A Supplementary material for Section 2 (Problem Statement and Modeling)

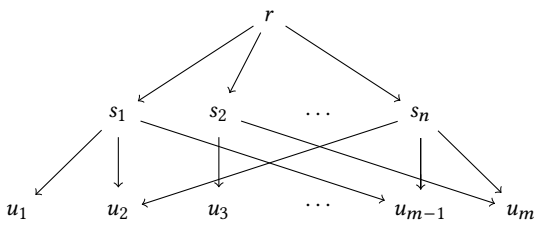


Figure 6: Graphical summarization of the graph G_{sc}

A.1 Graph Crawling Problem

Proposition 4. *Given a website graph $G = (V, E, r, \omega, \lambda)$, a subset $V^* \subseteq V$ and some $B \in \mathbb{R}^+$, determining whether there exists a crawl $T = (V', E')$ of G such that $V^* \subseteq V'$ and $\omega(T) \leqslant B$ is NP-complete; hardness holds even when ω is a constant function.*

Proof. To show NP-completeness, we must show that the problem belongs to NP and is NP-hard.

Let us start with the upper bound. Given a graph $G = (V, E, r, \omega, \lambda)$, we guess a subgraph $T = (V', E')$ of G (which is a polynomial-sized guess). In polynomial time, we check whether T is a r-rooted tree (i.e., whether it is connected, includes r, r has indegree 0 and other nodes indegree 1), we check that V' contains all nodes of V^*, and we check that $\omega(T) \leqslant B$. We accept if and only if these conditions are all satisfied. This yields a nondeterministic polynomial-time algorithm, meaning the problem is in NP.

We now move to the lower bound. Our crawling problem can be seen as a directed variant of the well-known NP-complete *Steiner Tree* [GJ79] problem. NP-hardness of the directed Steiner tree problem is mentioned in the literature (see, e.g., [WW16]), but as it is not formally shown there, we prefer for completeness of the presentation reducing from the set cover problem, a classic NP-hard problem [GJ79]. We denote $\mathcal{U} = \{u_1, \ldots, u_m\}$ a set of m elements called the universe. We also define a collection $\mathcal{S} = \{s_1, \ldots, s_n\}$ of n non-empty subsets, each of them containing some elements of $\mathcal{U}$, such that:

$$\bigcup_{s \in \mathcal{S}} s = \mathcal{U}.$$

In its decision version, the set cover consists in given such a universe and collection, given a natural integer B, determining whether there exists a *cover* $C \subseteq \{s_1, \ldots, s_n\}$ such that $|C| \leqslant B$ and:

$$\bigcup_{s \in C} s = \mathcal{U}.$$

We now propose a polynomial-time many-one reduction of the set cover problem to an instance of the graph crawling problem. We create a website graph $G_{\text{sc}} = (V_{\text{sc}}, E_{\text{sc}}, r, \omega, \lambda)$ as follows. We set V_{sc} to be $\{ u_1, \ldots, u_m, r, s_1, \ldots, s_n \}$, including representations for every element of the universe $\mathcal{U}$, every set of the collection $\mathcal{S}$, as well as a distinct root r (by abuse of notation, we do not distinguish between elements of $\mathcal{U}$, $\mathcal{S}$ and the way they are represented in V_{sc}). We define E_{sc} as $\{(r, s_i) \mid i \in \{1, \ldots, n\}\} \cup \{(s_i, u) \mid u \in s_i, i \in \{1, \ldots, n\}\}$. In other words, in G_{sc} from the origin (root) r, we model each element of $\mathcal{S}$ as a vertex, that can be reached following a dedicated (directed) edge. Finally, for each new vertex $s_i \in \mathcal{S}$, we have as many outgoing edges as there are elements of $\mathcal{U}$ in s_i. Finally, we set ω to be the constant function that assigns cost 1 to every vertex, and λ to be some constant function. The result is a graph in the form of a tree of depth 2, depicted in Figure 6. We fix V^* to be $\mathcal{U}$. We state that there exists $C \subseteq \{s_1, \ldots, s_n\}$ such that $|C| \leqslant B$ and $\bigcup_{i \in C} s_i = \mathcal{U}$ if and only if there exists a crawl T_{sc} of G_{sc} containing all elements of V^* and of total cost $\omega(T_{\text{sc}}) \leqslant |\mathcal{U}| + B + 1$.

