A Framework for Managing the Selection of Spatiotemporally Relevant Information Providers

Georgios Tychogiorgos
Electrical and Electronic Engineering
Imperial College
London SW7 2AZ, UK
g.tychogiorgos@imperial.ac.uk

Chatschik Bisdikian
IBM Research
Thomas J. Watson Research Center
Hawthorne, NY 10514, USA
bisdik@us.ibm.com

***Abstract*—The development of future pervasive sensor-enabled systems, where information is distributed on-demand across heterogeneous networks, highlights the necessity for an efficient framework to determine the relevancy of provided information with respect to one's needs. This paper considers the problem of selecting the most "spatiotemporally" relevant providers in order to meet a user's information needs over a time period of interest. Initially, a definition and a measure of spatiotemporal relevancy is developed to measure the degree of relevancy of sensory information with respect to both its spatial and temporal characteristics. Based on these, the selection of the most relevant set of providers under budget constraints is expressed as an integer programming optimization problem and a two-level dynamic programming (DP) algorithm is proposed to solve it optimally. Moreover, a number of alternative methods are proposed in order to accelerate the provider selection process by making approximations either to the overall optimization problem formulation or the relevancy calculation method itself. Finally, the performance of the proposed methods are examined both analytically and by simulation for a number of provider scenarios.**

I. Introduction

We are experiencing an evolution in information-rich sensor-enabled, pervasive computing environments, such as those exemplified by the *Internet of Things* [1] and *social sensing* (participatory- and crowd-sensing) [2], where information *consumers* bind to information *providers* (sensory and otherwise) openly, on demand, and across administrative domains. The latter applies to cases where the owners and operators of the end-applications (the consumers) do not necessarily own and/or operate the sensory information sources (the providers) they engage with, as the case could be with multi-agency, smart planet type of applications.

Managing the multi-aspect relationships between the sensory information consumers and producers in such dynamic environments is a challenging research pursue. Considering one aspect of such a relationship, that of selecting relevant providers, [3] presented some initial work exemplified by the following use-case scenario. A city agency would like to collect sensory information to support city-wide quality-of-life services (such as air-quality, noise levels, etc.) for its city dwellers and visitors. Because of lack of sufficient sensory resources, the geographical extend of the project, and of course budget restrictions, the city agency decided to engage with third party sensory information sources as necessary. In this setting, [3] focused on the challenge of the city agency (i.e., of an information consumer) selecting the set of sources (i.e., the information providers) among a set of provider candidates that were deemed most relevant to its information needs based on their *spatial relevance* given cost constraints. Spatial relevance relates to how close or far is the coverage and *quality of* the provided *information* (e.g., $\pm 5\%$ measurement deviations from the actual value for region $\mathcal{X}$, $\pm 10\%$ for region $\mathcal{Y}$, etc.) in relation to that desired, assuming, of course, the semantic equivalency between the desired and provided pieces of information, e.g., temperature measurements.

However, looking only along the spatial dimension reflects a rather *static* operational and management set-up, where a single provider selection decision remains valid throughout the temporal dimension. In this paper, we relax the static assumption to explicitly consider temporal variations as well and investigate the selection of providers that are *spatiotemporally relevant*. This will facilitate the process of managing the selection of providers in more dynamic cases where information consumers and providers exhibit *both* temporal *and* spatial behaviors, such as when: (a) the consumer requirements change over time and space (e.g., the city agency requires increased granularity for the air-quality measurement during rush hours and/or its regional interests change over the course of day or a week); or (b) the provider capabilities change over time and space, (e.g., using mobile sensors, and/or employing a varying number of them over time).

Spatiotemporal-related aspects for sensor networks have been studied in many occasions. For example, issues related to deployment strategies for effective spatial coverage of sensor networks are highlighted in [4], [5], [6] and references therein. Refs. [7], [8] provide overviews of spatio-temporal data models, the former about geospatial data in multidimensional databases, and the latter about spatial and temporal data models for storing sensor information in database systems.

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

978-3-901882-50-0 ©2013 IFIP

Ref. [9] considers spatial and thematic relevancy for matching documents to queries focusing in document ranking. Ref. [10] considers the problem of selecting the appropriate battery-operated sensors in order to maximize the life-time of the network based on the spatio-temporal relevancy correlation of the measured information between sensors in the same area. A similar problem is also investigated in [11] regarding in-network data aggregation of spatially and temporally correlated information generated by neighboring sensors.

Our work examines a distinctively different aspect for sensor networks from the above studies. It is concerned with the operational aspects of information consumers dynamically selecting information providers (possibly representing multiple sensor networks) under spatially varying and temporally evolving interests and capabilities of the consumers and providers, respectively, while considering *both* coverage *and* QoI aspects. Specifically, the contributions made in this paper are:

- the definition and a measure for *spatiotemporal relevance* between consumers and providers of sensory information;
- the optimization framework for managing the selection of sets of spatiotemporally relevant sensory information providers;
- the DP-based solution of the said optimization, its performance and complexity assessment;
- the development of solution acceleration techniques, their performance and time-complexity trade-offs, and their relationship to the system-level description.

The rest of the paper is organized as follows: Section II defines the spatiotemporal relevance. Section III formulates the selection of the best set of providers as an integer programming problem and presents a DP algorithm to solve it. Section IV presents three alternative problems that approximate for the problem in Section III, and Section V describes two relevancy approximation methods. Section VI provides simulation results to evaluate the accuracy and time requirements of the proposed methods. Finally, Section VII concludes our work and outlines our future research.

