Multi-Objective Query Optimization in Smartphone Social Networks

Andreas Konstantinidis, Demetrios Zeinalipour-Yazti, Panayiotis Andreou, George Samaras

Department of Computer Science, University of Cyprus,
P.O. Box 20537, 1678 Nicosia, Cyprus
{akonstan, dzeina, panic, cssamara}@cs.ucy.ac.cy

Abstract*— The bulk of social network applications for smartphones (e.g., Twitter, Facebook, Foursquare, etc.) currently rely on centralized or cloud-like architectures in order to carry out their data sharing and searching tasks. Unfortunately, the given model introduces both data-disclosure concerns (e.g., disclosing all captured media to a central entity) and performance concerns (e.g., consuming precious smartphone battery and bandwidth during content uploads). In this paper, we present a novel framework, coined SmartOpt, for searching objects (e.g., images, videos, etc.) captured by the users in a mobile social community. Our framework, is founded on an *in-situ* data storage model, where captured objects remain local on their owner's smartphones and searches then take place over a novel lookup structure we compute dynamically, coined the *Multi-Objective Query Routing Tree (MO-QRT)*. Our structure concurrently optimizes several conflicting objectives (i.e., it minimizes energy consumption, minimizes search delay and maximizes query recall), using a Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) that calculates a diverse set of high quality non-dominated solutions in a single run. We assess our ideas with mobility patterns derived by Microsoft's Geolife project and social patterns derived by DBLP. Our study reveals that SmartOpt can yield query recall rates of 95%, with one order of magnitude less time and two orders of magnitude less energy than its competitors.

I. Introduction

The widespread deployment of smartphone devices and the advent of social networks have brought a revolution in social-oriented applications and services for mobile phones. There is already a proliferation of innovative applications founded on the concept of a smartphone network[1]. One example is opportunistic and participatory sensing [4], [2], [3], where applications can task mobile nodes in a given region to provide information about their vicinity using their sensing capabilities. Another example is road traffic delay estimation [11] using WiFi beams collected by smartphone devices rather than invoking expensive GPS acquisition. On the social site, Google Latitude [8] enables users to track the places they and their social network have visited. The given service already reports over 3M enrolled users and over 1M active users, despite the controversial privacy concerns. Similarly, mobile social

[1]We define a Smartphone Network as "*a set of smartphone devices that communicate in an unobtrusive manner, without explicit user interactions, in order to realize a collaborative or social task.*"

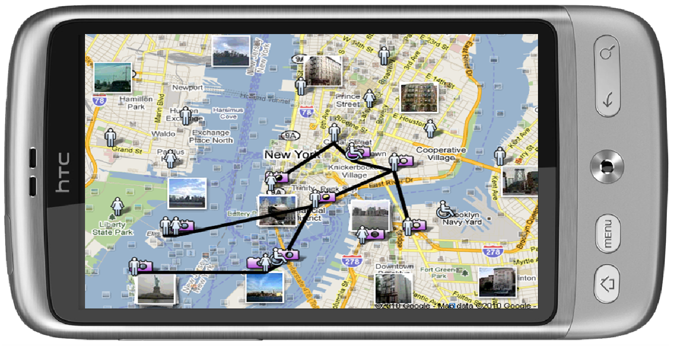


Fig. 1. A visual illustration of the *Multi-Objective Query Routing Tree (MO-QRT)* structure proposed in this work. Our SmartOpt Framework constructs MO-QRT structures optimized on several conflicting objectives (i.e., energy, time and recall). Our structure can be utilized for finding objects (e.g., images, videos, etc.) in a social neighborhood, without the necessity of having the objects disclosed to the social network provider.

networking applications like Foursquare, Gowalla and Loopt enjoy enormous success in the Smartphone community.

Currently, the bulk of social networking services, designed for smartphone communities, rely on centralized or cloud-like architectures. In particular, in order to enable content sharing and community search, the smartphone clients upload their captured objects (e.g., images uploaded to Twitter, video traces uploaded to Youtube, etc.) to a central entity that subsequently takes care of the content organization and dissemination tasks. Although certain types of objects, such as text-based micro-blogs, will behave reasonably well under this model, significant challenges arise for captured multimedia and sensor data (e.g., data captured by the camera, microphone, accelerometer, etc.) We claim that the centralization of these object types will be severely hampered in the future due to the following constraints:

i. **Data-Disclosure Constraints:** Continuously disclosing user-captured objects to a central entity might compromise user privacy in very serious ways[2].
ii. **Energy Constraints:** Smartphones have expensive communication mediums, thus by continuously transferring

[2]"Google Apologizes for Buzz Privacy", David Coursey, PC World Business Center (online), Feb. 15th, 2010.

massive amounts of data to a query processor, through WiFi/3G/4G connections, can both deplete the precious smartphone battery faster, increase query response times, but can also quickly degrade the network health[3].