Let us explain why this reduction is polynomial-time. In set cover, the universe $\mathcal{U}$ can be described by the number m of its elements, with representation size $\Theta(\log m)$. Each set of $\mathcal{S}$ needs to list every element within this set, so a set s_i has representation size $\Theta(\log m \times |s_i|)$. Finally, B has representation size $\Theta(\log B)$. This yields a total input size of $\Theta((\log m)\left(\sum_{i=1}^{n} |s_i| + 1\right) + \log B)$. Note that $\sum_{i=1}^{n} |s_i| \geqslant \max(m, n)$ so this is $\Omega(\max(m, n) \log m + \log B)$. But then, the construction depicted in Figure 6 can clearly be done in time polynomial in m and n (namely, in $O(m \times n)$ in the worst case where every set of the collection contains every element). The reduction is therefore polynomial-time.

We now proceed to show equivalence between the initial problem known to be NP-hard (set cover) and the graph crawling instance presented above. First, suppose that there exists $C \subseteq \{s_1, \ldots, s_n\}$ such that $|C| \leqslant B$ and $\bigcup_{i \in C} s_i = \mathcal{U}$. Then consider the crawl T_{sc} of G_{sc} formed by including r, every element of C using the edge from r to that element, and every edge from an element of C to an element of $\mathcal{U}$. Since C is a cover, this includes all elements of $\mathcal{U}$. The total cost of this crawl $\omega(T_{\text{sc}}) = 1 + |C| + |\mathcal{U}| \leqslant |\mathcal{U}| + B + 1$.

Now suppose that there exists a crawl T_{sc} of G_{sc} of total cost $\omega(T_{\text{sc}}) \leqslant |\mathcal{U}| + B + 1$. Note that by definition of ω, the cost is just the number of nodes in T_{sc}, and this crawl necessarily includes the root r as well as all vertices of $\mathcal{U}$. The remaining $\leqslant B$ vertices are therefore vertices of $\mathcal{S}$. We pose C to be those. Then $|C| \leqslant B$ and since T_{sc} is a crawl, for every $u \in \mathcal{U}$, there exists at least one $s \in C$ such that the edge (s, u) is in T_{sc}, meaning that $u \in s$. We indeed have $\mathcal{U} = \bigcup_{s \in C} s$. □

A.2 Data Acquisition as Graph Crawling

Here is the full list of the 38 MIME types used to identify targets in our implementation:

```
application/csv
application/json
application/msword
application/octet-stream
application/pdf
application/rdf+xml
application/rss+xml
application/vnd.ms-excel
application/vnd.ms-excel.sheet.macroenabled.12
application/vnd.oasis.opendocument.presentation
application/vnd.oasis.opendocument.spreadsheet
application/vnd.oasis.opendocument.text
application/vnd.openxmlformats-officedocument.presentationml.presentation
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
application/vnd.openxmlformats-officedocument.wordprocessingml.document
application/vnd.openxmlformats-officedocument.wordprocessingml.template
application/vnd.rar
application/x-7z-compressed
application/x-csv
application/x-gtar
application/x-gzip
application/xml
application/x-pdf
application/x-rar-compressed
application/x-tar
application/x-yaml
application/x-zip-compressed
application/yaml
application/zip
application/zip-compressed
text/comma-separated-values
text/csv
text/json
text/plain
text/x-comma-separated-values
text/x-csv
text/x-yaml
text/yaml
```

B Supplementary material for Section 4 (Experimental Results)

Figure 7 completes Figure 4 with the eight remaining websites plots that could not fit in the paper (see Table 1 for websites characteristics). Figures 8 to 13 present exhaustive graphical results of the hyper-parameters studies conducted on the 11 fully-crawled websites. Especially, Figures 8 and 9 study the exploration–exploitation coefficient α, Figures 10 and 11 study the choice of n in n-grams used in tag paths vector representation, and Figures 12 and 13 study the similarity threshold θ.