II. Spatiotemporal Relevance

In its broadest sense, the problem space at hand is that of information consumers selecting (and binding to) information providers that are most *spatiotemporally relevant* to their information needs while satisfying stated operational constraints. To this end, we start with definitions and terminology regarding spatiotemporal relevancy building upon those in [3] but pursuant of the broader spatiotemporal focus of this paper.[1]

Let the *spatiotemporal point* ϕ represent the pair (w,t), where w is a spatial point (i.e., a location) in $\mathbb{R}^n$, $n \geq 1$, e.g., $w=(x,y) \in \mathbb{R}^2$, and t be a time instant, $t \in \mathbb{R}$. Let R_e, $e \in \{d,p\}$, be a spatiotemporal region (volume, to be exact) where the *desired* (d) or *provided* (p) quantities of interest are defined, e.g., R_d denotes a volume of interest, such as, in the case of the city agency of our use scenario, the southern part of the city during a weekday evening rush hour. Let also R_e^t be the *spatial slice* of R_e at time t, i.e., $R_e^t = \{w : (w,t) \in R_e\}$. We can define the *temporal slice* R_e^w at location w likewise.

[1] As noted in the introduction, we will assume semantic equivalency of the sought after and provided pieces of information.

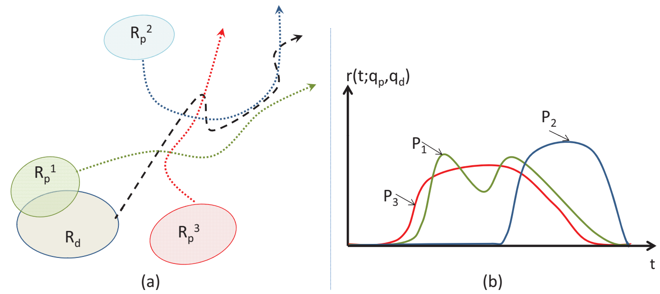

Fig. 1. (a) Moving consumer and providers, and (b) time-varying temporal relevancy

For each $\phi \in R_d$, let $q_d(\phi)$ represent the quality of the desired information, e.g., its accuracy, latency, etc., i.e., the *quality of information* (QoI) [12] a consumer, e.g., the city agency, seeks. Likewise, for each $\phi \in R_p$, let $q_p(\phi)$ represent the QoI provided by an information provider, e.g., by a sensory information provider owning air-quality measuring sensors mounted on buildings or on vehicles of a vehicle fleet, etc. We refer to $q_e(\cdot)$, $e \in \{d,p\}$, as the (spatiotemporal) *QoI functions*.

For given $q_d(\cdot)$ and $q_p(\cdot)$, and $\phi \in R_d \cap R_p$, let the *value of information function* (or *VoI function*) $v\big(q_p(\phi); q_d\big)$ represent the benefit (or value) that the consumer attains when using information of quality $q_p(\phi)$ when q_d was desired. We use the VoI (anticipated to be) received from a provider to define:

Definition 1. *The temporal relevancy $r(t; q_p, q_d)$ of a provider p at time t relative to a consumer is the ratio of the total value to be attained by the consumer from the information provided at time t and the total value it would have been attained if information of the desired quality at time t were to be received:*

$$r(t; q_p, q_d) = \frac{\int_{R_d^t \cap R_p^t} v\big(q_p(w,t); q_d\big)\, dw}{\int_{R_d^t} v\big(q_d(w,t); q_d\big)\, dw}. \qquad (1)$$

Fig. 1.a illustrates temporarily evolving desired and provided regions for three providers. Fig. 1.b illustrates the changing temporal relevancy for the providers noting the changes in the order of the most relevant providers with time.

If no extra value is gained when information of higher quality is provided than what is desired, then $v\big(q_p(\phi); q_d\big) \leq v\big(q_d(\phi); q_d\big)$ and, hence, $0 \leq r(t; q_p, q_d) \leq 1$. For the numerical results later in the paper, we will illustratively use the simple exemplary VoI function

$$v\big(q_p(\phi); q_d\big) = \min\{q_p(\phi), q_d(\phi)\}, \qquad (2)$$

which easily satisfies the previous condition.

Finally, let the time horizon of interest be $\mathcal{T} = \{t : \exists w \text{ s.t. } (w,t) \in R_d\}$ and let $|\mathcal{T}|$ be its "duration," then:

Definition 2. *The* spatiotemporal relevancy $r_{st}(q_p; q_d)$ *of a provider p relative to a consumer is its temporal relevancy* *in* (1) *averaged over the entire time horizon of interest* $\mathcal{T}$:

$$r_{st}(q_p; q_d) = \frac{\int_{\mathcal{T}} r(t; q_p, q_d)dt}{\int_{\mathcal{T}} dt} = \frac{1}{|\mathcal{T}|} \int_{\mathcal{T}} r(t; q_p, q_d)dt. \quad (3)$$

If information at one time is desired more than another (i.e., "must have" vs. "nice to have") multiplicative weights for $r(t; \cdot, \cdot)$ could have been used in (3). However, we assume that these are already accounted for within the definition of $r(t; \cdot, \cdot)$. Note that an expression corresponding to (3) can be defined for discrete time using sums instead. A relevancy measure can be defined likewise using spatial instead of temporal averages. However, such alternatives are not considered further in this paper, noting that the computation framework we present next applies similarly to these cases as well.

III. The Provider Selection Framework

In this section, we develop the core of the provider selection management framework expressed via an optimization problem model and a solution methodology. In the following sections, we present special cases of the optimization model that lead to accelerated but approximate solutions to the problem.

A. System model and assumptions

We consider a discrete time decision process for the information consumer. Specifically, over the time interval $\mathcal{T} = [0, T]$, let $\{t_n, n \in \mathcal{N} = \{1, \ldots, N\}\}$ be a sequence of provider selection points, i.e., instants at which the consumer may decide to switch providers. The decision seeks to maximize the spatiotemporal relevancy of the information received under various operational constraints. Note that we will also refer to t_n as time n and the time interval $\Delta t_n = [t_n, t_{n+1})$ as slot n; we will use the discrete time version of (3).

