Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

SUBSPACE COMMUNICATION DRIVEN SEARCH FOR HIGH DIMENSIONAL OPTIMIZATION

Logan Mathesen
Kaushik Keezhnagar Chandrasekar
Xinsheng Li
Giulia Pedrielli
K. Selçuk Candan

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
699 S Mill Ave
Tempe, AZ 85283, USA

ABSTRACT

Global optimization techniques often suffer the curse of dimensionality. In an attempt to face this challenge, high dimensional search techniques try to identify and leverage upon the *effective*, lower, dimensionality of the problem either in the original or in a transformed space. As a result, algorithms search for and exploit a projection or create a random embedding. Our approach avoids modeling of high dimensional spaces, and the assumption of low effective dimensionality. We argue that effectively high dimensional functions can be recursively optimized over sets of complementary lower dimensional subspaces. In this light, we propose the novel *Subspace COmmunication for OPtimization (SCOOP)* algorithm, which enables intelligent information sharing among subspaces such that each subspace guides the other towards improved locations. The experiments show that the accuracy of SCOOP rivals the state-of-the-art global optimization techniques, while being several orders of magnitude faster and having better scalability against the problem dimensionality.

1 Introduction

Many applications require optimization within high-dimensional problem spaces. A possible approach is to treat the high-dimensional problem as the optimization of a non-convex black-box function, which involves solving $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathscr{X}} f(\mathbf{x})$, where $\mathscr{X} \subset \mathbb{R}^d$ is a high-dimensional solution space and $f : \mathscr{X} \to \mathbb{R}$ is a non-linear non-convex function. Bayesian Optimization has witnessed an incredible growth. Nevertheless, the optimization of high-dimensional, highly non-linear, functions is still considered a hard problem (Rolland et al. 2018; Hoang et al. 2017). A common approach to improve the search efficiency is to note that for certain problem classes, most dimensions do not have a significant impact on the objective function (Figure 1), (Djolonga et al. 2013; Wang et al. 2016; Bergstra and Bengio 2012). This is commonly referred to as the *assumption of low effective dimensionality*. Many methods exploit this assumption to reduce the problem dimension for subsequent lower dimensional surrogate modeling and search (Carpentier and Munos 2012; Chen et al. 2012): if one can assume there exist some combination of dimensions that produces an effective subspace such that the objective does not change along any subspace which is orthogonal to it, then search can be performed efficiently. For instance, algorithms such as SI-BO (Djolonga et al. 2013) aim at learning and exploiting low effective dimensionality (Carpentier and Munos 2012; Chen et al. 2012). A different perspective is to exploit lower effective dimensionality without knowing which dimensions are important, in this second class of algorithms we find random search (RS) (Bergstra and Bengio 2012) and Random Embedding Bayesian Optimization (REMBO) (Wang et al. 2016), that includes a warped kernel

implementation (Binois et al. 2015) and a discussion on the choice of low-dimensional embedding domain (Binois et al. 2017).

However, most of machine learning hyper-parameter optimization problems are *inherently* highly dimensional (differently from physical problems). In these cases, dimensionality reduction may not return the expected decrease in problem size. It is in light of these applications that we propose the novel *Subspace COmmunication for OPtimization (SCOOP)* method for global optimization of non-linear functions. SCOOP does not make any assumption on low or reduced effective dimensionality (Figure 2). In fact, by modeling and optimizing over sequential sets of complementary subspaces, SCOOP enables intelligent information sharing among these such that each subspace randomly guides the others towards improved locations. We present three methods of information sharing, each of which makes a different use of the subspace related and shared information. Experiments on several function categories, with different characteristics, show that the accuracy of SCOOP rivals the state-of-the-art in existing high-dimensional global optimization techniques.

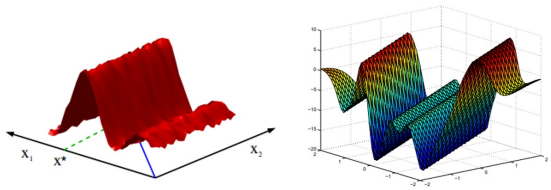


Figure 1: Function examples, from (Wang et al. 2016) and (Djolonga et al. 2013), with low effective dimensionality, a common assumption in high dimensional optimization.

