Querying Web-Accessible Life Science Sources: Which paths to choose?

Jens Bleiholder, Felix Naumann
Humboldt-Universität zu Berlin
Berlin, Germany
{bleiho,naumann}@informatik.hu-berlin.de

Louiqa Raschid
University of Maryland
College Park, Maryland
louiqa@umiacs.umd.edu

María Esther Vidal
Universidad Simón Bolívar
Caracas, Venezuela
mvidal@ldc.usb.ve

1 Introduction

Web-accessible life sciences sources are characterized by a complex graph of overlapping sources, and multiple alternate links between sources. A (navigational) query may be answered by traversing multiple alternate paths between a start source and a target source. Each of these paths may have dissimilar *benefit*, e.g., the cardinality of result objects that are reached in the target source. Paths may also have dissimilar costs of evaluation, i.e., the execution cost of a query evaluation plan for a path. Finally, since the result objects of alternate paths may overlap, the combined benefit of two paths are not independent.

In this context, we present two problems. The first problem is to determine the K-best paths or *Relevant Paths* with low cost and high benefit. The second problem is to choose a good combination of top-k (possibly overlapping) paths. While the first problem regards paths individually and finds the best ones among a vast number of paths, the second problem assesses the integrated result of a set of paths. Further, we discuss the interrelation between the two problems and motivate the importance of a practical solution.

2 Motivation

Web-accessible life sciences sources are characterized by a complex graph of overlapping sources, and multiple alternate links between sources. A navigational query can be answered by a choice of alternate paths between a start source and a target source. Consider the query *Return all citations of* PubMed *that are linked to an* Omim *entry that is related to some disease or condition.* A scientist may choose the Omim source, which contains information related to human genetic diseases, as a starting point for her exploration and wish to eventually retrieve citations from the PubMed source. Starting with a keyword search on a certain disease, she can explore direct links between genes in Omim and citations in PubMed, via the Entrez portal http://www.ncbi.nlm.nih.gov/Entrez/. She can also traverse paths that are implemented using additional intermediate sources, e.g., NCBI Protein and NCBI Nucleotide. Figure 1 illustrates a source graph for four data sources.

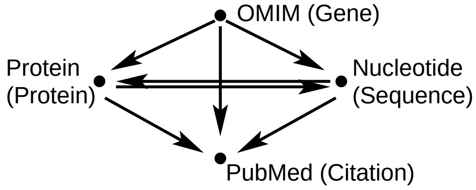

Figure 1: A source graph for Omim and NCBI data sources (and corresponding scientific entities)

While the figure presents a small number of sources and links, the number of public life science sources are in the thousands and increasing. A single query or experiment protocol may be interested in exploring tens of protein data sources, and following all the links from these sources. Further, there may be multiple semantics or meanings associated with the set of links between the objects of any two sources. Thus, a physical link between sources may implement multiple logical links. Consequently, the problem of enumerating all paths that satisfy a query and choosing paths becomes a challenge.

In the example of Figure 1 there are five paths (without loops) starting from Omim and terminating in PubMed. These paths are shown in Figure 2. The value (in bold) associated with each path, represents the cardinality of *distinct* PubMed objects that were retrieved along each path, starting from the same set of Omim objects; details of the experiment protocol to obtain these values are in [LMNR04]. As can be seen, there is a significant variation of the number of PubMed objects along each path. Object cardinality is a possible metric of the *benefit* associated with a path.

Each path is also associated with a *cost*, i.e., the cost of a query evaluation plan for the path. A plan for some path would be similar to a join query plan. Traversing a link can be implemented as a navigational join or a hash join. For example, for the link from Omim to PubMed, one could obtain all relevant objects from Omim and follow links to objects in PubMed. Alternately, one could download objects (and their links) from Omim and PubMed and perform the join locally. We note that there are often reverse links, e.g., PubMed to Omim; these links may not be symmetric and we do not consider reverse links here.

(P1)	Omim → PubMed	**7031**
(P2)	Omim → Nucleotide→ PubMed	**1736**
(P3)	Omim → Protein→ PubMed	**3275**
(P4)	Omim → Nucleotide→ Protein→ PubMed	**1753**
(P5)	Omim → Protein→ Nucleotide→ PubMed	**1570**

Figure 2: Five paths from Omim to PubMed

The example query from Omim to PubMed can be expressed as a simple regular expression; details of expressing such queries are in [LRV04]. Evaluating this query may involve first generating all paths that satisfy the query. Instead, one may wish to generate only the K-best paths or *Relevant Paths*, with low cost and high benefit, that satisfy the query. Finally, one may wish to choose the best path or some top-k combination of paths. The selected paths, (one, many, or all), must be evaluated to produce result objects.

3 Problem One: Multi-Criteria Optimization to Generate K-best Relevant Paths

In [LRV04], we presented an algorithm *ESearch*, based on a Deterministic Finite Automaton (DFA), to exhaustively enumerate all paths in a graph that satisfy some regular expression query representing a navigational query. The challenge is to efficiently find the K-best *Relevant Paths* with least cost and highest benefit.