The operational constraints are expressed through a resource (or budget) constraint B, which could represent a monetary constraint, an energy constraint, amount of risk, and so on. There is a set $\mathcal{K} = \{1, \ldots, K\}$ of providers that can be engaged at one time or another during the interval $\mathcal{T}$, and the cost of engaging with provider k over slot n is c_n^k. It is assumed that providers communicate their capabilities to the consumer using representations of their respective QoI functions $q_p^k(\cdot)$, $k \in \mathcal{K}$. The consumer combines this with its desires, expressed by its own QoI function $q_d(\cdot)$ to compute a provider's spatiotemporal relevance as described in the previous section.

Whenever a consumer engages with two or more providers serving the same spatiotemporal point, we assume that it "experiences" the QoI function of an aggregated provider that results from the QoI functions of these providers. Specifically, if ϕ is a spatiotemporal point within the volumes of two (or more) providers, and $\boldsymbol{x}_\phi$ is the provider *selection mask* indicating which providers the consumer is using at point ϕ,[2] then the aggregated provider QoI function $q_p^{\boldsymbol{x}}(\phi)$ is described by the transformation $h\big(q_p^k(\phi) | k \in \mathcal{K}, x(k) = 1\big)$. For example, if the accuracy of measurements at point ϕ from two providers is, say, 97% and 95%, then the QoI resulting from the aggregation of the two providers could be taken to be 97%, i.e,. the better accuracy of the two; hence, in this case, $h(\cdot) \equiv \max\{\cdot\}$. For the numerical results later on, we will illustratively use the latter exemplary QoI transformation $h(\cdot)$.

[2] $x_\phi(k) = 1$ if provider $k \in \mathcal{K}$ is used at the point ϕ, and 0 otherwise. For convenience, if context permits it, we will drop the ϕ from $\boldsymbol{x}_\phi$.

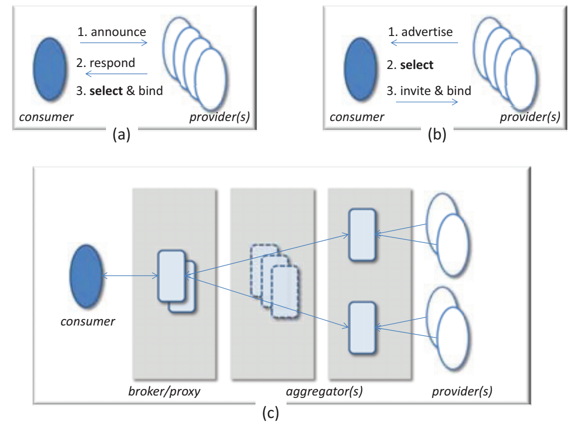

Fig. 2. Provider/consumer communication paradigms

Note that how exactly the aforementioned communication of provider capabilities to the consumer takes place, e.g., via publish/subscribe or service discovery advertisements [13], is beyond the scope of this paper. The computation framework presented next can be adapted to a number of communication paradigms. Examples of the latter are shown in Fig. 2, where in Fig. 2.a there is an active search on behalf of the consumer that first announces its information needs, while in Fig. 2.b the consumer passively listens to information capability advertisements from providers and then proceeds accordingly. Fig. 2.c shows a network topology where a consumer may delegate its selection process to a broker that interacts with the providers and makes selection recommendations to the consumer. On the other hand, owned to their spatial distribution, providers may advertise their capabilities via aggregators that aggregate the information capabilities of various local, regional, etc., sources. Through all these alternatives, the computation framework presented next relates to the `select` step.

In the sequel, we visualize the consumer/provider interaction paradigm according to, say, Fig. 2.b, without precluding any alternative paradigms, for example, our framework could be used iteratively at each aggregation level in Fig. 2.c.

B. Problem formulation

Let $\mathbf{I} = [I(k,n)]_{K \times N}$ be the $K \times N$ provider selection matrix, where $I(k;n)$ equals 1 when provider k is engaged during slot n, and 0 otherwise. Also, let $q_p^{\mathbf{I}}(\cdot)$ be the QoI function of a "super-provider" created from the aggregation of the providers marked in the selection matrix $\mathbf{I}$. The problem at hand is, then, described by the following optimization formulation:

Problem $\mathbf{\Pi}_0$: Find the provider selection matrix $\mathbf{I}_{K \times N}$ that maximizes $r_{st}\big(q_p^{\mathbf{I}}; q_d\big)$, such that:

$$\sum_{n=1}^{N} \sum_{k=1}^{K} I(k,n) c_n^k \leq B, \text{ where } I(k,n) \in \{0,1\} \ \forall (k,n). \quad (4)$$

Algorithm 1 – First Level DP Algorithm (for given time n)

output: for given n, $q_d(\cdot, n)$, $q_d^k(\cdot, n)$, the matrix $\mathbf{V}^n_{K\times B}$ containing the optimal spatiotemporal relevancy along with the selection masks $\boldsymbol{x}^k_{n,b}$, $k \in \mathcal{K}, b \in \mathcal{B}$ (see [3] for details).

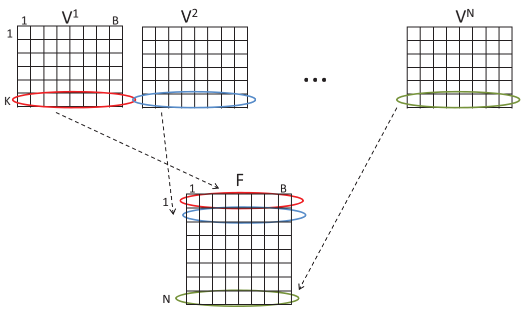

Fig. 3. Matrix merging at the second level DP algorithm

Algorithm 2 – Second level DP Algorithm

```
1: providers: k ∈ 𝒦 = {1,...,K}, time: n ∈ 𝒩 = {1,...,N}, budget: b ∈ ℬ = {1,...,B}
2: for n = 1 to N do
3:    Run Algorithm 1 for q_d(·,n) and q_p^k(·,n) ∀k ∈ 𝒦;
4: end for
5: Create matrix F, where F[n,b] = V^n[K,b] (see Fig. 3), and keep masks x^K_{n,b}, n ∈ 𝒩, b ∈ ℬ;
6: for n = 2 to N do
7:    for b = 1 to B do
8:       ℬ_b = {1,...,b};
9:       S(n,b) = max_{β∈ℬ_b} {S(n−1, b−β) + F(n,β)}; where S(1,b) = F(1,b), ∀b ∈ ℬ, S(n,0) = 0, ∀n ∈ 𝒩;
10:      β_max = arg max_{β∈ℬ_b} (S(n−1, b−β) + F(n,β));
11:      I^b_n = x^K_{n,β_max}; Prev(n,b) = b − β_max;
12:   end for
13: end for
```