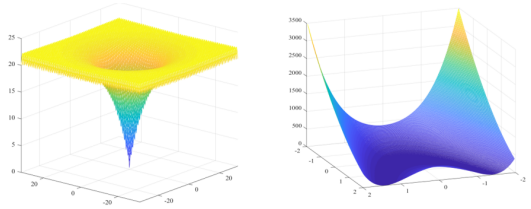


Figure 2: Function examples with high effective dimensionality – *this is the environment that we aim to optimize over*.

2 Background & Related Work

2.1 Problem Formulation

In this work, we will assume that the nonlinear optimization problem is defined over a compact solution set $\mathscr{X}$. The considered objective function $f : x \in \mathscr{X} \subset \mathbb{R}^d \rightarrow f(\mathbf{x}) \in \mathbb{R}$ is deterministic in nature, and can only be evaluated by running simulation at the desired location. The objective is to develop an efficient search algorithm that finds the global minimum of $f : \mathscr{X} \rightarrow \mathbb{R}$, namely:

$$\begin{aligned} P &: \min f(x) \\ \text{s.t.} \quad & x \in \mathscr{X} \end{aligned}$$

2.2 Relevant Literature

Bayesian Optimization (BO) is a highly recognized framework for optimizing expensive, possibly noisy, black-box functions. It offers a clean, concise, and formal approach to explicitly capturing uncertainty about the objective function behavior through the use of prior and posterior model updates via sequential data observations (Močkus 1975; Brochu et al. 2010; Shahriari et al. 2016). BO based samplers try to achieve balancing of exploitation and exploration at each sample in the search, outperforming other algorithms on many challenging benchmark functions (Jones 2001), and gaining a strong foothold in the machine learning community (Snoek et al. 2012; Marchant and Ramos 2012). A common model (or prior distribution) for BO–especially for continuous functions–is the Gaussian process (GP) due to its flexibility and closed form expressions. While in this work we make use of the BO framework with GP prior, other notable past proposed optimization methods include: grid and random search (Bergstra and Bengio 2012;

Zabinsky et al. 1993), randomly restarted direct searches (Molvalioglu et al. 2010), as well as random forests (Hutter et al. 2011) and bandit based Gaussian process modeling methods (Srinivas et al. 2009).

Despite the success of BO with GP (Jones et al. 1998; Hoffman et al. 2011), the general approach is limited to a moderate dimensions of about 10 (Wang et al. 2016). This limitation is due to three intertwined reasons: (a) the GP learning effort grows cubically with the number of observations as an inversion of a dense covariance matrix is required; (b) as the dimensionality increases, the number of observations needed to provide good coverage of the solution space increases exponentially; and (b) the computational challenge of maximizing acquisition functions generally scales exponentially with the number of dimensions. In fact, the cubic scaling in (a) and the required sequential sampling derived from (c) renders parallelism generally ineffective. For these reasons, scaling of BO to high dimensional spaces is still an open research question.

Recently, additive Gaussian process (add-GP) modeling was proposed to extend BO into high dimensional spaces while avoiding the assumption of low effective dimensionality (Kandasamy et al. 2015). The key structural assumption underlying this approach is that f decomposes as:

$$f(\mathbf{x}) = \sum_{i=1}^{M} f^{(i)}(\mathbf{x}^{(i)}) \tag{1}$$

where $\mathbf{x}^{(i)} \in \mathscr{X}^{(i)}$, and $\mathscr{X}^{(i)}$ are lower dimensional subspaces such that the several $f^{(i)}(\mathbf{x}^{(i)})$ can be independently modeled as GP's and *added* together to recover the full dimensional model, addressing issue (a) above. An additive kernel based acquisition function can then be assessed over $\mathscr{X}$ and can subsequently be optimized by maximizing over each subspace to recover sampling in $\mathscr{X}$, thus addressing issue (c). A great deal of work has been conducted in determining the appropriate decompositions, $\mathscr{X}^{(i)}$, to use in additive GP optimization, including: structured kernel learning (Wang et al. 2017), factor graph methods for disjoint subspaces (Hoang et al. 2017), MCMC methods (Gardner et al. 2017), overlapping groups with Gibbs sampling based graph learning for efficient acquisition function optimization (Rolland et al. 2018), overlapping groups with Fourier Features approximation for efficient acquisition function optimization (Mutny and Krause 2018).