In this paper, we present techniques to enable smartphone users keep their data *in-situ*, for data-disclosure and performance reasons, offering at the same time high performance search capabilities over other user's data in the social community. When a user invokes a search to find an object of interest, e.g., *"Pictures of street artists performing in Manhattan"* (see Figure 1), the user first downloads a *Query Routing Tree (QRT)* $\mathcal{X}$ from a SmartOpt server. The $\mathcal{X}$ structure resembles spanning tree structures constructed during searches in unstructured Peer-to-Peer (P2P) systems [13], [12] or aggregation trees used in sensor networks [1], but $\mathcal{X}$ is tuned to optimize several objectives concurrently during searches in a smartphone network.

In particular, the QRTs proposed in this work are optimized to (i) minimize energy consumption during search; (ii) minimize the query response time in conducting the search; and (iii) maximize the recall rate of the user query. Most existing works optimize the objectives (i-iii) individually, or optimize one and constrain the complementation. This often results in sub-optimal solutions since the objectives are conflicting and a decision maker needs an optimal trade-off [6]. To the best of our knowledge, no research study has ever dealt with the QRT problem as a *Multi-objective Optimization Problem (MOP).* The particular issue is that there is no single solution that can optimize all objectives in a MOP, but a set of non-dominated solutions, commonly known as the *Pareto Front (PF).*

Our main contributions are summarized as follows:

- We propose the *Multi-Objective Query Routing Tree (MO-QRT)* problem for Smartphone Networks and formulate it as a MOP that minimizes the energy consumption and time overhead during searches but also maximizes the recall rate of answers.
- We present SmartOpt, which is an efficient algorithm for the MO-QRT problem using a *Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D).*
- We evaluate our *SmartOpt Framework* using mobility patterns derived from GeoLife [15] and social behavior patterns derived from DBLP [5].

The remainder of the paper is organized as follows: Section II, provides our system model and defines the problem, while Section III, introduces the SmartOpt framework. Our experimental methodology and results are presented in Section IV, while Section V concludes the paper.

II. System Model and Problem Formulation

In this section we outline our system model and formulate the problem SmartOpt aims to solve. A table of respective symbols is summarized in Table I.

[3]"Customers Angered as iPhones Overload AT&T", Jenna Wortham, The New York Times (online), Sept. 2nd, 2009.

A. System Model

Overview: Let $\mathcal{C}$, denote a social networking service that maintains centrally the profiles $\mathcal{P} = \{p_1, p_2, ..., p_M\}$, for each of its M subscribed users (i.e., $\mathcal{U} = \{u_1, u_2, ..., u_M\}$). The profiles record basic user details, authentication credentials, the user interests (e.g., traveling, sports, music, etc.) and friendship relations that define the conceptual social network graph $\mathcal{G}$ among the M users. In our setting, a user u_i ($i \leq M$) uses a smartphone (or tablet) device to both perform its day-to-day activities but also to capture objects of interest at arbitrary moments (e.g., "take a picture of the Liberty Statue".) Each object o_{ik} might be tentatively *"tagged"* with GPS information and other user tags (e.g., *"lat: 40.689201355, long: -74.0447998047, tags: "Statue Liberty Ellis Island"*).

Connection Modalities: Each u_i features different Internet connection modalities that provide intermittent connectivity to $\mathcal{C}$ (e.g., WiFi, 2G/3G/4G). Each u_i also features peer-to-peer connection modalities that provide connectivity to nodes in spatial proximity (e.g., Bluetooth or Portable WiFi available in Android). We assume that when u_i is connected to $\mathcal{C}$, then $\mathcal{C}$ is aware of u_i's absolute location (e.g., GPS) or u_i's relative location (e.g., the cell-ids within u_i's range, WiFi RSS indicators within u_i's range or other means utilized for geo-location). Notice that each of the connection modalities comes at different energy and data transfer rate characteristics. For example, we've profiled an Android-based HTC Hero and found that WiFi consumes 39mW/byte, 3G consumes 24mW/byte and Bluetooth consumes 14mW/byte. Additionally, Bluetooth had a symmetric data rate of 864kbps, WiFi an asymmetric data rate of 123Kbps (up) and 2Mbps (down) and 3G an asymmetric data rate of 2.7Mbps (up) and 7.2Mbps (down). The nominal data rates for the aforementioned modalities might differ significantly, as this is also validated in [**?**], mainly due to the deployment environment. Moreover, while the power consumption on the different kinds of radios can be comparable, the energy usage for transmitting a fixed amount of data can differ an order of magnitude because the achievable data rates on these interfaces differ significantly [10]. Finally, the availability characteristics of these kinds of modalities can vary significantly. The penetration of some form of cellular availability (e.g., WiFi or 3G) is significantly higher than Bluetooth, on average. Thus, uploading or downloading large data items using Bluetooth can be more energy-efficient than using a radio network, but Bluetooth may not always be available and it is often slower.

