Technical Report: LP Randomized Rounding for Maximum Coverage Problem and Minimum Set Cover with Threshold Problem

Samir Khuller, Louiqa Raschid, Yao Wu
University of Maryland
{samir|yaowu}@cs.umd.edu, louiqa@umiacs.umd.edu

Abstract

There are abundance of web accessible life science sources. Traversal of a particular path can answer a navigational query, returning a target object set (TOS). The cardinality of TOS is considered as the benefit of a path and there is some cost function associated with each path. It is common that multiple alternate paths satisfy the query and we are not allowed to pick all these paths to answer the query, since there could be exponential number of paths in a graph. We are interested in selecting a subset of these paths.

We present two problems in this context. The first problem is to select a subset of paths of maximum benefit within a cost budget. This is known as *Budgeted Maximum Coverage Problem* in the literature. The second problem is to select a subset of paths of minimum cost with a threshold benefit guarantee. This is the *Minimum Set Cover with Threshold Problem.* We develop randomized approximation algorithms based on LP rounding and conduct experiments.

1 Introduction

The last few years have seen an explosion in the number of public life science data sources, as well as the volume of data entries about scientific entities, such as genes, proteins, sequences, molecules, etc., that are characterized in these sources. Consequently, biologists spend a considerable amount of time navigating through the contents of these sources to obtain useful information.

Life sciences sources, and the navigational queries that are of interest to scientists, pose some unique challenges. First, information about a scientific entity, e.g., a protein, may be available in a large number of autonomous sources, each of which may provide different characterizations of the protein. While it is clear that the contents of these sources *overlap*, these sources are not replicas since they do not always cover the same instances of proteins, and they do not characterize the instances in an identical manner. Second, the *links* between scientific entities (links between data objects) in the different sources are unique in this domain in that they capture significant knowledge about the relationship and interactions between these entities. These links are uncovered in the process of navigating between sources. They may change over time as new knowledge is discovered. Third, limited by bandwidth or connection cost, since most of these sources are autonomous and distributed, it is usually not feasible to collect all the paths satisfying the query.

We consider a set of sources, and we further assume that the data objects in any of these sources have links to data objects in one or more of the other sources. We further assume that a (simple) navigational query identifies an origin class, e.g., protein and possibly a (set of) origin sources that are of interest, e.g., UniProt. The query also identifies a target class of interest, e.g., publications, as well as an optional list of intermediate sources. Then, answering the query first involves exploring the data sources and classes, and the links between data sources. Our goal is to find paths at the logical level (among classes) and paths at the physical level (among sources implementing these classes). While we note that the query language can be extended to other query types, for our study we use a simple query.

Each path is associated with a *benefit*, namely the number of distinct objects reached in the target object set (TOS) in the target class. Each path is also associated with a *cost* of evaluating the query on the sources to compute the TOS. Given the overlap between sources and the highly interconnected nature of the object graph, each m-way combination of TOSs of paths is also associated with a *TOS overlap*. This overlap represents same objects reached in the TOS using different paths, and reduces the combined benefit of this path combination.

We present dual problems in this context of selecting the best set of paths. The first problem is to select a set of paths that satisfy a constraint on the evaluation cost while maximizing the benefit or the number of distinct objects in the TOS of these paths. This problem maps to the budgeted maximum coverage (BMC) problem [8]. We expect that in many cases, a user is more interested in reaching some desired threshold or number of objects and may not set a constraint on the budget. To explore this situation, we consider the dual problem, which selects a set of paths that satisfies a threshold of the TOS benefit with minimal evaluation cost. The dual can be mapped to the maximal set cover with a threshold (MSCT).

The problems we address apply to many other scenarios. Consider a general problem - find a best set of paths to the data sources - and a simpler subproblem - find the best set of sources, ignoring that there might be multiple heterogeneous paths to reach these sources. This subproblem arises in many data integration situations, namely whenever (i) the integrated system has access to multiple sources that overlap in the data they store, (ii) it is not necessarily required to retrieve *all* answers to a query (some are enough), and (iii) some per-source cost is incurred to find and retrieve answers. Applications include metasearch engines and search engines for intranets, stock information systems (queries cost money), shopping agents that integrate multiple online shops, and digital libraries. For each of these systems it is worthwhile to access only some data sources and still present satisfying results to a user or application.

The problem is also interesting in a scenario that has the restriction that some sources cannot be reached directly but can be reached only via other sources. Most prominently, P2P file sharing systems try solve this problem. Their typical solution is to broadcast a query to all peers (within a certain scope, such as 7 hops), but more sophisticated methods using specialized indices have been proposed. In the intersection of P2P systems and data integration systems lie so called peer data management systems (PDMS), for which many application scenarios have been proposed [12].