Note, for the additive approach to be effective, a tractable likelihood function is needed to enable estimation of a high dimensional global model. Additive-GP models require strict assumptions about the separable nature and structure of the function. Our approach we avoids estimation of a high dimensional global model, and thus make no assumptions on the additive nature of the objective function. Two recent approaches that do not use additive modeling assumptions include, *LineBO* (Kirschner et al. 2019) restricts the problem to sequentially solving one dimensional subproblems, converging when the problem is strongly convex, and *BOCK* (Oh et al. 2018) which addresses the boundary issue in high dimensional optimization (Swersky 2017) by transforming the search space ball geometry with a cylindrical transform.

2.3 Contribution

In contrast to additive Gaussian process modeling and random embedding techniques, in this paper, we propose SCOOP which leverages multiple lower dimensional subspaces and *executes local BO directly over these lower dimensional spaces* – the outcomes are then iteratively stitched through information sharing to obtain the overall solution. This partition-stitch approach has been used succesfully in other problem domains, such as high-dimensional tensor analysis (Li et al. 2018), but its use in the domain of optimization requires *a novel framework of information sharing and learning between complementary subspaces*. Sequentially optimizing over complementary lower dimensional subspaces, and communicating information between them, we adaptively project onto the original space avoiding all high dimensional modeling *and* the assumption of low effective dimensionality. This unique subspace decomposition enables the sharing of information amongst subspaces such that complementary subspaces can guide one another towards improved locations.

Our representation contrast the overlapping additive Gaussian process modeling used in (Rolland et al. 2018), that models several subcomponents of the domain and adds them together as in (1), e.g.,

$f(\mathbf{x}) = f^{(1)}(x_1) + f^{(2)}(x_2, x_3) + f^{(3)}(x_3)$. Whereas we condition certain variables and optimize directly over the resulting subspace, e.g., over $\{\mathbf{x} \in \mathscr{X} : x_3 = a\}$, and avoids modeling and assumptions of $f(\mathbf{x})$. We differ from the works (Binois et al. 2017; Binois et al. 2015; Wang et al. 2016) as we experiment over test functions of full dimension, testing problems where all dimensions contribute to the objective function.

Intuitively, SCOOP adds significant degrees of freedom to the additive approach by only modeling subspaces, and avoiding estimation of a high dimensional global model altogether, essentially decoupling the modeling and partitioning problem. By no longer modeling the high dimensional space, there is no longer a need for additive or effective lower dimension assumptions to reduce the complexity of the problem. Thus SCOOP is afforded a great deal of flexibility in the range of problems it can address, and we note that while not explored in this contribution, decomposition methods from (Wang et al. 2017; Hoang et al. 2017; Rolland et al. 2018) can still fit within the overall SCOOP optimization framework to identify optimal subspace partitions and communication patterns.

3 Subspace COmmunication for OPtimization (SCOOP)

Figure 3 outlines the proposed *Subspace COmmunication for OPtimization (SCOOP)* algorithm. Starting from the original space, a "slicing operation" is performed to decompose the space into n_{sub} subspaces ($SS_i, i = 1, \ldots, n_{sub}$, with $n_{sub} = k$ in Figure 3) each with a collection of non-shared dimensions I_{ss_i} and shared dimensions $\mathbf{x}_i$, such that $\cap_{i=1}^{k} I_{ss_i} = \emptyset$ and $\cup_{i=1}^{k} I_{ss_i} \cup \mathbf{x}_i = \mathscr{X}$. In each of the subspaces, the "hidden" dimensions are set to a value that is produced by the information sharing algorithm. In fact, optimization is performed locally, i.e., in each subspace, and the identified solution is passed intelligently to other complementary subspaces that will consider whether to move towards directions recommended by the communicating subspace optimizations, by changing the value of their hidden dimensions.