Search Techniques: Now let an arbitrary user u_j ($j \leq M$), be interested in answering a query[4] $\mathcal{Q}$ over its social neighborhood $\mathcal{G}'$ ($\mathcal{G}' \subseteq \mathcal{G}$). For instance, let $\mathcal{Q}$ be a depth-bounded breadth first search query over u_j's neighbors in the $\mathcal{G}$ graph (i.e., in $\mathcal{G}'$). This kind of conceptual query can be realized in the following manners:

1) *Centralized Search (CS):* This algorithm assumes that the multimedia objects and tags are all uploaded to

[4]Without loss of generality we assume simple keyword queries over tags

TABLE I
TABLE OF SYMBOLS

Symbol	Description
$\mathcal{C}$	(Centralized) Social Networking Service
$\mathcal{U}$	Users of the Social Network (i.e., $\{u_1, u_2, ..., u_M\}$)
$\mathcal{P}$	User Profiles stored by $\mathcal{C}$ for $\mathcal{U}$s (i.e., $\{p_1, p_2, ..., p_M\}$)
o_{ik}	Object k (images, videos, etc.) recorded by user i.
$\mathcal{G}$	Conceptual Graph connecting the users in $\mathcal{U}$.
$\mathcal{G}'$	Social Neighborhood of some arbitrary user.
$\mathcal{Q}$	Query conducted in social neighborhood $\mathcal{G}'$ ($\mathcal{G}' \subseteq \mathcal{G}$).
$\mathcal{X}$	Query Routing Tree constructed to answer $\mathcal{Q}$.
$\mathcal{U}'$	Users that are connected to $\mathcal{C}$ during the execution of $\mathcal{Q}$.

$\mathcal{C}$ prior query execution. Once $\mathcal{Q}$ is posted, $\mathcal{C}$ can locally derive the answers (using its local tag database) and return the answers to u_j. This model, which is currently utilized by all social networking sites (such as Twitter, Youtube, Loopt, etc.), performs well in terms of query response time but performs poor both in terms of *data disclosure* (i.e., o_{ik} objects and tags need to be continuously disclosed to $\mathcal{C}$) and performance (i.e., data transmission of large objects over radio links is energy demanding).

2) *Distributed Random Search (DRS):* This algorithm assumes that the objects and tags are all stored in-situ (on their owner's smartphones). In order to realize the search task, a querying node u_j downloads from $\mathcal{C}$ the addresses (e.g., IP) of its first line neighboring nodes (i.e., $\mathcal{G}'' \subseteq \mathcal{G}'$). u_j then contacts the nodes in $\mathcal{G}''$ in order to conduct a depth-bounded breadth first-search in a P2P fashion (i.e., using a pre-specified $\mathcal{Q}_{TTL} > 0$). Once some arbitrary node $u_x \in \mathcal{G}'$ receives $\mathcal{Q}$, it looks both at its local tags, in order to identify an answer, and also forwards the request further until $\mathcal{Q}_{TTL}$ becomes zero.

Although the DRS approach improves the data-disclosure drawback of the *CS* algorithm, it is quite inefficient during search and also is inefficient in respect to energy consumption. In particular, $\mathcal{Q}$ has to go over a random neighborhood rather than a neighborhood that is contextually related to the query. For instance, in our Liberty Statue query example, we would have preferred querying a friend living in lower Manhattan rather than a person living in California (as the former would have a higher probability of capturing the statue). Also, if u_j had two friends, u_x and u_y, both living in lower Manhattan, with u_x being in spatial proximity to u_j during the query (i.e., within a few meters), while u_y being far away, would have made u_x a better choice for posting the query (as u_x could have been queried through a local link such as Bluetooth).