Multi-objective optimization for database queries has been previously studied [DH02, GGK$^+$03, Nau02, PY01, SAL$^+$96, YNGM00]. The trade-off of execution cost versus delay in producing the results was studied in [SAL$^+$96] in the context of the Mariposa wide area DBMS. More recently, the trade-off of execution cost versus coverage was studied in [DH02, Nau02, YNGM00] in the context of Internet sources with overlapping coverage. [GGK$^+$03] studied the trade-off between the accuracy of results versus the index space needed to provide approximate answers to queries. Our problem is similar to [Nau02, SAL$^+$96, YNGM00], where the authors developed heuristics to rank sources; here we wish to develop heuristics to generate good paths.

Given a set of paths, each characterized by a benefit and a cost, multi-objective optimization will try to find solutions that maximize benefit and minimize cost. A *dominant* path has the property that no other path dominates it for both benefit and cost. The set of all such dominant paths is a *Pareto surface or curve* for the paths. There has been much work on computing such curves. While there may be exponentially many paths on the Pareto curve, an *approximate Pareto curve* can often be constructed in polynomial time [PY01].

Our first problem is to efficiently generate some K-best *Relevant Paths* that is Pareto dominant. To define Pareto dominance of a set of paths, $p_1, \ldots, p_n$, we consider the *Maximum Cost* of this set as the maximum *cost* for evaluating plans for each of the paths in the set. We also consider the *Maximum Benefit* as the *sum* of the result object cardinality *benefit* for the set of paths. We will consider overlap of paths in the next section. A Pareto dominant set of paths is one where there is no other set of paths that has *both* higher Maximum Benefit and lower Maximum Cost. In other words, we are interested in minimizing the (maximum) cost and maximizing the (sum of the) benefit of the set of paths of the K-best *Relevant Paths.* We note that we could choose different definitions of the Pareto dominant set.

In [VRM04], we present a heuristic solution that uses some *local utility functions* to rank sources that may produce good paths. Our results showed that local utility functions may lead to poor sub-paths, which do not lead to good paths. We are exploring good heuristics and utility functions that rank sources, links and sub-paths, rather than only ranking sources, so as to improve the solutions obtained.

4 Problem Two: Choosing a top-k Combination of Paths

Just as we associate a *benefit* with a path, we can also associate a benefit with (the results of) a query. Suppose we wish to maximize the benefit or the cardinality of PubMed objects reached by some query from

	P1	P2	P3	P4	P5
	7,031	1,736	3,275	1,753	1,570
P1		661	941	580	557
P2			1,531	1,589	1,513
P3				1,510	1,511
P4					1,501

Table 1: Overlap of PubMed entries retrieved by each path

Omim to PubMed. Then, we can improve the benefit of the result by not just evaluating a single path. Instead, we can choose a combination of paths. However, the target objects reached by all the paths from Omim to PubMed may not be disjoint. Thus, the benefit of following two overlapping paths does not necessarily equal the sum of the benefits of the two paths.

Recall the five paths from Omim to PubMed in Figure 2, each of which had a different benefit. Table 1 illustrates the *overlap* of distinct PubMed objects obtained along a pair of paths. This overlap represents a reduction of the benefit when evaluating both paths.

Figure 3 visualizes the five overlapping paths with the path cardinality and overlap given in Table 2. We note that for simplicity of the example, we do not use the exact values from Table 1. The surrounding rectangle represents the set of all objects (publications) contained in the target source (PubMed source), and the ellipses mark the objects (publications) reached by a particular path. Publications reached by more than one path are in overlapping regions of two or more ellipses.

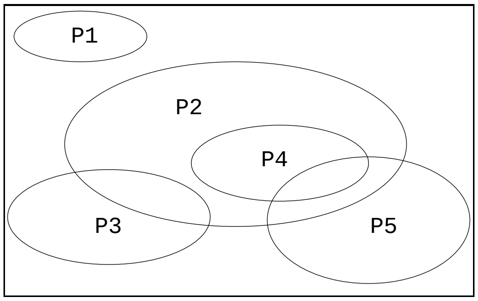

Figure 3: Some overlapping paths reaching objects in a target source

Path	size	Combination	overlap
P1	100	P2 & P3	100
P2	1,000	P2 & P4	200
P3	300	P2 & P5	200
P4	200	P4 & P5	100
P5	500	P2 & P4 & P5	100

Table 2: Path Cardinality and Non-Zero Overlap

We now consider the *overlap-adjusted benefit* or the benefit of the path subtracted by the overlap with other, already evaluated paths. For instance in this example, path P4 has cardinality 200 and is completely subsumed by path P2. So the overlap-adjusted benefit of visiting P4, after visiting P2 is zero. Further, the overlap between pairs of paths may not be independent. So the overlap of P2 and P5 is 200, the overlap of P4 and P5 is 100, while the overlap of P2, P4, and P5 is also 100.

Determining the exact overlap of two paths may be a problem by itself. As long as the number of sources and their objects is relatively small, one can assume perfect information and be able to correctly compute and store all overlap statistics by just following all existing links. However, as the number of sources and objects increases this is not feasible anymore and one has to find a way to approximate overlap. Sample queries or simply sampling the sources is one way to achieve this task and it has been applied to the domain of bibliographic citations [NKH03].