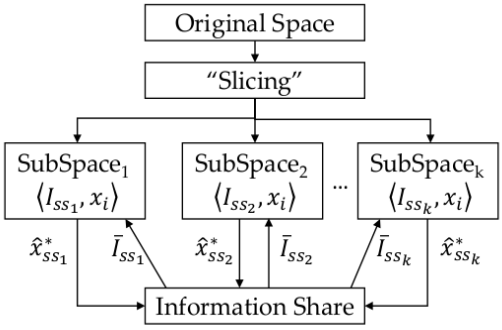


Figure 3: SCOOP Approach

In the following, we define the main components of SCOOP, detailing the initialization (Section 3.1), the subspace optimizer (Section 3.2), and the information sharing strategies (Section 3.3). The SCOOP pseudo code is reported in Algorithm 1.

3.1 Algorithm Inputs and Initialization

Given a d-dimensional non linear non-convex objective function to minimize, we assume there exist standard upper and lower bound (box) constraints on each dimension or decision variable, i.e., $x_i \in [LB_i, UB_i]\ \forall i = 1, ..., d$. The remaining user inputs include:

- *total function sampling budget, B*, total objective function evaluations allowed during the search;
- *number of subspaces*, n_{sub}, is the number of (complementary) subspaces SCOOP considers. In this manuscript, we project by setting a subset of hidden dimensions to constant values (extensions to subspace creation based upon more complex projection schemes could be considered);
- *number of "hidden" dimensions*, n_{hid}, to be held constant to obtain each subspace, n_{hid} then implies the number of active dimension per subspace as $SS_{dim} = d - n_{hid}$;
- *initial sampling budget*, b_{init}, to initialize the subspace search at each new subspace projection;
- *search budget*, b_{opt}, available number of evaluations for each optimization iteration in the same subspace.

Algorithm 1 Subspace COmmunication based OPtimization

Input: $f(\mathbf{x})$, $a(\mathbf{x})$, B, n_{sub}, n_{hid}, b_{init}, b_{opt}, information sharing strategy
Output: $\hat{\mathbf{x}}^*, f(\hat{\mathbf{x}}^*)$: the best sampled full dimensional location

Initialize: Randomly assign all hidden dimension values for each of subspaces, there will be a total of $(n_{hid} \times n_{sub})$ hidden dimensions assigned.
while $B > 0$ **do**
Execute Subspace Optimizations
  for $i = 1, ..., n_{sub}$ **do**
    Create initial space filling design with b_{init} points over active/free dimensions of subspace i: $\mathbf{X}_{0i} \in \mathbb{R}^{(b_{init} \times d - n_{hid})}$
    Create $\mathbf{X}_{0i}^{full}$ by augmenting each design point in $\mathbf{X}_{0i}$ with current subspace hidden dimension values: $\mathbf{X}_{0i}^{full} \in \mathbb{R}^{(b_{init} \times d)}$
    Sample objective function at each of these points: $f(\mathbf{X}_{0i}^{full}) \in \mathbb{R}^{(b_{init} \times 1)}$. Update $B \leftarrow B - b_{init}$, break if $B \leq 0$
    Fit a subspace GP to $\mathbf{X}_{0i}, f(\mathbf{X}_{0i}^{full})$
    for $j = 1, ..., b_{opt}$ **do**
      Discover $\mathbf{x}_{next} = \arg\max_{\mathbf{x} \in \mathscr{X}} a(\mathbf{x})$: $\mathbf{x}_{next} \in \mathbb{R}^{(1 \times d - n_{hid})}$
      Augment $\mathbf{x}_{next}$ with hidden dimension values: $\mathbf{x}_{next}^{full} \in \mathbb{R}^{(1 \times d)}$
      Update $\mathbf{X}_{ji} \leftarrow \mathbf{X}_{(j-1)i} \cup \mathbf{x}_{next}$ and $\mathbf{X}_{ji}^{full} \leftarrow \mathbf{X}_{(j-1)i}^{full} \cup \mathbf{x}_{next}^{full}$
      Sample $f(\mathbf{x}_{next}^{full})$. Update $B \leftarrow B - 1$, break if $B = 0$
      Fit updated subspace GP to $\mathbf{X}_{ji}, f(\mathbf{X}_{ji}^{full})$
    end for
  end for
Execute Subspace Information Sharing: Update Hidden Dimension Values
  for $i = 1, ..., n_{sub}$ **do**
    for $j = 1, ..., n_{hid}$ **do**
      For hidden dimension j of subspace i, determine the complementary subspace with a shared active parameter and free dimension corresponding to subspace i's j^{th} hidden dimension
      Update value of subspace i's hidden dimension j by sharing information with the identified complementary subspace
    end for
  end for
end while