B. Problem Formulation

The *Multi-Objective Query Routing Tree (MO-QRT)* structure, proposed in this paper, improves the search operation of the DRS algorithm by optimizing the neighbor selection process. In particular, a node downloads from $\mathcal{C}$ a QRT $\mathcal{X}$ that is optimized according to the following formulation: *Given a social network of users $\mathcal{U}$, a list of active users $\mathcal{U}'$ and their coordinates, the profiles $\mathcal{P}$ of these users and a query $\mathcal{Q}$, posted by an arbitrary user u_j, $\mathcal{C}$ aims to optimize an $\mathcal{X}$ structure using the following* **objectives**:

Objective 1: *Minimize the total* Energy *consumption of $\mathcal{X}$*

$$Energy(\mathcal{X}) = min \sum_{\forall (u_a,u_b) \in \mathcal{X} (\mathcal{X} \subseteq \mathcal{U}')} e(u_a, u_b) \tag{1}$$

where, $e(u_a, u_b)$ denotes the energy consumption for transmitting one bit of data over the respective edge (WiFi, Bluetooth and 3G).

Objective 2: *Minimize the* Time *overhead of $\mathcal{X}$*

$$Time(\mathcal{X}) = min(max_{(u_a,u_b) \in \mathcal{X}} t(u_a, u_b)) \tag{2}$$

where, $t(u_a, u_b)$ denotes the delay in transmitting one bit of data over the respective edge.

Objective 3: *Maximize the* Recall *rate of $\mathcal{X}$*

$$Recall(\mathcal{X}, \mathcal{Q}) = max(\frac{Relevant(\mathcal{Q}) \cap Retrieved(\mathcal{X}, \mathcal{Q})}{Relevant(\mathcal{Q})}) \tag{3}$$

where *Relevant($\mathcal{Q}$)* denotes the set of all objects in $\mathcal{U}'$ that are relevant to $\mathcal{Q}$, formally as: *Relevant($\mathcal{Q}$)* = $\bigcup_{\forall u_a \forall k (u_a \in \mathcal{U}')}(o_{ak}))$, given that u_a's profile (denoted as p_a) contains terms found in $\mathcal{Q}$. On the other hand, *Retrieved($\mathcal{X}$,$\mathcal{Q}$)* denotes the set of objects that have been retrieved in response to $\mathcal{Q}$ over structure $\mathcal{X}$, formally as *Retrieved($\mathcal{X}$, $\mathcal{Q}$)* = $\bigcup_{\forall u_a \forall k (u_a \in \mathcal{X})}(o_{ak}))$, again given that p_a contains terms found in $\mathcal{Q}$.

In a MOP, there is no single solution $\mathcal{X}$ that optimizes all objectives simultaneously, but a set of trade-off candidates. The set of trade-off solutions, commonly known as the Pareto Front (PF), is often defined in terms of Pareto Optimality [7]. That is, considering a maximization MOP with n objectives: a solution $\mathcal{X}^*$ is considered non-dominated or Pareto optimal with respect to another solution $\mathcal{Y}$, iff $\forall i \in \{1, ..., n\}, \mathcal{X}_i \geq \mathcal{Y}_i \wedge \exists i \in \{1, ..., n\} : \mathcal{X}_i > \mathcal{Y}_i$, this is denoted as $\mathcal{X} \succ \mathcal{Y}$.

III. THE SMARTOPT OPTIMIZER

In this section, we present the SmartOpt Query Optimizer (referred to *SmartOpt* hereafter), which solves the MO-QRT problem in an online manner. We assume that some arbitrary user u_j generates a query $\mathcal{Q}$ and forwards it to the $\mathcal{C}$. $\mathcal{C}$'s optimizer is then employed for finding a diverse and high-quality set of non-dominated smartphone QRTs that can facilitate the resolution of $\mathcal{Q}$. Our framework proceeds in three phases: a) *the Pre-Processing phase*, during which the problem is decomposed into a set of sub-problems; b) *the Optimization Phase*, during which a set of Pareto-optimal QRTs is identified; and c) *the Dissemination Phase*, during which the solution QRT is propagated to u_j and the search process is initiated.

A. Pre-Processing Phase

The pre-processing phase consists of representing a QRT and decomposing the problem into a set of scalar sub-problems.

1) **Representation:** In our approach, a solution[5] $\mathcal{X}$ is a query routing tree with $|\mathcal{G}'|$ active smartphone users that can participate in the resolution of $\mathcal{Q}$. Without loss of generality, let $\mathcal{X}$ be represented as a vector in which each index i corresponds to a user u_i and the value of that position corresponds to u_i's parent. The root of the tree is the query user (for simplicity noted as u_1). A negative value -1 in any position indicates that the given users is not currently selected in the query routing tree $\mathcal{X}$.