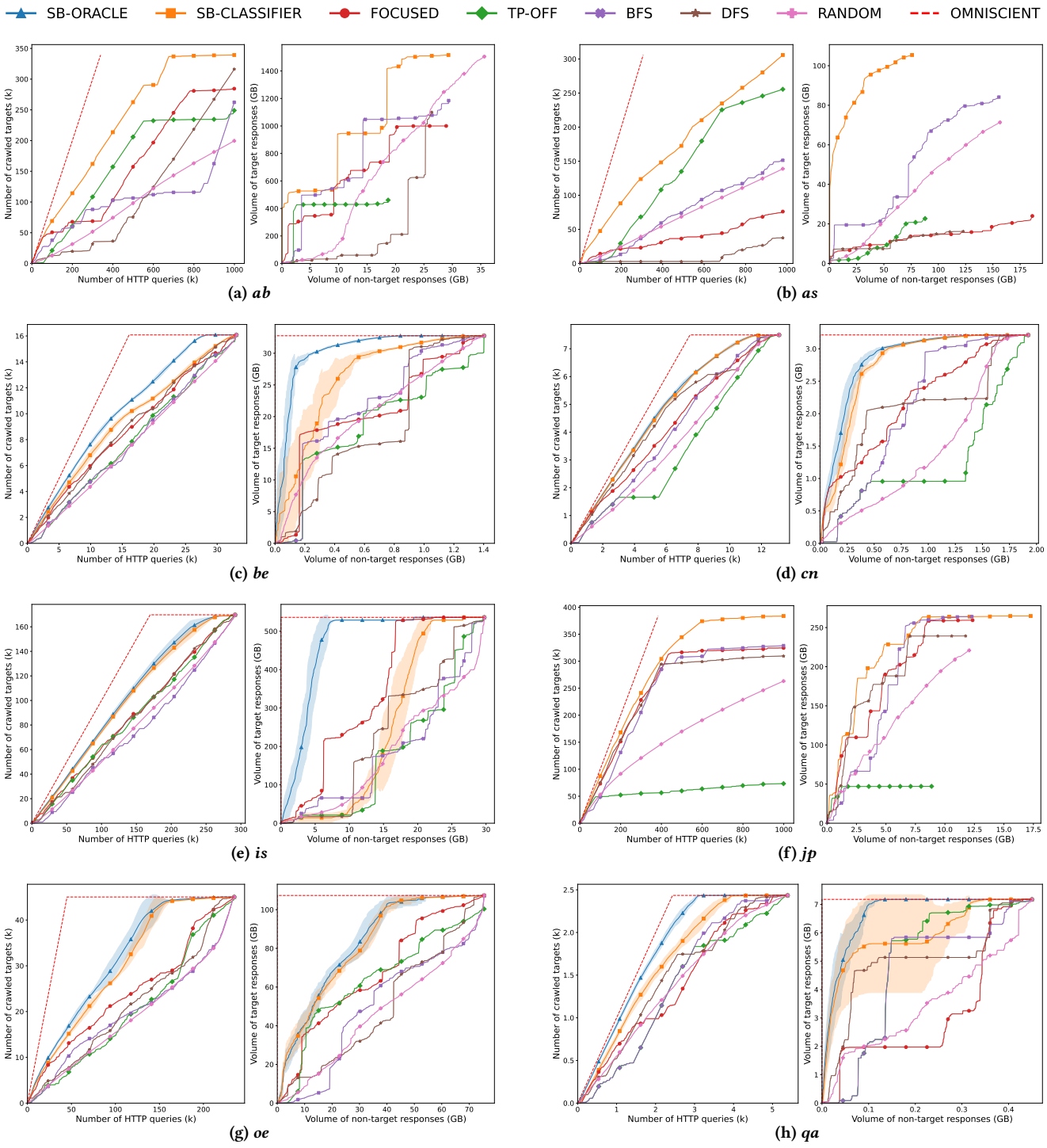


Figure 7: Comparison of different crawler performance for the 8 websites not presented in Figure 4 due to space reasons