When answering a query in the spirit of the previous section, we now consider a combination of paths. Each path has an individual cost associated with it. The set of paths will also have an overlap-adjusted benefit. Minimizing the sum of costs and maximizing overlap-adjusted benefit turns the problem of choosing some combination of paths into an optimization problem. It is different from the multi-criteria optimization problem, since in that problem, we do not consider the overlap-adjusted benefit. Thus, given a total cost limit (budget), the second problem is to identify the top-k combination of paths, so that the total sum of costs does not exceed the budget, while maximizing the overlap-adjusted benefit. To determine a solution to this problem, the actual sequence of visited paths is not needed. The important information is the overlap of the paths, the overlap-adjusted benefit, and the costs of the paths.

We can model the problem as follows: each path is a *virtual* source that covers a certain amount of objects of the target source. The benefit is the cardinality of objects covered, and the costs are not uniform. The problem of choosing a combination of top-k paths is essentially the same as choosing k overlapping sources with a maximum overlap-adjusted benefit. This problem is known in the literature as the *budgeted maximum coverage problem* [KMN99], which is NP-hard. Algorithms that produce exact solutions are therefore feasible only for small instances. As the size of the graph increases, and thus the number of sources, the number of physical links between sources, and the number of logical links associated with some physical link increase, the number of possible paths that must be explored may reach the thousands. Exact algorithms will no longer be tractable and there is a need to improve on them, and to develop new methods or new heuristics, in order to solve the problem efficiently.

5 Combining Problems One and Two: Future Work

When there is no tractable exact solution to the problem of choosing a top-k combination of paths, there are, in principle, two possibilities. First, one can find a heuristic that reduces the complexity of the exact algorithm. Second, one can reduce the complexity of the exact solution by limiting the top-k search to a small set of promising candidates, instead of using all possible paths. With this latter approach, any given exact solution to the top-k problem is feasible. However, the quality of the result will depend heavily on the quality of the promising candidates. Perfect overlap information for this small set of candidates can be determined, or at least estimated. If this is done by instantaneous sampling an additional cost should be taken into account.

Recall, that given the large number of paths that may satisfy a query, the first problem, described in Section 3 was a multi-criteria optimization problem to choose the *K-Best Relevant Paths*. We can now apply this first problem to produce the input candidate paths to the second problem of finding the top-k paths. Thus, the quality of the top-k paths solution will make use of the K-Best Relevant Paths solution. We note that the challenge for K-Best Relevant Paths was to identify good utility functions, so as to choose good sources, links and sub-paths. Combining the two problems as described, we plan to investigate good utility functions that also identify good combinations of paths with high overlap-adjusted benefit.

References

[DH02] A. Doan and A. Halevy. Efficiently ordering query plans for data integration. In *Proceedings of the International Conference on Data Engineering (ICDE)*, San Jose, CA, 2002.

[GGK+03] S. Guha, D. Gunopulos, N. Koudas, D. Srivastava, and M. Vlachos. Efficient approximation of optimization queries under parametric aggregation constraints. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 778–789, Berlin, Germany, 2003.

[KMN99] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70:39–45, April 1999.

[LMNR04] Z. Lacroix, H. Murthy, F. Naumann, and L. Raschid. Links and paths through life sciences data sources. In *Proceedings of the International Workshop on Data Integration for the Life Sciences (DILS)*, pages 203–211, Leipzig, Germany, 2004.

[LRV04] Z. Lacroix, L. Raschid, and M.E. Vidal. Efficient techniques to explore and rank paths in life science data sources. In *Proceedings of the International Workshop on Data Integration for the Life Sciences (DILS)*, pages 187–202, Leipzig, Germany, 2004.

[Nau02] F. Naumann. *Quality-driven Query Answering for Integrated Information Systems*, volume 2261 of *Lecture Notes on Computer Science (LNCS)*. Springer Verlag, Heidelberg, 2002.

[NKH03] Zaiqing Nie, Subbarao Kambhampati, and Thomas Hernandez. Bibfinder/statminer: Effectively mining and using coverage and overlap statistics in data integration. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, pages 1097–1100, Berlin, Germany, 2003.

[PY01] C.H. Papadimitriou and M. Yannakakis. Multiobjective query optimization. In *Proceedings of the Symposium on Principles of Database Systems (PODS)*, Santa Barbara, CA, 2001.

[SAL+96] M. Stonebraker, P. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu. Mariposa: a wide-area distributed database system. *VLDB Journal*, 5(1):048–063, 1996.

[VRM04] M.E. Vidal, L. Raschid, and J. Mestre. Challenges in selecting paths for navigational queries: Trade-off of benefit of path versus cost of plan. In *Proceedings of the ACM SIGMOD Workshop on The Web and Databases (WebDB)*, Paris, France, 2004.

[YNGM00] R. Yerneni, F. Naumann, and H. Garcia-Molina. Maximizing coverage of mediated web queries. *Stanford University Technical Report, Computer Science Department*, 2000.