2) **Decomposition:** Initially, the MOP should be decomposed into m sub-problems by adopting any technique for aggregating functions [14], e.g., the Chebyshev approach used here. In this paper, the i^{th} sub-problem is in the form

$$maximize \quad g^i(\mathcal{X}|w^i_j, z^*) = max\{w^i_j|f_j(\mathcal{X}) - z^*_j|\} \qquad (4)$$

where f_j, $j = 1, 2, 3$, are the objectives of our MOP formulated earlier in Subsection II-B, $z^* = (z^*_1, z^*_2, z^*_3)$ is the reference point, i.e. the maximum objective value $z^*_j = max\{f_j(\mathcal{X}) \in \Omega\}$ of each objective $f_j, j = 1, 2, 3$ and Ω is the decision space.

B. Optimization Phase

In this phase, SmartOpt optimizes in an online manner the solution space using a set of genetic operators.

1) **Initialization Step:** In Step 1 of our algorithm, we adopt a random method to generate m QRT solutions for the initial internal population (i.e., IP_0). Namely, a QRT solution $\mathcal{X}$ is initiated by setting each smartphone user $u_i, i = 1 \ldots M$ as a parent. Then, mobile users $u_j, j = 1 \ldots M$ are uniformly randomly selected, and u_i is set as u_j's parent iff $i \neq j$ and u_i is either the root or has already a parent. If u_j has already a parent then we stop and we set as parent the user u_{i+1}. This continues until all users u_i are set as parents once. Thereinafter, the Internal Population IP_{gen} is used to store the best QRT solution $\mathcal{X}^i$ found for each sub-problem g^i during the search, i.e. in each generation gen.

2) **Genetic Operator Step:** The genetic operators (i.e. selection, crossover and mutation) are then invoked on IP for offspring reproduction, i.e. generate a new QRT solution $\mathcal{Y}^i$ for each sub-problem $g^i, i = 1 \ldots m$. A M-tournament selection operator [9], a two-point (2x) crossover and a swap mutation operator [14] are utilized in our approach. The selection operator is responsible to greedily select two parent solutions for mating and forward them to the crossover operator. The *$2x$-point crossover* exchanges information from two parent QRT solutions (Pr_1, Pr_2) and generates two new QRTs O_1, O_2 - the offspring, as follows:

- Two crossover points x_1 and x_2 are uniformly randomly selected from numbers 1 to M-1, where $x_1 < x_2$.
- The pieces of the parents Pr_1 and Pr_2 falling within x_1 and x_2 are exchanged to produce two offspring, e.g., O_1, O_2.
- The best offspring O is then forwarded to the mutation operator, where $O = O_1$ if $g^i(O_1, w^i_j) > g^i(O_2, w^i_j)$ and $O = O_2$ otherwise.

[5]The terms *"solution"*, *"vector"* and *"QRT"* are utilized interchangeably.

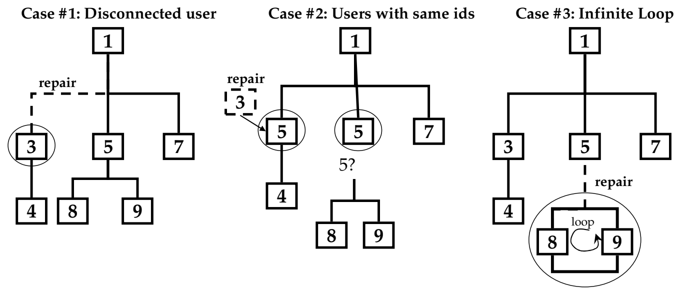


Fig. 2. The repair operator of SmartOpt optimizer.

The *swap mutation operator* modifies an offspring O to a solution $\mathcal{Y}$ with a probability r_m by uniformly randomly swapping the values (i.e. parents in the tree) of two indexes j, z of the QRT $\mathcal{Y}$. The modified QRT solution $\mathcal{Y}$ is then forwarded to the repair heuristic.