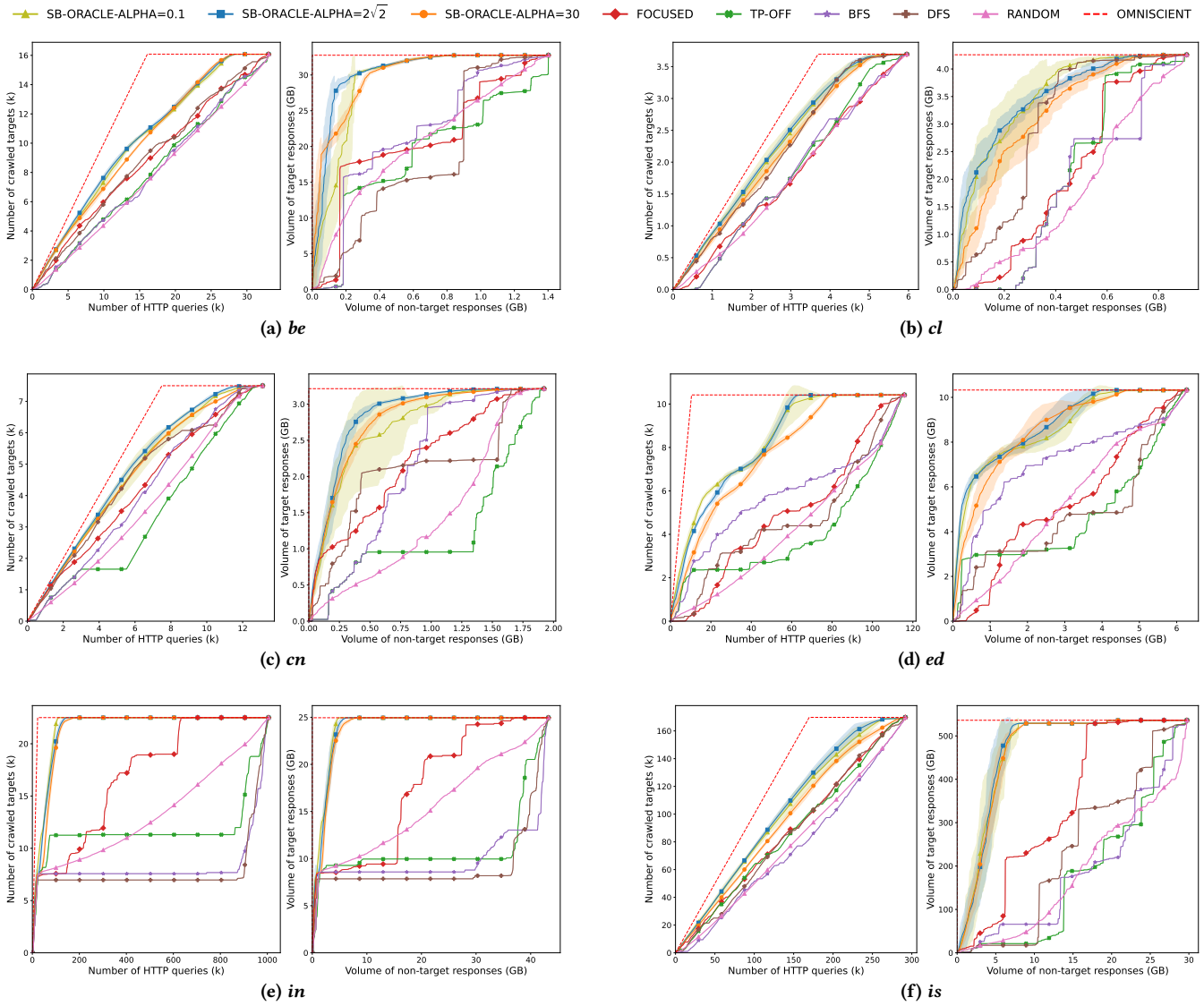
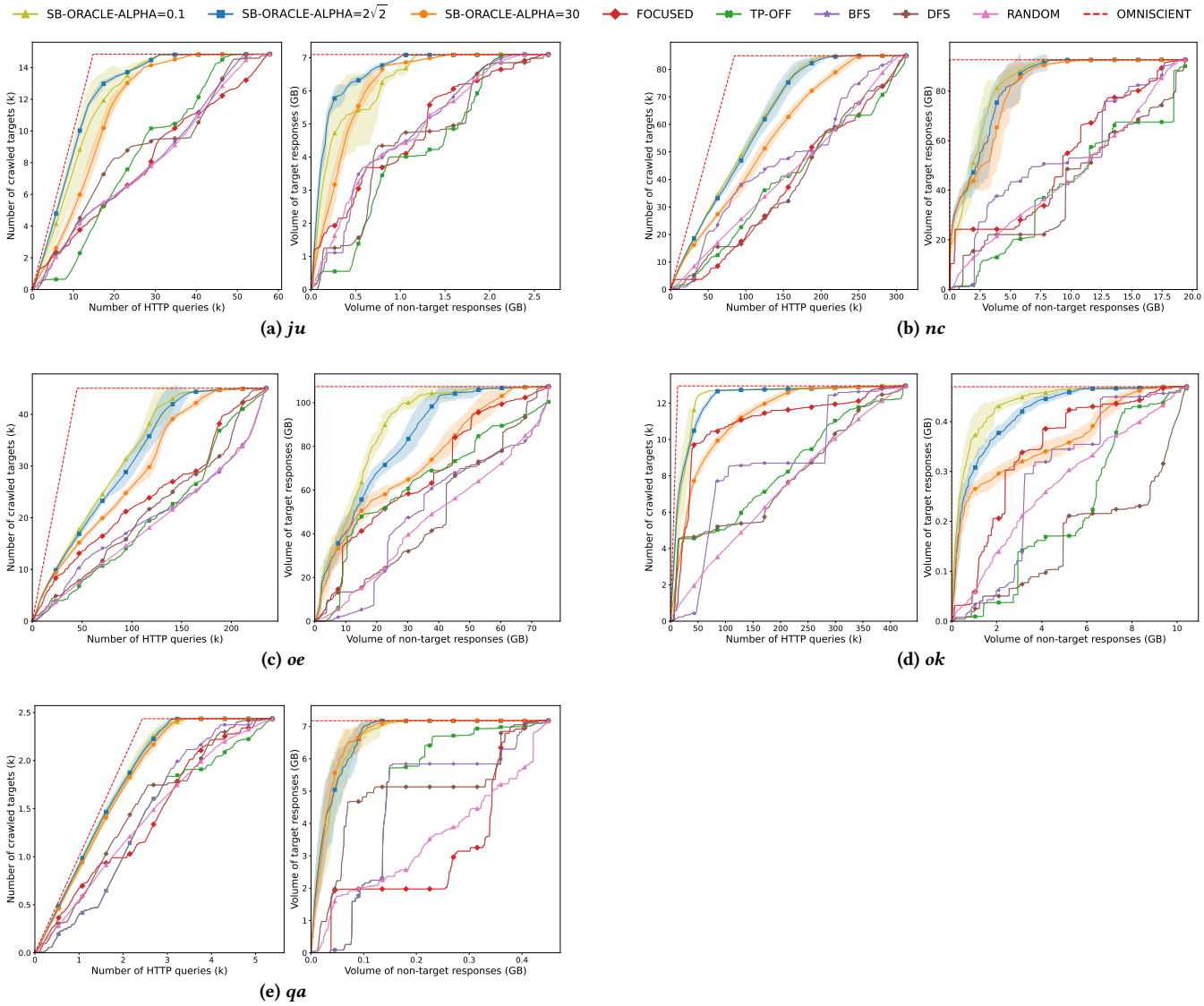