Finally, SCOOP takes as input an information sharing strategy, which describes how the optimization outcome in one subspace informs the search in complementary subspaces. In Section 3.3, we give three alternate strategies, which differ for the *level of trust* in the subspace optimization.

3.2 Subspace Bayesian Optimization

In SCOOP, each subspace optimization is carried out through iterative Bayesian Optimization (BO). Hence, sampling decisions (i.e., parameter instances for which the function f is evaluated) are determined by optimizing a surrogate objective function. The surrogate is estimated from all $\{f(\mathbf{x}_i)\}_{i=1}^{k}$ evaluations up to iteration k (Močkus 1975), effectively considering *all* previously learned input-target black-box pairs. This requires two main components: (1) a *prior surrogate form* to model the function over the subspaces, and (2) an *acquisition function* to drive subsequent sampling decisions.

Prior BO forms employed range from radial basis functions to complex neural networks (Snoek et al. 2015); amongst the most common forms are Gaussian processes (GP's), which are used in this research. Bayesian optimization in this set-up has a rich body of literature which we refer the interested reader to (Wang et al. 2016; Snoek et al. 2012; Rasmussen 2004; **?**; Brochu et al. 2010).

3.3 Information Sharing Strategies

Note that, while each subspace is modeled with a low dimensional GP and acquisition functions, sample locations are defined in the original *full dimension* space to enable function evaluation. Given any current subspace projection (i.e., a set of hidden dimension values), we recover the *full dimensional* samples from the *low dimensional subspace* by augmenting samples in active subspace dimensions with current subspace hidden dimension values. Starting with a random initialization of the hidden dimensions, SCOOP learns globally optimal subspace hidden dimension values by implementing a *communication scheme between*

subspaces. In fact, communicating information between the subspaces to update hidden dimension values creates new subspace projections allowing the entire full dimensional space to be explored and optimized while only explicitly modeling and optimizing over low dimensional subspaces. With this in mind, information communication and sharing between complementary subspaces becomes the crux of the proposed SCOOP algorithm, as it how the low dimensional subspace projections search the full high-dimensional space. We present three strategies for information sharing among subspaces.

Best Observed Sample Sharing (SCOOP-B) According to this strategy, referred to as SCOOP-B in the following, each subspace optimizer exhausts its subspace optimization budget b_{opt} and returns the location within the subspace with best observed function value. The "local" best solution is shared to all other subspaces, who accept the relevant values and appropriately update their own hidden dimension values.

Due to the purely exploitative greedy nature of this sharing strategy. We conjecture that this direct sharing strategy will be particularly helpful with problems that exhibit pronounced optimum behavior in low dimensional projections.

Marginalized Expected Improvement (SCOOP-E) The second information sharing mechanism was designed to improve the robustness with respect to the dimension(s) shared among subspaces. This robustness is achieved by selecting the value(s) of the hidden subspace dimension(s) as the *maximizer of the posterior expected improvement function marginalized with respect to shared active dimension(s)*. The marginalized expected improvement for subspace j, with shared dimension(s) i is:

$$EI_M(\mathbf{x}, x_i) = \int_{SS_j} EI(\mathbf{x}) \mathrm{d}x_i$$

where x_i represents the potential values of shared dimension i. Integrating, with respect to all shared dimensions i, over the expected improvement $EI(\mathbf{x})$ (Jones et al. 1998) yields a marginalized score for all locations of the non-shared active dimension of SS_j, and the maximizer can then be shared to all other subspaces. Using a shared dimension x_1 results in: $a = \{x_3 : \mathbf{x} = \arg\max_{\mathbf{x} \in SS_2} \int_{SS_2} a(\mathbf{x}) dx_1\}$ and $b = \{x_2 : \mathbf{x} = \arg\max_{\mathbf{x} \in SS_1} \int_{SS_1} a(\mathbf{x}) dx_1\}$, as the respective updates to the hidden dimension values of subspace-1 and subspace-2. We conjecture that this sharing strategy will be especially suited to problems with complex high dimensional interactions as it enables a balanced, explorative and exploitive, search through hidden dimensions, which is tempered against shared active dimension information.