Problem $\mathbf{\Pi}_0$ is a generalization of the classical *knapsack problem* in two ways: (a) because of the potential overlap of the provider coverage regions, the benefit of selecting a particular provider depends on the providers that have already been selected; and (b) there is the temporal aspect of the problem. The 0-l knapsack problem is an NP-hard problem [14] and therefore problem $\mathbf{\Pi}_0$ is also NP-hard and there is no known algorithm that calculates the optimal solution in polynomial time. The most efficient algorithm to solve the 0-l knapsack problem is a dynamic programming (DP) algorithm that manages to find the optimal solution in *pseudo-polynomial* time by splitting the problem into smaller subproblems and storing intermediate results in memory.

C. The solution

To solve problem $\mathbf{\Pi}_0$, we use a two-layer algorithmic procedure comprising algorithms 1 and 2. Algorithm 1 is based on the DP algorithm in [3]. For a given time $n \in \mathcal{N}$, and corresponding QoI functions $q_d(\cdot, n)$ and $q_d^k(\cdot, n)$, $k \in \mathcal{K}$, Algorithm 1 produces the optimal selection mask $\boldsymbol{x}^k_{n,b}$ when the available budget is $b \in \mathcal{B} = \{1, \dots, B\}$ in the presence of providers 1 through k only. It also produces the $K \times B$ matrix $\mathbf{V}^n$, whose (k, b) entry contains the optimal spatiotemporal relevancy that corresponds to $\boldsymbol{x}^k_{n,b}$. Due to space limitations, see [3] for further details on Algorithm 1.

Clearly, for each of the $\mathbf{V}^n$ matrices, $n \in \mathcal{N}$, the interest lies with their last row which represents the optimal decision for a given budget $b \in \mathcal{B}$ while considering all K providers. Thus, Algorithm 2 (not in [3]) uses the last row of the $\mathbf{V}^n$ matrices as the n-th row of the $N \times B$ matrix $\mathbf{F}$, see Fig. 3 and line 5 in Algorithm 2. Subsequently, the matrix $\mathbf{F}$ is used to construct the matrix $\mathbf{S}$ necessary for the calculation of the optimal selection masks of the overall problem.

The key steps in constructing $\mathbf{S}$ are lines 6–13 in Algorithm 2. Specifically, at each iteration n, we: (a) choose the allocation of budget $\beta_{\max}$ (hence, the allocation of $b-\beta_{\max}$ for all previous $n-1$ slots as well) that gives the highest aggregate relevancy; and (b) store a reference (noted by $Prev(n, b)$) towards the optimal solution of the previous time instants (the element $S(n-1, b-\beta_{\max})$ of $\mathbf{S}$). Finally, element $S(N, B)$ will have the optimal aggregate relevancy for the overall time period $\mathcal{T}$ and by tracing the values of matrix $\mathbf{Prev} = [Prev(n, b)]$, we can construct the optimal selection masks to achieve the maximum aggregate relevancy. For instance, $Prev(N, B)$ will contain the budget that should be allocated for the previous $n-1$ time instants and $S\big(N-1, Prev(N, B)\big)$ will be the maximum aggregate relevancy for instants 1 to $N-1$. Then, the optimal selection mask at time slot $N-1$ will be $\boldsymbol{I}^{Prev(N,B)}_{N-1}$, and so on. Note that the optimal selection matrix $\mathbf{I}$ will consist of the N optimal selection masks for times $n \in \mathcal{N}$.

D. Complexity

The time complexity and memory requirements of Algorithm 2 can be calculated based on the complexity of Algorithm 1 and the additional matrix operations of the second level of the optimization algorithm. Algorithm 1 has a worst case complexity of $O(K^2B)$ [3] and is repeated once for each time instant n. Then, lines 6–13 of Algorithm 2 are independent of the number of providers and only depend on the total budget B and the number of time instants N. Hence the worst case complexity of Algorithm 2 is $O(K^2BN)$.

The memory requirements of the algorithm can be calculated as follows. The N runs of Algorithm 1 need to store all the $\mathbf{V}^n$ matrices of total size $N \times K \times B$, and the optimal selection masks $[\boldsymbol{x}^K_{n,b}]_{N\times B\times K}$. Then, the additional matrix manipulations of Algorithm 2 need to store matrices $\mathbf{F}_{N\times B}$ and $\mathbf{S}_{N\times B}$, the optimal selection mask of size K for each of their cells, and matrix $\mathbf{Prev}_{N\times B}$.

In case that the greatest common divisor gcd of all the provider costs c^k_n and the available budget B is larger than 1, the execution time of the algorithms can improve by iterating through the budget $b \in \mathcal{B}$ in increments of size gcd.

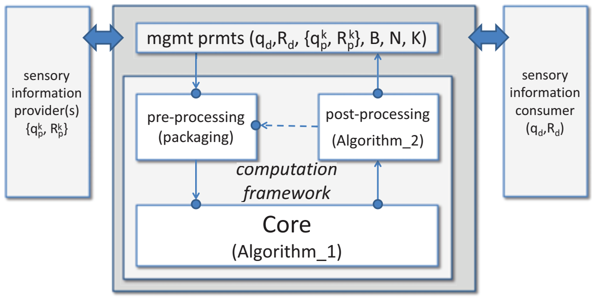

Fig. 4. Functional view of the computation framework

E. A functional view of the computation framework

We close this section by providing a functional view of the computation framework in this paper, see Fig. 4. The framework uses various management input parameters collected from the consumer and providers, such as the QoI functions, B, N, etc., to manage the selection of providers; as noted earlier, the collection process itself is outside the scope of this paper. The framework packages (pre-process) these parameters and entities to drive executions of Algorithm 1 which seats at the core of the computation framework. The collections of outputs from repeated uses of Algorithm 1 is then used by Algorithm 2 to produce the set of recommended spatiotemporally relevant providers. Note that we do not preclude an iterative process (through the dashed arrow) prior to producing the final recommendation.