Figure 8: Crawler performance for hyper-parameter study on exploration–exploitation coefficient α, for websites *be*, *cl*, *cn*, *ed*, *in*, and *is*



(a) *ju* (b) *nc*

(c) *oe* (d) *ok*

(e) *qa*

Figure 9: Crawler performance for hyper-parameter study on exploration–exploitation coefficient α, for websites *ju*, *nc*, *oe*, *ok*, and *qa*

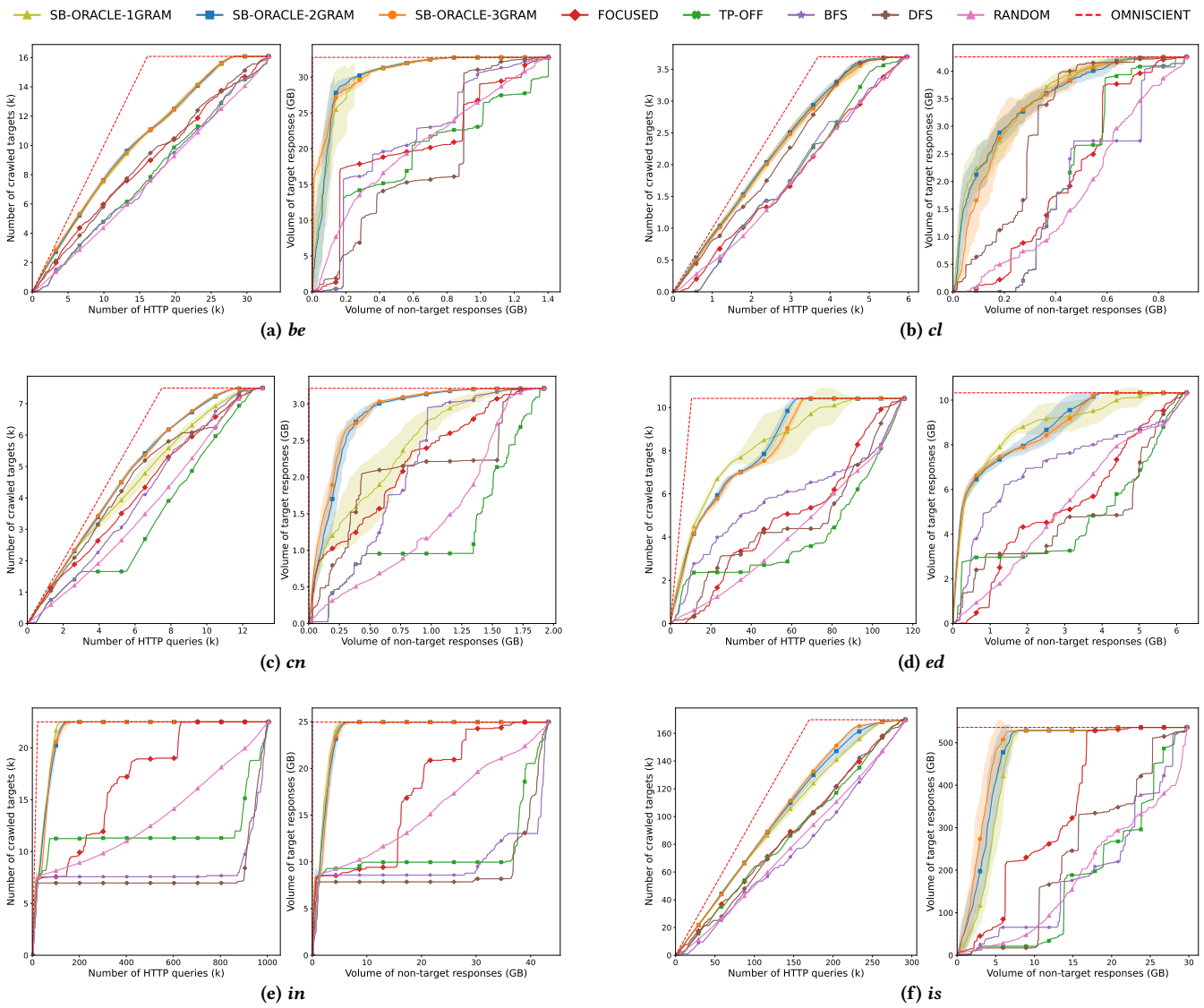
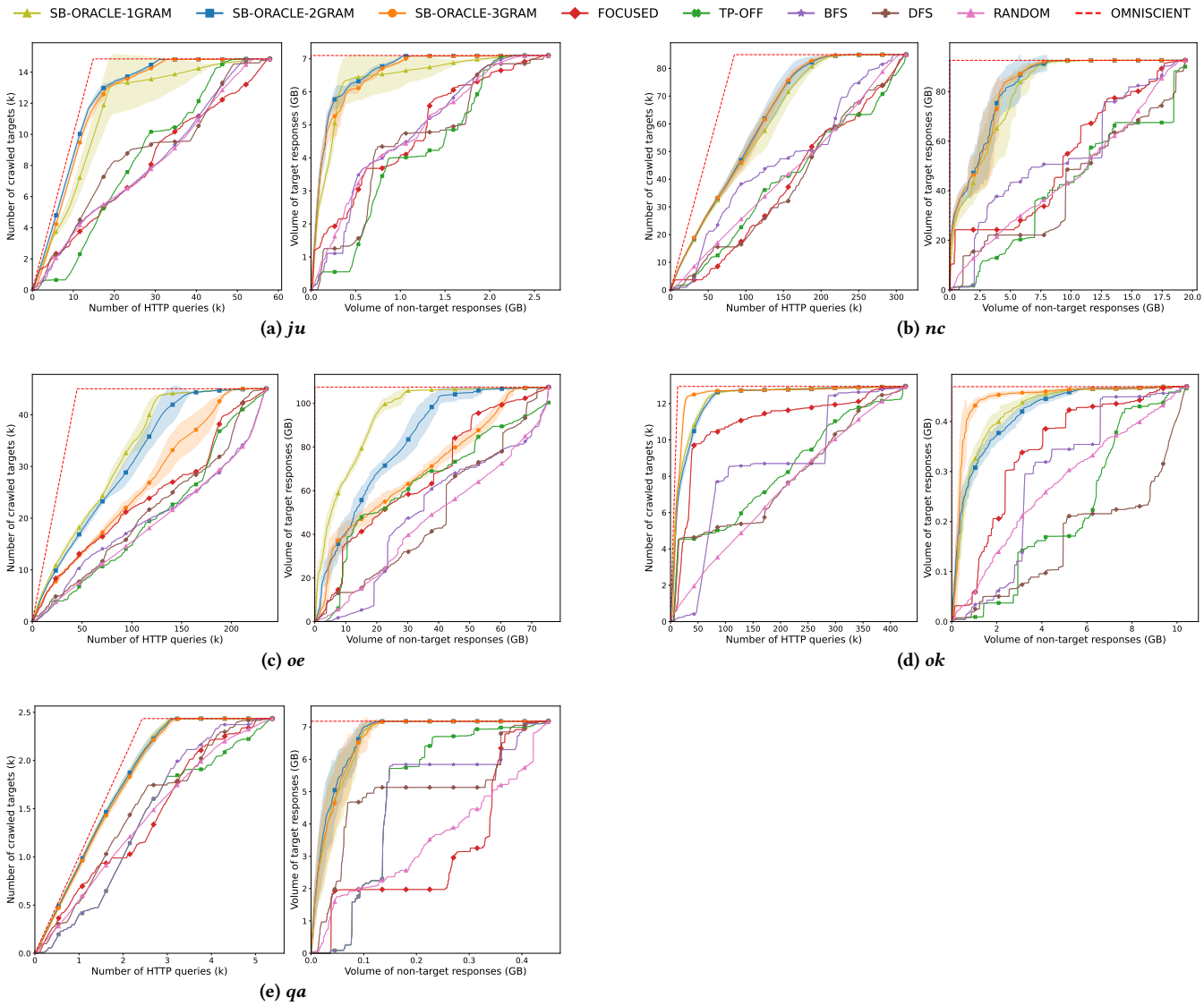