3) **Repair Operator Step:** In Step 2.2 of our algorithm, a local heuristic checks a QRT solution $\mathcal{Y}$ and calculates a QRT $\mathcal{Z}$ iff:

- **Case #1:** there is a disconnected user u_i in QRT $\mathcal{Y}$ (i.e. u_i with or without children that does not have a parent);
- **Case #2:** two or more user ids i of user u_i are the same in QRT $\mathcal{Y}$;
- **Case #3:** there is an infinite loop in QRT $\mathcal{Y}$;

In all cases (illustrated in Figure 2), the solution $\mathcal{Y}$ is considered infeasible. An infeasible solution can be generated during reproduction (i.e. genetic operation). A local heuristic repairs the QRT solution $\mathcal{Y}$ to $\mathcal{Z}$ by: uniformly randomly generating a parent for the disconnected user u_i in Case #1, replacing the duplicate user u_i with another user u_j in Case #2, breaking the loop by connecting a random user of the loop with another user out of the loop in Case #3. All repair techniques are shown with dotted lines in Figure 2. The repair heuristic continuously repairs solution $\mathcal{Y}$ until it does not fall in any of the Cases #1, #2 or #3. Solution $\mathcal{Z}$ is then used to update the populations of MOEA/D.

C. Dissemination Phase

The proposed SmartOpt opts for the most suited Pareto-optimal QRT $\mathcal{X}^* \in PF$ based on instant requirements and forwards it to the query user u_j. The query user u_j then utilizes QRT X^* to initiate the search and find objects of interest o_{ik} recorded by user $u_i \in \mathcal{X}^*$ and related to query $\mathcal{Q}$.

IV. Experimental Evaluation

In this section we present our experimental methodology and the results of our evaluation.

A. Evaluation Methodology

In this section we describe our trace-driven experimental methodology in order to assess the effectiveness of our framework.

Datasets and Queries: In the absence of a real dataset capturing our problem setting, we have constructed a synthetic scenario from the following two real datasets:

TABLE II
EXPERIMENTAL EXECUTION SCENARIOS

Scenario	$\mathcal{Q}$	Time	$\mathcal{G}'$	# Objects	Relevant Objects
T1	Query1	Morning	49	3877	82
T2	Query1	Noon	58	5504	73
T3	Query1	Night	95	8884	121
T4	Query2	Morning	49	3877	319
T5	Query2	Noon	58	5504	477
T6	Query2	Night	95	8884	695

i) *GeoLife* [15]: This real dataset by Microsoft Research Asia, includes 1,100 trajectories of a human moving in the city of Beijing over a life span of two years (2007-2009). The average length of each trajectory is $190,110 \pm 126,590$ points, while the maximum trajectory length is 699,600 points. Notice that 95% of the GeoLife dataset refers to a granularity of 1 sample every 2-5 seconds or every 5-10 meters.

ii) *DBLP* [5]: This real dataset by the DBLP Computer Science Bibliography website, includes over 1.4 million publications in XML format. In particular, the dataset records the paper titles, paper urls, co-authors, links between papers and authors and other useful semantics. In order to map this dataset to our problem, we assume that each object o_{ik} is an author's paper. We also assume that each object is "tagged" by the keywords found in the paper title. The social graph $\mathcal{G}$ is constructed by the co-author relationships that are part of the dataset.

In order to link datasets (i) and (ii) and create our execution, we have mapped the top 1,100 DBLP authors (those with the most papers), using a 1:1 correspondence, to the 1,100 trajectories found in the GeoLife dataset. We then utilize the following two queries with the combinations presented next:

```
-- Query 1:
SELECT S.title, S.url
FROM SmartphoneUsers S, Query Q
WHERE (distance(S.x,S.y,Q.x,Q.y) < 10 KM)
      AND S.Title LIKE '%optimization%';

-- Query 2:
SELECT S.title, S.url
FROM SmartphoneUsers S, Query Q
WHERE (distance(S.x,S.y,Q.x,Q.y) < 10 KM)
      AND S.Title LIKE '%networks%';
```

We execute six different scenarios using Query 1 and Query 2 as shown on Table II. Our scenarios are executed for three different time periods (i.e., during the morning, during noon and during night), in order to capture different mobility patterns that are inherent in the GeoLife dataset.

ii) Search Algorithms: We have implemented the *Centralized Search* algorithm (CS), the *Distributed Random Search* (DRS), as these were described in Subsection II-A, as well as the *SmartOpt* search algorithm that utilizes an optimized QRT $\mathcal{X}$ to carry out a query execution as opposed to the random neighbor selection utilized in DRS. We evaluate the search algorithms using the following metrics: *Time*, *Energy* and *Recall*, as these were defined in Section II-B and using the time and energy profiles for our Smartphone devices, we have presented in Section II-A.