Subspace Link Failure (SCOOP-•L) The previous methods always trust the information shared from other subspaces. To control this aspect, we *interpret subspaces as communicating agents* and use communication failure between subspaces to mimic the theory of hyper-jumps (Wang and Elia 2012). Specifically, prior to every communication and information sharing event, a random number is uniformly drawn. If this random number exceeds a user defined probability of link failure, α, then communication and sharing is completed as intended, otherwise we instead draw a random subspace hidden dimension value. If a "link failure" occurs, then the j^{th} hidden dimension value is drawn from a beta distribution that reflects the density of previous values for hidden dimension j. The hidden dimension value is drawn from: $beta(\gamma, \xi)$, with $\gamma = \max(\frac{|\mathbf{H}_{jl}|}{|\mathbf{H}_{ju}|}, 1)$, $\xi = \max(\frac{|\mathbf{H}_{ju}|}{|\mathbf{H}_{jl}|}, 1)$, where $\mathbf{H}_{jl}$ is the set of all previous values of the j^{th} hidden dimension in the lower half of the j^{th} dimension support; similarly, $\mathbf{H}_{ju}$ are those in the upper half. Thus, we probabilistically bias the next subspace projection to be located in a less explored region. Intuitively asymptotic convergence is yielded as a result of link failure since, as the number of evaluations approaches infinity, each subspace projection is revisited infinitely often; thus the entire solution space has probability strictly greater than zero of being sampled.

We note that *subspace link failure* can be implemented alongside/in addition to the first two strategies, yielding *SCOOP-BL* and *SCOOP-EL*, and is especially suited to problems with trapping local minima as it allows search focus to randomly enter unexplored regions.

4 Experimental Evaluation

Experiment Settings

(a) We first test how information sharing strategies impact algorithm convergence and finite time performance over functions with different unique characteristics; then we study the affects of the amount of link failure, and size (and number) of subspaces in higher dimensional problems. (b) Next we present scalability results against state-of-the-art competitors in problems with effective dimension of 20, 50, and 100, (c) and conclude with a application searching for optimal neural net configurations.

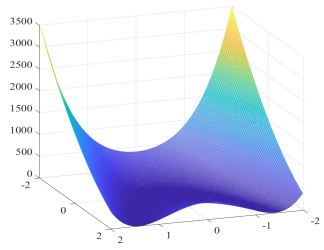


Figure 4: Rosenbrock in $\mathbb{R}^2$; smooth gradients with global minimum hidden by flat valley.

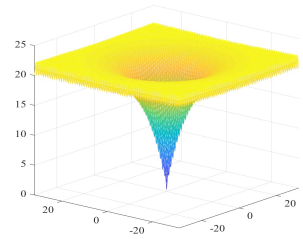
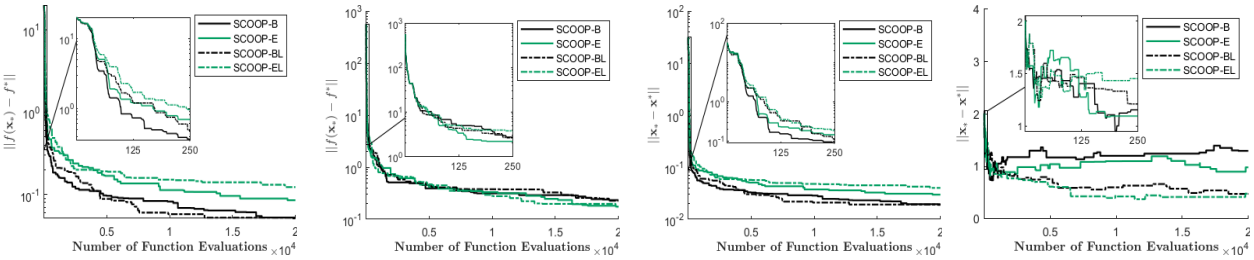


Figure 5: Ackley in $\mathbb{R}^2$; pronounced global optimum and thousands of local minima.