The outline of this report is as follows: Sec. 2 presents LP rounding algorihtm for the first problem and shows optimality results. Sec. 3 presents LP rounding algorihtm for the second problem and shows optimality results. Finally Sec. 4 concludes.

2 BMC problem

This section solves the problem using standard LP relaxation and rounding approach. We are able to show that the expected cost does not exceed budget and the expected cost is within a factor of optimal solution.

2.1 Integer Programming and Linear Programming Formulation

Let $\mathcal{S}$ be a family of sets(paths), $\mathcal{S} = \{S_1, S_2, \ldots S_m\}$; Z be the set of elements(objects), $Z = \{z_1, z_2, \ldots z_n\}$, B be the budget allowed to choose a subset of paths $\mathcal{S}^{‘}$; $c(S_i)$ be the cost of picking set S_i; w_j be the benefit of covering element z_j. In our problem, we consider a uniform benefit for all objects; that is, $w_j = 1$ for each object z_j. We can set integer variables $x_i = 1$ iff set S_i is picked in $\mathcal{S}^{‘}$ and $y_j = 1$ iff z_j is covered. The IP formulation is as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} y_j \cdot w_j \\ \text{subject to} & \end{array}$$

$$\begin{array}{ccc} \sum_{i=1}^{m} c(S_i) \cdot x_i & \leq & B \\ y_j & \leq & \sum_{\{i|z_j \in S_i\}} x_i \quad for \quad all\ j \\ x_i \in \{0,1\} & & \text{for all } i \\ y_j \in \{0,1\} & & \text{for all } j \end{array}$$

Although IP gives optimal solution to the problem, it is impractical to compute exact solution as IP problem is NP-complete.

By relaxing the constraints that x_i and y_j must be integers, we have the following Linear Program (LP) formulation. Note that only the two last constraints of the IP formulation have been modified as follows:

$$\begin{array}{ccc} x_i & \leq & 1 \\ y_j & \leq & 1 \end{array}$$

2.2 Algorithm and Analysis

We will show that using a standard technique such as Randomized rounding [10], we can derive a randomized algorithm whose expected cost is at most B and at the same time the expected weight of the covered elements is at least $(1 - \frac{1}{e})$ times the LP benefit. Since the LP benefit is an upper bound on the optimal integral solution, this would be an alternate way of deriving the bound developed earlier using a greedy algorithm combined with an enumeration approach [8].

We solve the Linear Program developed earlier (using CPLEX) thus obtaining an optimal fractional solution, called (x^*, y^*). We now obtain a collection of sets $\mathcal{S}'$ such that $Pr[\text{ Set } S_i \text{ is chosen in } \mathcal{S}'] = x_i^*$.

Algorithm BMC_LP
Solve LP relaxation, get fractional solution (x^*, y^*)
Rounding x^* values to pick a subset of paths

Lemma 2.1 *The expected cost of $\mathcal{S}'$ is at most B.*

Proof. Let us compute the expected cost of $\mathcal{S}'$ as follows.

$$E[c(\mathcal{S}')] = \sum_{i=1}^{m} c(S_i) \cdot x_i^* \leq B$$

Thus the expected cost of the rounded solution is at most B. ∎

Lemma 2.2 *The total weight of the covered elements is at least $(1-\frac{1}{e})w(OPT)$ where $w(OPT)$ is the weight of an optimal set of covered elements by choosing a collection of sets of cost at most B.*

Proof. We prove this as follows. We first consider the probability that an element z_j is covered. We will abuse notation and let $\mathcal{S}'$ refer to the collection of sets chosen by the algorithm as well as the collection of covered elements.

$$Pr[z_j \in \mathcal{S}'] = 1 - Pr[z_j \notin \mathcal{S}']$$

Note that $z_j \notin \mathcal{S}'$ if and only if each set that z_j belongs to is not included in $\mathcal{S}'$.

$$Pr[z_j \in \mathcal{S}'] = 1 - \prod_{z_j \in S_i} (1 - x_i^*)$$

We know that for any real number p, $1 - p \leq e^{-p}$, where $0 \leq p \leq 1$. This can be shown by elementary calculus. Define a function $f(p) = e^{-p} - 1 + p$. We have $f(0) = 1$ and $f'(p) = 1 - e^{-p}$. $f'(p) > 0$ for all $p > 0$. $f(p)$ is increasing when $p > 0$; therefore, $e^{-p} - 1 + p > 0$ when $p > 0$. We have $1 - p \leq e^{-p}$. We replace $(1 - x_i^*)$ by $e^{-x_i^*}$ in the formula, get:

$$Pr[z_j \in \mathcal{S}'] \geq 1 - \prod_{z_j \in S_i} e^{-x_i^*}$$

Rewirte the formula, we have:

$$Pr[z_j \in \mathcal{S}'] \geq 1 - e^{\sum_{z_j \in S_i} -x_i^*}$$

According to the LP constraints, we have $\sum_{z_j \in S_i} -x_i^* \geq y_j^*$,

$$Pr[z_j \in \mathcal{S}'] \geq 1 - e^{-y_j^*}$$

The expected benefit of $\mathcal{S}'$ will be:

$$E(w(\mathcal{S}')) = \sum_{j=1}^{n} w_j Pr[z_j \in \mathcal{S}']$$

With $Pr[z_j \in \mathcal{S}'] \geq 1 - e^{-y_j^*}$, the expected benefit is guaranteed to be "good":

$$E(w(\mathcal{S}')) \geq \sum_{j=1}^{n} w_j (1 - e^{-y_j^*})$$

If we consider the ratio of the expected weight of $\mathcal{S}'$ to the LP cost, $\sum_{j=1}^{n} w_j y_j^*$, which is an upper bound of optimal value. We get

$$\frac{E(w(\mathcal{S}'))}{\sum_{j=1}^{n} w_j y_j} \geq \frac{\sum_{j=1}^{n} w_j (1 - e^{-y_j^*})}{\sum_{j=1}^{n} w_j y_j}$$

Next we show that $1 - e^{-q} \geq (1 - \frac{1}{e}) \cdot q$ for $0 \leq q \leq 1$. Let us consider function $f(q) = 1 - e^{-q}$ where $0 \leq q \leq 1$. This is a concave function in interval $[0, 1]$. The function $f(q)$ lies above the straight line joining points $(0, f(0))$ and $(1, f(1))$, that is $(0, 0)$ and $(1, 1 - \frac{1}{e})$. Therefore, $1 - e^{-q} \geq (1 - \frac{1}{e})q$.

Since $(1 - e^{-y_j^*}) \geq (1 - 1/e)y_j^*$ we get that the ratio is at least $(1 - \frac{1}{e})$. ■

Immediately from lemma 2.1 and 2.2, we have the following:

Theorem 2.3 *Algorithm* BMC_LP *is a randomized* $(1 - \frac{1}{e})$ *approximation algorithm with expected cost* B*.*

3 MSCT Problem

However, users are usually more concerned about quality of query answers instead of the cost of answering query. The problem becomes minimizing the cost of query plan while guarantee some quality value. More formaly, we are interested in the problem that selecting a subset of paths with a threshold, with minimum cost.

3.1 IP and LP Formulation

The notation is the same as in Section 2, except that we want to choose a subset of paths that meet the threshold T while minimizing the cost. The IP formulation is as follows:

$$\text{minimize} \quad \sum_{i=1}^{m} c(S_i) \cdot x_i$$

subject to

$$\begin{array}{ccll} \sum_{j=1}^{n} y_j \cdot w_j & \geq & T & \\ y_j & \leq & \sum_{\{i \mid z_j \in S_i\}} x_i & \text{for all } j \\ x_i \in \{0,1\} & & \text{for all } i & \\ y_j \in \{0,1\} & & \text{for all } j & \end{array}$$

By letting x_i and y_j be reals in $[0, 1]$ and ignoring one side of the constraints, we replace the last two constraints in IP to obtain the LP formulation:

$$\begin{array}{ccc} x_i & \geq & 0 \\ y_j & \geq & 0 \end{array}$$

3.2 Algorithm and Analysis

We use the similiar randomized rounding approach as before.

Let (x^*, y^*) be the fractional solution obtained by CPLEX. We choose a collection of sets $\mathcal{S}'$ such that $Pr[\text{ Set } S_i \text{ is chosen in } \mathcal{S}'] = \min(1, \alpha x_i^*)$, where $\alpha \geq 1$ is a boosting factor to ensure that we reach the threshold. This algorithm produces solutions with expected benefit at least $(1 - \frac{1}{e^\alpha}) \cdot T$ and expected cost at most $\alpha \cdot$ OPT.

Algorithm MSCT_LP
Solve LP relaxation, get fractional solution (x^*, y^*)
Rounding x^* values such that
$Pr[\text{ Set } S_i \text{ is chosen in } \mathcal{S}'] = \min(1, \alpha x_i^*)$

Lemma 3.1 *The expected cost of $\mathcal{S}'$ is at most α OPT.*

Proof. Let us compute the expected cost of $\mathcal{S}'$ as follows.

$$\begin{array}{rcl} E[c(\mathcal{S}')] & = & \sum_{i=1}^{m} c(S_i) \cdot \min(1, \alpha \cdot x_i^*) \\ & \leq & \sum_{i=1}^{m} c(S_i) \cdot \alpha \cdot x_i^* \\ & \leq & \alpha \sum_{i=1}^{m} c(S_i) \cdot x_i^* \\ & \leq & \alpha \mathrm{OPT} \end{array}$$

Thus the expected cost of the rounded solution is at most α OPT. ∎

Lemma 3.2 *The total weight of the covered elements is at least* $(1 - \frac{1}{e^\alpha}) \cdot T$.