Experimental Setup: Our simulation experiments were performed on a Lenovo Thinkpad T61p PC with an Intel Core 2 Duo CPU running at 2.4GHz and 4.0 GB of RAM. In order to collect realistic results for a large period of time, we collect statistics for 100 time instances in each experiment. To increase the fidelity of our measurements we have repeated each experiment 5 times and present the average performance for each type of plot.

B. Evaluation of SmartOpt Search

In this section we evaluate the performance of the SmartOpt optimizer against the Centralized Search (CS) and Distributed Random Search (DRS) (presented in Section II-A), using 100 consecutive timestamps from the GeoLife dataset. At each *timestamp (ts)*, we compare the energy consumption, time overhead and recall of all algorithms. Since SmartOpt returns multiple solutions at each timestamp, we plot the average of all solutions (denoted as SmartOpt$_{avg}$). Additionally, we plot the best solution (denoted as SmartOpt$_{best}$) to evaluate the efficiency of SmartOpt under each performance metric. Figure 3 illustrates the results of our experiment for all performance metrics.

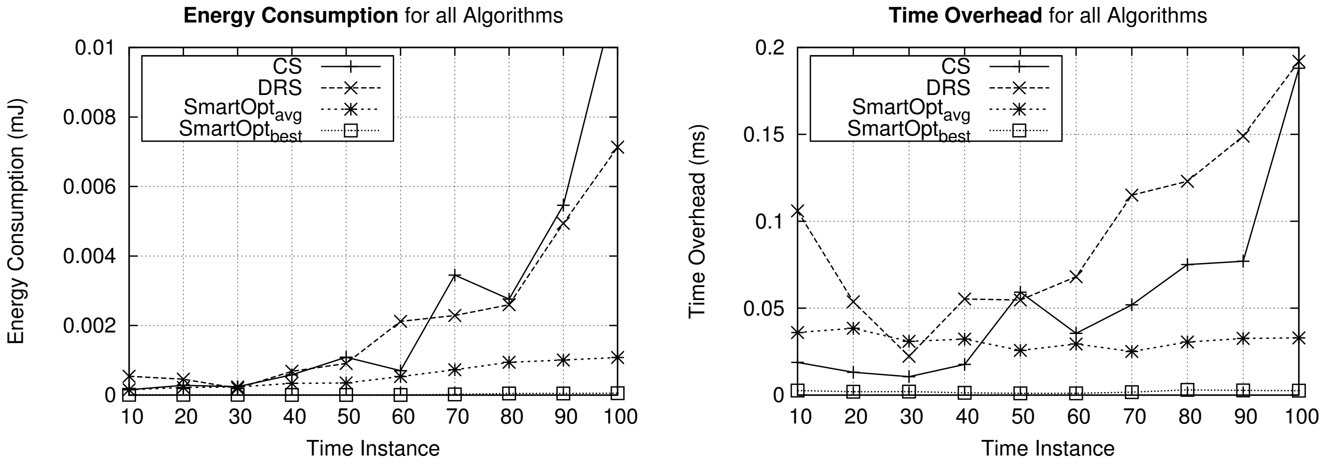
In Figure 3 (top/left) we observe that the energy consumption of SmartOpt$_{avg}$ and SmartOpt$_{best}$ always outperform CS and DRS in all timestamps. This is more evident when $ts \geq$60 where the number of users in the network rapidly increases. In these timestamps the difference in energy consumption reaches as high as 1467%, which translates to two orders of magnitude energy savings by the SmartOpt.

Similar observations apply for Figure 3 (top/right) where we demonstrate the time overhead for all algorithms. However, in this Figure, we observe that in the range 0$\leq$ ts $\leq$38, CS outperforms SmartOpt$_{avg}$. The reason behind this is that CS uses more 3G direct connections to the query user thus minimizing the time overhead. This, however, introduces an additional energy cost as shown previously.

Finally, in Figure 3 (bottom) we show the recall performance for all algorithms. In this Figure, we observe that CS outperforms all algorithms by demonstrating always 100% recall. This is expected as CS always retrieves results from all users in the network. SmartOpt$_{best}$ comes second with $\approx$ 95% recall while DRS comes third with $\approx$ 54%.