Test Functions We test over three test functions with unique characteristics. The Rosenbrock function, Figure 4, with support $x_i \in [-2,2]\ \forall i$, has global minimum at $(1,\ldots,1)$, and local minimum at $(0,\ldots,0)$ when $d > 4$. The Ackley function, Figure 5, has thousands of local minima across its domain $x_i \in [-32.7,32.7]\ \forall i$, with global minimum at $(0,\ldots,0)$. at $(15,15,15)$ and $(-12,-12,-12)$.

4.1 SCOOP Performance under different Sharing Strategies, function classes and problem dimensions

Effectiveness of SCOOP Information Sharing Strategies for Different Classes of Functions To study the influence of information sharing strategy we test four SCOOP instantiations in 3 dimensions.In all four cases we use $n_{sub} = 2$, $n_{hid} = 1$ with dimension x_1 as the shared active dimension, We set $b_{init} = 10$ and $b_{opt} = 15$, as only 2 dimensions are modeled in each subspace, and allow $B = 40{,}000$ so convergence behavior can be observed. We study the behavior over the 3 dimensional Ackley and Rosenbrock functions. of SCOOP with best observed sample sharing (*"SCOOP-B"*), SCOOP with best sharing and link failure (*"SCOOP-BL"*), and SCOOP with marginalized expected improvement sharing with and without link faliure (*"SCOOP-E"* and *"SCOOP-EL"* respectively). Results are presented in Figure 6(a)-6(d), where we inlay a plot of the first 250 evaluations to highlight the initial phase of the search, while showing asymptotic behavior.



(a) Ackley $|f(x)-f^*|$ error by function evaluations.

(b) Rosenbrock $|f(x)-f^*|$ error by function evaluations.

(c) Ackley $||x-x^*||$ error by function evaluations.

(d) Rosenbrock $||x-x^*||$ error by function evaluations.

Figure 6: Comparison of information sharing strategies over Ackley and Rosenbrock for the 3-d case.

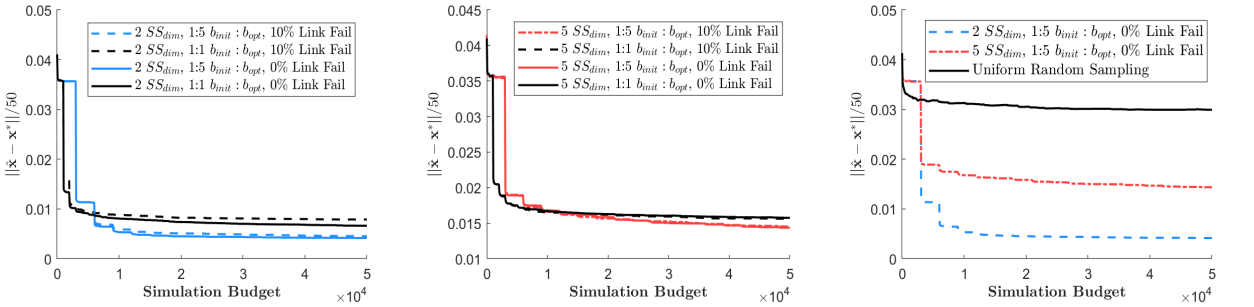
As we observe from the results, SCOOP-B and SCOOP-E show different performance over different data sets. On the Ackley function, with a pronounced global minimum, SCOOP-B outperforms SCOOP-

E (Figure 6(a),6(c)) due to the effectiveness of the greedy move over such a function with a pronounced minimum. However, for the Rosenbrock function, where the optimum is buried deep within a valley, we loose the advantage of the greedy search and the two approaches become comparable (Figure 6(b),6(d)). For both functions, we see that link sharing has strong impact in converging to the optimum asymptotically. While link sharing does not provide any benefits during the initial phase of the search.