We mark each block in the framework with a generic name as well (pre-/post-processing, and core) as we will present alternative realizations of their contents to produce faster but approximate solutions instead next. The functional view in Fig. 4 will help us maintain a coherent view of these alternatives and their relationships.

IV. Accelerating the Provider Selection

On occasion it might be necessary to accelerate the provider selection process, e.g., with increasing number of providers, at the expense of optimality. We describe three such acceleration alternatives next; their evaluation is presented later in Section VI.

A. Independent per slot optimization

If the budget allocations b_n at the decision points $n \in \mathcal{N}$ were predetermined, then, clearly, these budget allocations will be outside the purview of the general optimization problem $\mathbf{\Pi}_0$. In this case, we could avoid the extra calculations of Algorithm 2, and instead solve a sequence of independent optimization problems for each decision time n, i.e., have the following problem:

Problem $\mathbf{\Pi}_1$: Given a collection of budget allocations b_n, $n \in \mathcal{N}$, with $\sum_{n\in\mathcal{N}} b_n = B$, find the provider selection matrix $\mathbf{I}_{K\times N}$ that maximizes $r_{st}\big(q_p^{\mathbf{I}}; q_d\big)$, such that for each $n \in \mathcal{N}$:

$$\sum_{k=1}^{K} I(k,n)c_n^k \le b_n, \text{ where } I(k,n) \in \{0,1\}\ \forall(k,n). \quad (5)$$

Compared with problem $\mathbf{\Pi}_0$, $\mathbf{\Pi}_1$ runs N times Algorithm 1 only to get the optimal selection mask $\boldsymbol{x}_{n,b_n}^K$ at each time n (which will become the n^{th} column of matrix $\mathbf{I}$) and saves the time and memory requirements of Algorithm 2 at the expense of a reduction in the overall spatiotemporal relevancy achieved. Therefore, the time and memory requirements of problem $\mathbf{\Pi}_1$ are the same as for Algorithm 1.

There could be numerous budget partitioning policies which may be based on actual needs and constraints (e.g., the city agency may have both daily (b_n) and weekly (B) operating constraints), accounting conveniences, and so on. These policies could depend on n, other b_n's (i.e., have "memory"), etc. In the simplest case, the budget B could be partitioned uniformly across the time horizon, and hence $b_n = B/N$ for all n. With such a memoryless budget assignment, any unused portion of the budget at each slot will go wasted. This may not be desirable though and instead one may want to rollover the unused budget to subsequent slots, e.g., at the next slot or along all remaining slots, to reduce the unused portion of the budget. As an example, let the provider selection mask at time n be $\mathbf{I}_n = \{I(1,n),\ldots,I(K,n)\}$. Then the budget slack due to this selection will be $b'_n = b_n - \sum_{k=1}^{K} I(k,n)c_n^k$. If this slack is to be distributed equally across the remaining slots and taken advantage when provider selection decisions are made at time $(n+1)$, then:

$$b_{n+1} \overset{def}{=} b_{n+1}^{new} = b_{n+1}^{old} + \frac{b'_n}{N-n}, \quad (6)$$

where the designations "old" and "new" apply to the old budget available for decision instant $(n+1)$ (from budget slack assignments made prior to time n) and the updated budget at this time instant, respectively. We use this case later in our numerical evaluations.

With regard to the functional view in Fig. 4, this approximation implements a simplified post-processing operation as it practically bypasses it. On the other hand, the pre-processing implements the budget partitioning (or, allocation) policy, such as based on daily expense constraints.

B. Boolean relaxation

Problem $\mathbf{\Pi}_0$ is an integer programming problem that is NP-hard as a generalization of the knapsack problem. However, the problem can be easily solved once the requirement for deriving binary (0-1) values for the selection matrix is relaxed. *Boolean relaxation* [15] is a common technique with optimization problems that involve boolean (i.e., binary) variables. For an optimization variable x by relaxing the constraint from $x \in \{0,1\}$ to $x \in [0,1]$, we convert the problem into a continuous optimization problem that is in general easier to solve, particularly, if the resulting relaxed problem is convex. Hence, in this case, the problem becomes:

Problem $\mathbf{\Pi}_2$: Find the provider selection matrix $\mathbf{I}$ that maximizes $r_{st}\big(q_p^{\mathbf{I}}; q_d\big)$, such that:

$$\sum_{n=1}^{N}\sum_{k=1}^{K} I(k,n)c_n^k \le B, \text{ where } 0 \le I(k,n) \le 1\ \forall(k,n). \quad (7)$$

Algorithm 3 – Myopic Algorithm

Initialize $I(k,n) = 0\ \forall k \in \mathcal{K}, n \in \mathcal{N}$
Calculate $r_{st}\big(q_p^k; q_d\big)$, $k \in \mathcal{K}$, and sort in descending order—let $idx(m)$ be the index of ordered provider m;
Set $b = B$; $m = 1$;
while $(b > 0\ \&\ m \leq K)$ **do**
  if $b \geq c_{idx(m)}$ **then**
    $I(idx(m), n) = 1$ for $n \in \mathcal{N}$;
  end if
  $b = b - c_{idx(m)}$; $m = m + 1$;
end while

Problem $\mathbf{\Pi}_2$ is a convex continuous optimization problem since the constraints are linear functions of the optimization variables $I(k,n)$, for $k \in \mathcal{K}$ and $n \in \mathcal{N}$, and the objective function is concave as a summation of linear functions of the optimization variables. Therefore, problem $\mathbf{\Pi}_2$ can be easily solved with gradient based optimization algorithms that require linear time [16]. Note that there is an appealing interpretation of the derived solution in this case. The selection $I(k,n) = \alpha \in [0,1]$ implies that the consumer decides at time n to bind with (and pay) provider k only for the fraction α of slot n.