V. CONCLUSIONS

In this paper, we present the SmartOpt framework for searching objects captured by the users in a mobile social community. Our framework, is founded on an *in-situ* data storage model and searches then take place over the MO-QRT structure we propose in this paper. Our structure concurrently optimizes several conflicting objectives (i.e., energy, time and recall). Our experimental evaluation, with mobility patterns derived by Microsoft's Geolife project and social patterns derived by DBLP, shows reveals that SmartOpt can yield query



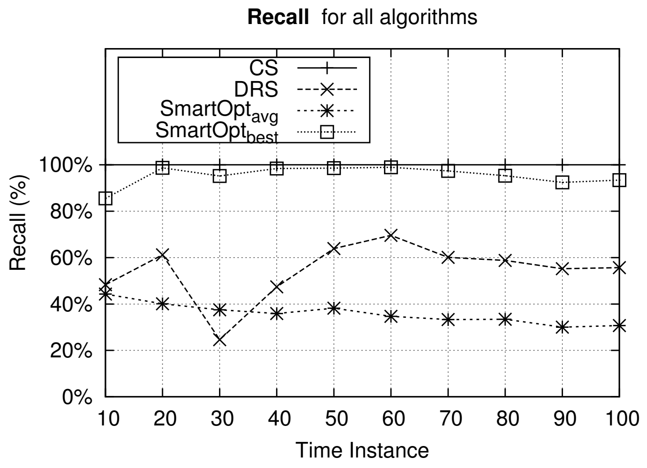


Fig. 3. Evaluation of the CS, DRS and SmartOpt search algorithms using the energy, time and recall performance.

recall rates of 95%, with one order of magnitude less time and two orders of magnitude less energy than its competitors. In the future we plan to implement a real prototype of our framework using the SmartNet programming cloud, which consists of several Android-based smartphones, that we are currently setting up. We additionally plan to evaluate extensions of our framework addressing multi-query optimization techniques.

Acknowledgments: This work was supported in part by the second author's Startup Grant, funded by the University of Cyprus, EU's FP7 CONET project (#224053), EU's FP6 Marie Curie TOK "SEARCHiN" project and Cyprus National project FireWatch (#ΑΕΙΦΟΡΙΑ/ΦΥΣΗ/0609(BIE)/09).

REFERENCES

[1] Andreou P., Zeinalipour-Yazti D., Pamboris A., Chrysanthis P.K., Samaras G., "Optimized Query Routing Trees for Wireless Sensor Networks," In *Information Systems*, Elsevier, 2011.

[2] Azizyan M., Constandache I., Choudhury R.-R., "SurroundSense: mobile phone localization via ambience fingerprinting," In *MobiCom'09*.

[3] Campbell A., Eisenman S., Lane N., Miluzzo E., and Peterson R., "People-centric urban sensing," In *WICON*, 2006.

[4] Das T., Mohan P., Padmanabhan V.N., Ramjee R., Sharma A., "PRISM: platform for remote sensing using smartphones," In *MobiSys*, 2010.

[5] DBLP Computer Science Bibliography, 2010.

[6] Deb K., "Multi-Objective Optimization Using Evolutionary Algorithms," *Wiley and Sons*, 2002.

[7] Deb K., Pratap a., Agarwal S., Meyarivan T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA II," *IEEE TEC*, 2002.

[8] Google Latitude, 11/2010, http://www.google.com/latitude

[9] Konstantinidis A., Yang K., Zhang Q., Zeinalipour-Yazti D., "A Multi-Objective Evolutionary Algorithm for the Deployment and Power Assignment Problem in Wireless Sensor Networks," In Computer Networks, Elsevier, 2010.

[10] Ra M.-R. et. al. "Energy-delay tradeoffs in smartphone applications," In *MobiSys*, 2010.

[11] Thiagarajan A., Ravindranath L., LaCurts K., Madden S., Balakrishnan H., Toledo S., Eriksson J., "VTrack: Accurate, Energy-aware Road Traffic Delay Estimation using Mobile Phones," In *SenSys*, 2009.

[12] Zeinalipour-Yazti D., Kalogeraki V., Gunopulos D., "pFusion: An Architecture for Internet-Scale Content-Based Search and Retrieval," IEEE TPDS, vol. 18, no. 6, pp. 804-817, June 2007.

[13] Zeinalipour-Yazti D., Kalogeraki V. and Gunopulos D., "Exploiting Locality for Scalable Information Retrieval in Peer-to-Peer Systems," Information Systems (InfoSys), vol. 30, iss. 4, pp. 277-298, 2005.

[14] Zhang Q., Li H., "MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition," IEEE Transactions on Evolutionary Computation, 2007.

[15] Zheng Y., Liu L., Wang L., Xie X., "Learning transportation mode from raw gps data for geographic applications on the web," In *WWW'08*.