Particularly, we set the default configuration to Configuration 0 (see Table 1) with a fixed budget 6000, vary one parameter a time and then compare the performance of GBQ, HBQ and SA in terms of utility and running time.

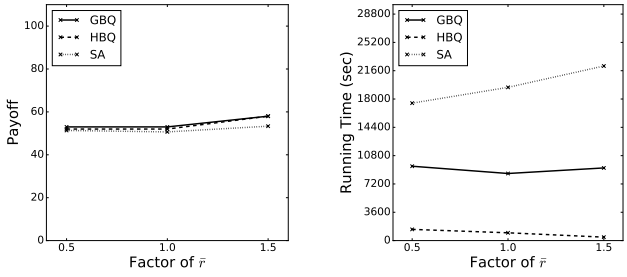
Figure 4 compares utility and running time of GBQ, HBQ and SA over different factors of $\alpha$, which is a parameter defined in the keyword auction simulator. Note that when the factor of $\alpha$ equals to 1, it corresponds to default value of $\alpha$ in Configuration 0, whereas, factor of $\alpha$ equals to 0.5 and 1.5 are 0.5 times and 1.5 times of the default $\alpha$ respectively. Figure 5 compares utility and running time of all three algorithms over different factors of $\bar{r}$ for the direct mailing simulator. Similarly, factor of $\bar{r} = 1$ is the default setting. Figure 6 shows comparison among GBQ, HBQ and SA over different factors of $\beta$ as defined in the broadcast marketing simulator, with factor of $\beta = 1$ the default value. Figure 7 displays comparison among GBQ, HBQ and SA over different options of $R_{dml}$ for the direct mailing simulator. Figure 8 compares GBQ, HBQ and SA over different choices of $R_{brc}$ for the broadcast marketing simulator.

Clearly, in all experimented cases, HBQ is the most efficient achieving competitive utilities significantly faster than GBQ, and both algorithms outperform the simulated annealing baseline.



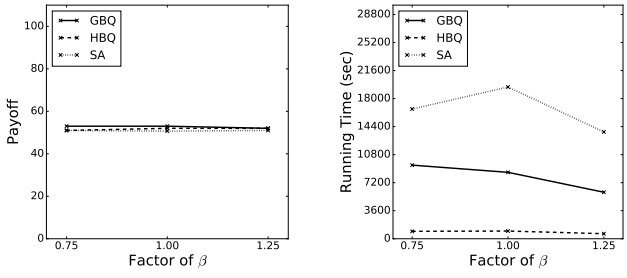

(a) (b)

Figure 4: Payoff (a) & run time (b) comparison among algorithms over different factors of $\alpha$ for online ads marketing.



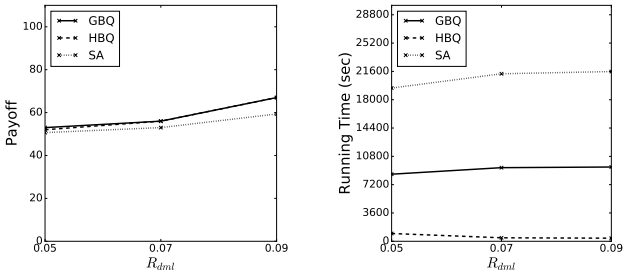

(a) (b)

Figure 5: Payoff (a) & run time (b) comparison among algorithms over different factors of $\bar{r}$ for direct mail marketing.



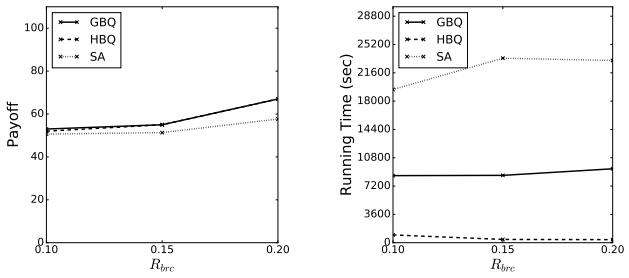

(a) (b)

Figure 6: Payoff (a) & run time (b) comparison among algorithms over different factors of $\beta$ for broadcast marketing.




(a) (b)

Figure 7: Payoff (a) & run time (b) comparison among algorithms over different $R_{dml}$ for direct mail marketing.




(a) (b)

Figure 8: Payoff (a) & run time (b) comparison among algorithms over different $R_{brc}$ for broadcast marketing.