With regard to the functional view in Fig. 4, the gradient, for example, algorithm becomes the core algorithm and post-processing is again bypassed (or, equivalently, absorbed by the core algorithm).

C. Myopic algorithm

In circumstances that a fast decision about selecting an appropriate set of providers is of the highest priority, we may resort to myopic algorithms that select providers based on a rather limited view of the problem. We have considered a simple myopic algorithm where the spatiotemporal relevancy $r_{st}\big(q_p^k; q_d\big)$ of each provider k over the time horizon $\mathcal{T}$ is calculated first. Then, the providers are ordered according to their respective relevancy $r_{st}\big(q_p^k; q_d\big)$. Based on this relevancy and the total cost $c^k = \sum_{n\in\mathcal{N}} c_n^k$ for engaging (whenever possible) with provider k during the period $\mathcal{T}$, the top M providers are chosen so that the total cost is as close to B as possible. In case that $\sum_{m=1}^{M} c^m < B$ and $\sum_{m=1}^{M+1} c^m > B$, we may select a provider that is further down in the relevancy-ordered list of providers but with cost low enough to fit under the remaining budget. Algorithm 3 summarizes the myopic algorithm. Regarding its complexity, the calculation of the spatiotemporal relevancy of providers requires $O(K)$ time, the ordering of the providers can be calculated in $O(K \log K)$ time and the selection of the providers is $O(KB)$ in the worst case.

This myopic algorithm will operate well when the providers experience inconsequential overlapping in which case the effects of provider aggregation do not manifest themselves strongly. This is because in this case, the increase of the *aggregate* spatiotemporal relevancy by including a provider will be equal to its individual spatiotemporal relevancy. However, in general, the more the providers overlap, the more discrepancy would be in the computed relevancy with regard to the optimal one which could effect the provider selections.

With regard to the functional view in Fig. 4, the core algorithm calculates the spatiotemporal relevancy using (3), while post-processing orders the providers according to their individual relevancy and selecting the top ones that cumulatively do not violate the budget constraint.

V. Accelerating the Relevancy Assessment

The implementation of DP Algorithm 2 and the three approximation methods presented in Section IV involve multiple evaluations of the integral expressions in (1) and (3), a process which could be computationally taxing. Hence, to further accelerate the relevancy assessment, next we highlight two simplifications that are based on creating QoI function summaries along polygon approximations of the desired and provided regions R_d and R_p^k, $k \in \mathcal{K}$, respectively, which can then leverage standard polygon operations [17], [18].

With regard to Fig. 4, what we present next represents specific implementations of the relevancy calculation subroutine residing within the core module.

As all variables highlighted next relate to a given slot n, $n \in \mathcal{N}$, we will drop n from the notation for brevity as necessary.

A. Polygon intersection: `PolIn`

For each n, the temporal relevancy $r\big(n; q_p^{\boldsymbol{x}}, q_d\big)$ needs to be computed each time a candidate selection mask $\boldsymbol{x}$ is considered. To accelerate this computation, we approximate the regions R_d and R_p^k's with polygons and then compute relevancy manipulating these polygons and values assigned to them.

Specifically, let the coverage region R_p^k of provider k (at slot n) be partitioned into L_p^k polygons[3] and associate with polygon $m_p^{k,l}$, $l \in \{1, \ldots, L_p^k\}$, the *single* QoI value $q_p^{k,l}$. Likewise, let the region R_d desired by the consumer be partitioned into L_d polygons and associate with polygon m_d^l, $l \in \{1, \ldots, L_d\}$, the QoI value q_d^l. The QoI values $q_p^{k,l}$ and q_d^l could represent, for example, the mean, median, max, min, etc., derived from the actual range of the respective QoI functions $q_p^k(\cdot)$ and $q_d(\cdot)$ over the the pertinent polygons.

Then, for a candidate provider selection mask $\boldsymbol{x}$, we aggregate all the polygons related with $\boldsymbol{x}$ by calculating the intersections and subtractions for all the combinations of the polygons from the providers in $\boldsymbol{x}$. If $L_p^{\boldsymbol{x}}$ is the resulting number of polygons, we associate with polygon $m_p^{\boldsymbol{x},l}$, $l \in \{1, \ldots, L_p^{\boldsymbol{x}}\}$, the QoI value $q_p^{\boldsymbol{x},l}$. This value is obtained by applying the QoI provider aggregation operator $h(\cdot)$ (e.g., see Sect. III-A) on the QoI values of all the provider polygons that were involved in the creation of $m_p^{\boldsymbol{x},l}$.

In similar fashion, next we intersect these $L_p^{\boldsymbol{x}}$ provider polygons with the L_d consumer polygons to produce a final number $S_{d\cap\boldsymbol{x}}$ of polygons. We associate polygon $m_{d\cap\boldsymbol{x}}^s$, $s \in \{1, \ldots, S_{d\cap\boldsymbol{x}}\}$, with the QoI value $q_{d\cap\boldsymbol{x}}^s$, which is calculated by applying the VoI function $v(\cdot)$ (e.g., eq. (2)) on the QoI

[3]These regions can be approximated by a set of polygons of arbitrary number of edges, e.g., using *polygonal modeling* from computer graphics [19].