Figure 10: Crawler performance for impact study of the choice of n in n-grams used in tag path vector representation, for websites *be*, *cl*, *cn*, *ed*, *in*, and *is*



(a) *ju* (b) *nc*

(c) *oe* (d) *ok*

(e) *qa*

Figure 11: Crawler performance for impact study of the choice of n in n-grams used in tag path vector representation, for websites *ju*, *nc*, *oe*, *ok*, and *qa*

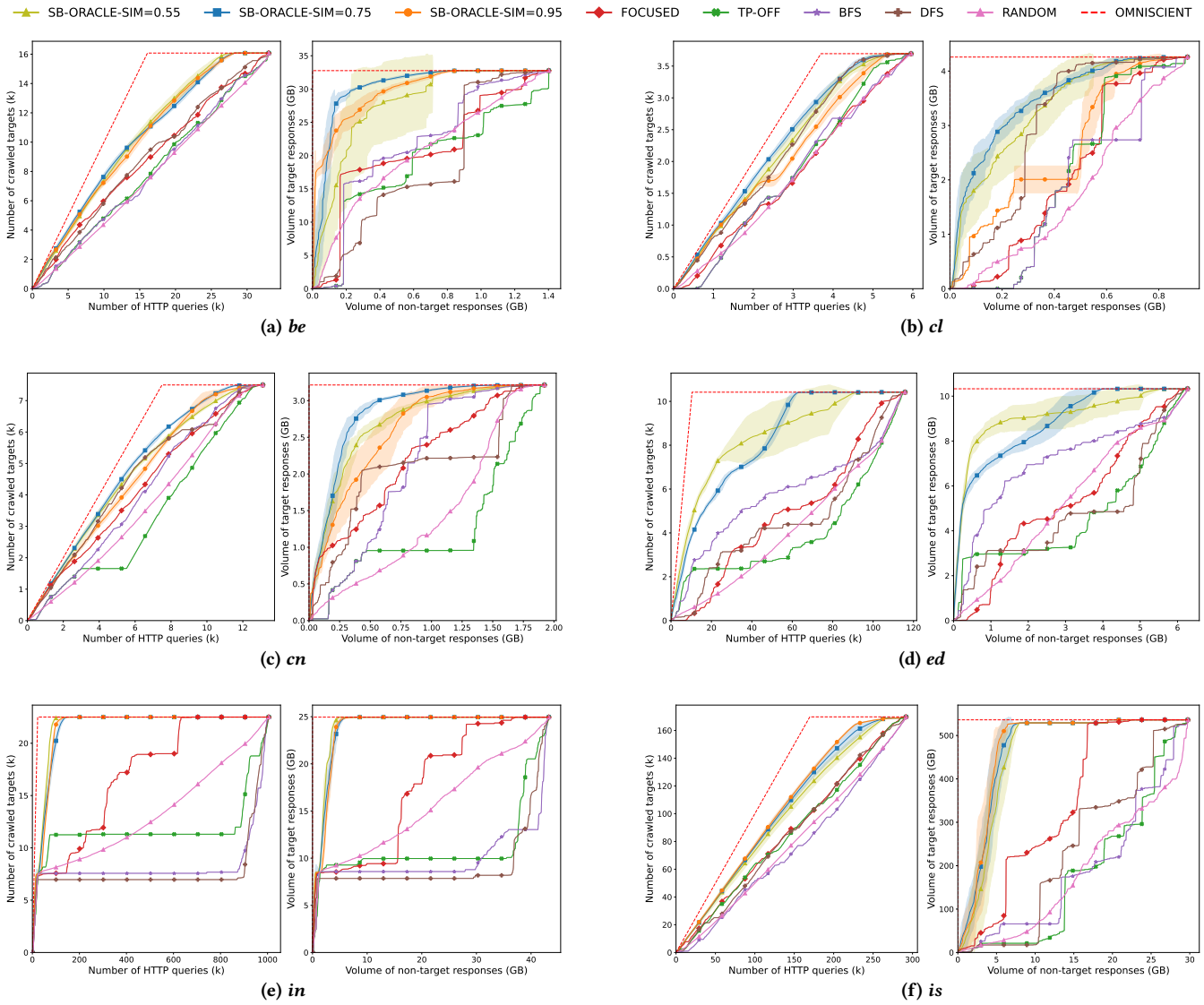
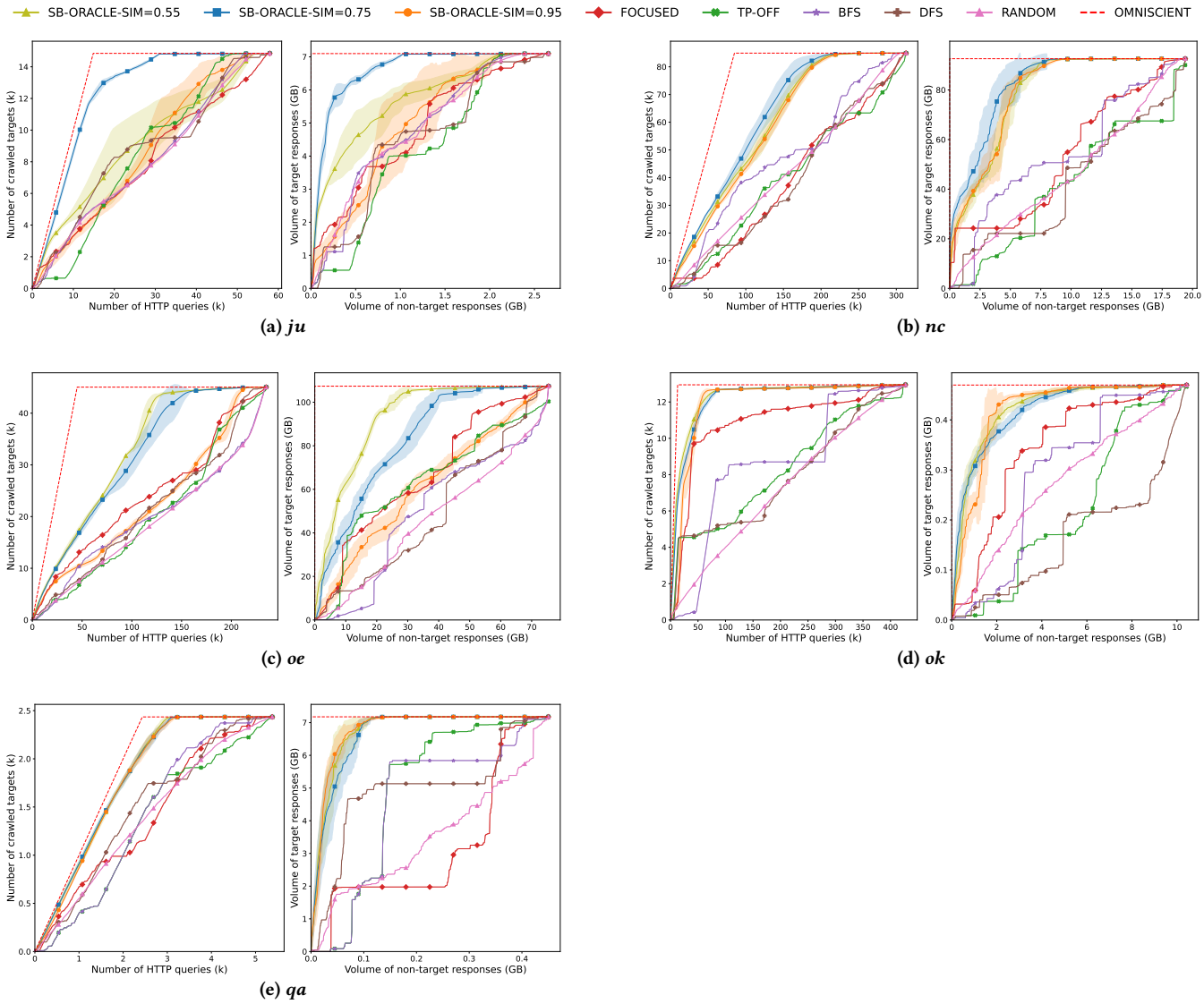


Figure 12: Crawler performance for impact study on similarity threshold θ, *cl*, *cn*, *ed*, *in*, and *is*



(a) *ju* (b) *nc*

(c) *oe* (d) *ok*

(e) *qa*

Figure 13: Crawler performance for impact study on similarity threshold θ, for websites *ju*, *nc*, *oe*, *ok*, and *qa*

References for the Appendix

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.
[WW16] Dimitri Watel and Marc-Antoine Weisser. A practical greedy approximation for the directed steiner tree problem. *J. Comb. Optim.*, 32(4):1327–1370, 2016.