Proof. Again we first consider the probability that an element z_j is covered.

$$\begin{array}{rcl} Pr[z_j \in \mathcal{S}'] & = & 1 - \prod_{z_j \in S_i} (1 - x_i^*) \\ & = & 1 - \prod_{z_j \in S_i} (1 - \alpha x_i^*) \\ & \geq & 1 - \prod_{z_j \in S_i} e^{-\alpha x_i^*} \\ & \geq & 1 - e^{-\alpha \sum_{z_j \in S_i} x_i^*} \\ & \geq & 1 - e^{-\alpha y_j^*} \end{array}$$

The expected benfit of $\mathcal{S}'$ is:

$$\begin{array}{rcl} E(w(\mathcal{S}')) & = & \sum_{j=1}^{n} w_j Pr[z_j \in \mathcal{S}'] \\ & \geq & \sum_{j=1}^{n} w_j (1 - e^{-\alpha y_j^*}) \end{array}$$

Therefore, we have:

$$\begin{array}{rcl} \frac{E(w(\mathcal{S}'))}{\sum_{j=1}^{n} w_j y_j} & \geq & \frac{\sum_{j=1}^{n} w_j (1 - e^{-\alpha y_j^*})}{\sum_{j=1}^{n} w_j y_j} \\ & \geq & (1 - e^{-\alpha}) T \end{array}$$

∎

Immediately from lemma 3.1 and 3.2, we have the following:

Theorem 3.3 *Algorithm* MSCT_LP *is a randomized α approximation algorithm with expected benefit at least* $(1 - \frac{1}{e^\alpha}) T$.

4 Conclusion

Originally motivated by the problem of finding good paths and sets of paths through NCBI life sciences sources we have generalized the problem to data integration in the presence of overlapping sources, which applies to many different kinds of information systems. We presented a pair bounded randomized approximation algorithms. To summarize, life sciences data sources are an excellent field to test new query models (paths through sources) and optimization problems (overlap-adjusted benefit), all the while solving problems that are relevant to biologists.

Possible future work is abundant. In a direct continuation of the work presented here we plan to expand on the types of queries to find out how our algorithms fare under different applications. Further strands of research are in the field of path query languages, efficient enumeration of all possible paths, and finally optimization techniques on the actual web-accessible NCBI sources rather than on large sampled sets stored in a local database.

References

[1] Jens Bleiholder, Felix Naumann, Louiqa Raschid, and Maria Esther Vidal. Querying web-accessible life science sources: Which paths to choose? In *Proceedings of VLBD Workshop on Information Integration on the Web*, 2004.

[2] Barbara Eckman, Kerry Deutsch, Marta Janer, Zoé Lacroix, and Louiqa Raschid. A query language to support scientific discovery. In *CSB '03: Proceedings of the IEEE Computer Society Conference on Bioinformatics*, page 388. IEEE Computer Society, 2003.

[3] T. Etzold and G. Verde. Using views for retrieving data from extremely heterogeneous databanks. *Pacific Symposium on Biocomputing*, 2:134–141, 1997.

[4] Thure Etzold and P. Argos. Srs - an indexing and retrieval tool for flat file data libraries. *Computer Applications in the Biosciences*, 9(1):49–57, 1993.

[5] Gerhard Goos. *Vorlesungen ber Informatik - Paralleles Rechnen und nicht-analytische Lsungsverfahren*, volume 4. Springer Verlag, Berlin, Germany, 1998.

[6] L. M. Haas, P. Kodali, J. E. Rice, P. M. Schwarz, and W. C. Swope. Integrating life sciences data-with a little garlic. In *BIBE '00: Proceed-*

ings of the 1st IEEE International Symposium on Bioinformatics and Biomedical Engineering, page 5. IEEE Computer Society, 2000.

[7] Graham Kemp, Chris Robertson, and Peter Gray. Efficient access to biological databases using corba. *CCP11 Newsletter*, 3.1, 1999.

[8] Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45, 1999.

[9] Zoé Lacroix, Kaushal Parekh, Louiqa Raschid, and Maria-Esther Vidal. Navigating through the biological maze. In *2004 IEEE Computational Systems Bioinformatics conference(BSB'04)*, pages 594–595, 2004.

[10] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

[11] Maria-Esther Vidal, Louiqa Raschid, and Julian Mestre. Challenges in selecting paths for navigational queries: trade-off of benefit of path versus cost of plan. In *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases*, pages 61–66. ACM Press, 2004.

[12] Alon Y. Halevy and Zachary G. Ives and Peter Mork and Igor Tatarinov Piazza: data management infrastructure for semantic web applications. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 556–567, 2003