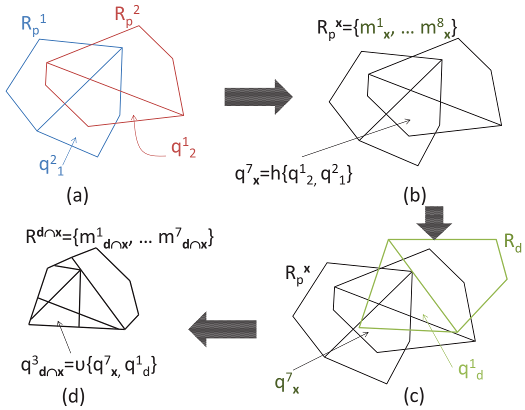

Fig. 5. Example of the `PolIn` method (for slot n)

values of the aggregated provider and consumer polygons that were involved in the creation of polygon $m^s_{d\cap \boldsymbol{x}}$.

Fig. 5 illustrates the described procedure for $K = 2$ providers, $L^1_p = L^2_p = 2$, and $L_d = 2$. The aggregation operation of the 4 provider polygons produced $L^{\boldsymbol{x}}_p = 8$ polygons, whereas the subsequent intersection of the aggregated provider and desired polygons produced $S_{d\cap \boldsymbol{x}} = 7$ polygons.

Finally, the temporal relevancy $r\big(n; q^{\boldsymbol{x}}_p, q_d\big)$ is calculated according to the following expression ($\texttt{plgAr}\{\cdot\}$ stands for the area of the respective polygon):

$$r_s\big(n; q^{\boldsymbol{x}}_p, q_d\big) \approx \frac{\sum_{s=1}^{S_{d\cap \boldsymbol{x}}} q^s_{d\cap \boldsymbol{x}} \cdot \texttt{plgAr}\{m^s_{d\cap \boldsymbol{x}}\}}{\sum_{l=1}^{L_d} q^l_d \cdot \texttt{plgAr}\{m^l_d\}}. \quad (8)$$

Apart from the `PolIn` approximation above in computing the temporal relevancy, the rest of the procedure for solving problem $\boldsymbol{\Pi}_0$ using algorithms 1 and 2 remains identical.

B. *Mean value polygon intersection:* `mvPolIn`

To further increase the computation speed, we have also considered the `mvPolIn` simplification of `PolIn` which reduces the number of polygons $L^{\boldsymbol{x}}_p$ considered and, hence, the related computations of the previous subsection. Due to space limitations, we only mention here the following: Upon the provider aggregation according to the candidate selection mask $\boldsymbol{x}$, `mvPolIn` summarizes the QoI values of *all* the provider polygons related to $\boldsymbol{x}$ into a *single* QoI value that is then applicable to all these provider polygons. This common QoI value is calculated as a weighted average of the QoI values of each provider polygon with the polygon areas acting as the weights. We include results for the `mvPolIn` acceleration technique in the next section.

VI. Performance Evaluation

The DP Algorithm 2 and the other alternative methods presented in the previous sections were simulated in a MATLAB environment to study their performance vs. time-complexity trade-offs.

The various methods were evaluated through simulations in multiple randomly generated experiments for $N = 8$ time instants. At every instance of these experiments, a number of Gaussian mixtures, representing the desired (q_d) and provided (q^k_p) QoI functions, where randomly scaled and placed on the plane. Fig. 6 shows an example time instant where the consumer (e.g., the city agency) has to determine the spatiotemporal relevancy of $K = 8$ providers.

The QoI shapes for all N instants where then fed to the various optimization methods where each of them determined the most appropriate set of providers according to the budget constraints, while measuring the necessary optimization time of each method. The above experiments were repeated for number of providers $K \in [4, 11]$, corresponding budgets $B \in \{75, 85, 98, 110, 115, 130, 138, 150\}$, and fixed provider costs $\boldsymbol{c}_n = [2, 3, 4, 6, 3, 4, 2, 5, 4, 2, 1]$, i.e., $c^1_n = 2$, $c^2_n = 3$, etc., $\forall n$.

Fig. 7 shows the spatiotemporal relevancy of each method with respect to the number of providers K. As expected, Algorithm 2 and the Boolean relaxation method lead to almost identical values of spatiotemporal relevancy, which is also higher than all other methods. Then, the `PolIn` method results to values that are relatively close to the optimal (as calculated by Algorithm 2). More specifically, the value of the spatiotemporal relevancy calculated by `PolIn` was on average 5% below the optimal solution. On the other hand, the performance of `mvPolIn` and the myopic methods were significantly suboptimal, with the former being on average 20% suboptimal and the latter achieving around 26% worse performance. This high percentage of error is observed due to the several approximations each of the methods makes to accelerate their execution (see discussion of Fig. 8).

However, the accuracy of spatiotemporal relevancy calculations are only one side of the coin. The execution time of each of the methods is also important to determine their usability. Fig. 8 shows the execution times of the 6 methods. According to this, Algorithm 2 is the slowest one as a result of the numerical integrations necessary to calculate the spatial relevancy of every candidate selection mask at each time instant. The Boolean relaxation method converges to the optimal solution in relatively short time as it solves a continuous optimization problem for which there are efficient linear gradient based solution algorithms [16]. The independent per slot optimization algorithm is faster than Algorithm 2

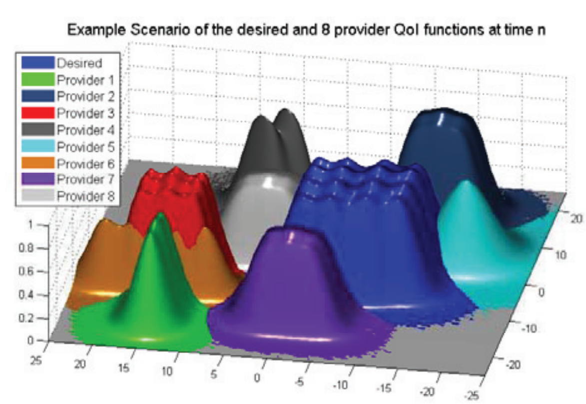

Fig. 6. An example of the QoI functions a consumer (blue) and 8 providers

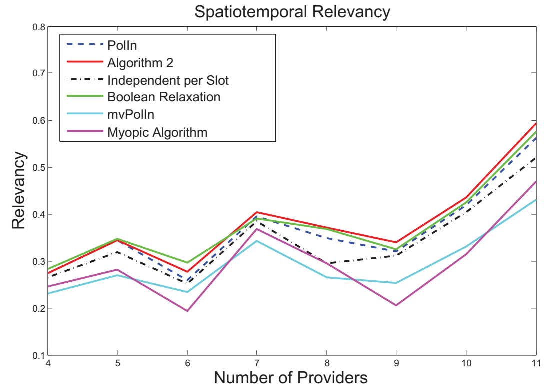

Fig. 7. Aggregate spatiotemporal relevancy of various methods

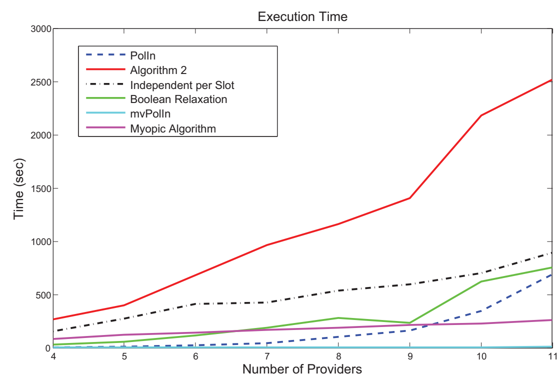

Fig. 8. Execution time of Algorithm 2 and approximation methods using a 2.4 GHz dual core PC with 4 GB or RAM.

since it only executes Algorithm 1 N times. Moreover, it can make more efficient use of parallel processing capabilities of current computer systems and the acceleration method using the *greater common divisor* of the provider cost and the available budget (see Section III). Then, the myopic algorithm needs almost linear execution time with respect to the number of providers, and, finally, `mvPolIn` is the fastest method due to the significant simplifications done in the calculation of the spatial relevancy.

It is therefore evident that each algorithm offers a different trade-off between accuracy and time efficiency that might make some algorithms more suitable than others depending on the requirements (time and accuracy) of an application.

VII. Conclusion

Motivated by evolutionary trends in smart planet applications, this paper considered a novel computational framework for managing the selection of sensory information providers based on the spatiotemporal relevancy between the information sought by a consumer and provided by the sensory information providers. We have considered dynamic cases where relevancy changes with time due to the mobility, or changes in the interests or capabilities of the information consumers and providers.

We have defined and provided a metric for spatiotemporal relevancy and used these to formulate an integer programming optimization problem for selecting the most relevant set of providers over a period of time. We have developed a pseudo-polynomial time solution to the problem using a two-level dynamic programming algorithm. We also presented a number of approximate problem formulations and solutions that identify relevant providers faster at the expense of attaining suboptimal levels of relevancy. The optimal solution and the proposed approximations were evaluated through simulations.

Future work considerations include operational aspects, such as QoI advertisements, incorporating more advanced provider mobility models, consideration of additional provider aggregation operators $h(\cdot)$ and VoI functions $v(\cdot)$, as well as trade-offs between system dynamics, distributed operation (e.g., as per Fig. 2.c) and computational efficiency.

References

[1] N. Gershenfeld, R. Krikorian, and D. Cohen, "The Internet of Things," *Scientific American*, October 2004.

[2] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *World Sensor Web Workshop (in ACM Sensys'06)*, Boulder, CO, USA, Oct. 31 2006.

[3] G. Tychogiorgos and C. Bisdikian, "Selecting relevant sensor providers for meeting "your" quality information needs," in *IEEE Mobile Data Management*, Lulea, Sweden, June 2011.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *in IEEE INFOCOM*, 2001.

[5] C. fu Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *in WSNA*. ACM Press, 2003.

[6] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," *Pervasive and Mobile Comp.*, June 2008.

[7] N. Pelekis, B. Theodoulidis, I. Kopanakis, and Y. Theodoridis, "Literature review of spatio-temporal database models," *The Knowledge Engineering Review*, September 2004.

[8] R. Devillers, Y. Bdard, and R. Jeansoulin, "Multidimensional management of geospatial data quality information for its dynamic use within GIS," *Photogrammetric Engineering & Remote Sensing*, February 2005.

[9] B. Yu and G. Cai, "A query-aware document ranking method for geographic information retrieval," in *4th ACM Workshop on Geographical Information Retrieval (GIR'07)*, Lisbon, Portugal, Nov. 9, 2007.

[10] P. Basu, A. Nadamani, and L. Tong, "Extremum tracking in sensor fields with spatio-temporal correlation," in *MILCOM'10*, Oct. 31–Nov. 3 2010.

[11] A. Deligiannakis and Y. Kotidis, "Geosensor networks," S. Nittel, A. Labrinidis, and A. Stefanidis, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, ch. Exploiting Spatio-temporal Correlations for Data Processing in Sensor Networks.

[12] C. Bisdikian, L. M. Kaplan, M. B. Srivastava, D. J. Thornley, D. Verma, and R. I. Young, "Building principles for a quality of information specification for sensor information," in *12th Intl Conf. on Information Fusion (FUSION'09)*, Seattle, WA, USA, July 2009.

[13] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S-a publish/subscribe protocol for wireless sensor networks," in *2nd Workshop on Intelligent Networks: Adaptation, Communication & Reconfiguration (IAMCOM'08)*, Bangalore, India, Jan. 10, 2008.

[14] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, 1990.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[16] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[17] R. Bulbul and A. U. Frank, "Intersection of nonconvex polygons using the alternate hierarchical decomposition," in *Geospatial Thinking*, ser. Lecture Notes in Geoinformation and Cartography, M. Painho, M. Y. Santos, and H. Pundt, Eds. Springer Berlin Heidelberg, 2010.

[18] J. O'Rourke, C.-B. Chien, T. Olson, and D. Naddor, "A new linear algorithm for intersecting convex polygons," *Computer Graphics and Image Processing*, pp. 384–391, 1982.

[19] M. Russo, *Polygonal Modeling: Basic and Advanced Techniques*. Worldware Publishing Inc., 2006.