Impact of SCOOP Parameterization We next test over a 50 dimensional Ackley function (Figure 5). Observing the strong performance of SCOOP with best observed sample sharing from the previous experiments we run a designed experiment over SCOOP-BL to study (1) the ratio of subspace initialization to optimization sampling budget per iteration, $b_{init} : b_{opt}$, (2) the number of active dimension dimensions per subspace, SS_{dim}, and (3) link failure percentage. For each of these three parameters we select two levels to execute a 2^3, 8-run, factorial design with 50 replications for each run.

For subspace active dimensions $SS_{dim} = 2$ or $SS_{dim} = 50$, such that $n_{hid} = 48$ with 25 subspaces or $n_{hid} = 45$ with 10 subspaces. The ratio $b_{init} : b_{opt}$ is set to $1:1$ or $1:5$, and $b_{init} = 10 \times SS_{dim}$ such that b_{opt} takes values of 20 and 100 when $SS_{dim} = 2$, and 50 or 250 when $SS_{dim} = 5$. Link failure percentage is set to either 0% or 10%. We report the average error per dimension with the intuition of spreading the gathered Euclidean error uniformly across each dimension, a fair measure for the symmetric Ackley function. As such in Figure 7, the dependent axis reports $\|\hat{\mathbf{x}} - \mathbf{x}^*\|/50$.

Average results from the 8 runs are presented in Figure 7. Figure 7(a) plots the four experiments where $SS_{dim} = 2$, where there are 25 subspaces with 2 active and 48 hidden dimensions. Intuitively, $b_{init} : b_{opt} = 1:5$ shows better convergence behavior despite needing more samples prior to making substantial improvements, since more samples are spent optimizing each subspace before information sharing occurs. Similarly when $SS_{dim} = 5$, in Figure 7(b), the $1:5$ ratio outperforms $1:1$ in the long run, but not as drastically as the 2 dimensional subspace case. Figure 7(c) illustrates the benefit of having a many subspaces, a large n_{sub}, with many hidden dimensions, a large n_{hid}. $SS_{dim} = 2$ outperforms $SS_{dim} = 5$, as it is much simpler to optimize in 2 dimensions than in 5. Indicating that, when using the greedy best observed sample sharing, the quality of the subspace-generated candidate solution is critical to the overall performance of SCOOP.



(a) Average Euclidean error per dimension for SCOOP formulations with $SS_{dim} = 2$, over the 50 replications.

(b) Average Euclidean error per dimension for SCOOP formulations with $SS_{dim} = 5$, over the 50 replications.

(c) Average error (50 replications) of best $SS_{dim} = 2$ and $SS_{dim} = 5$ SCOOP formulations, and uniform random sampling.

Figure 7: Average results of alternative SCOOP formulations tested on 50 dimensional Ackley function with a normalized to domain, i.e., $[0,1]^{50}$. All results presented are over 50 replications.

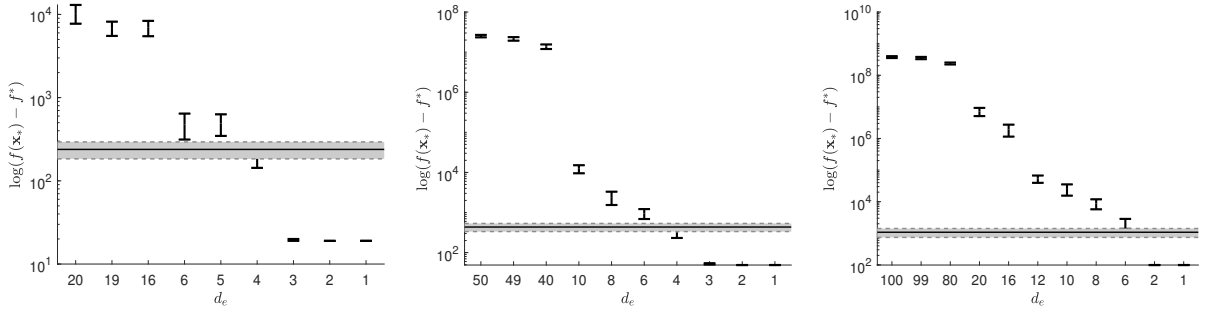
4.2 Performance Comparison against State of the Art High Dimensional Bayesian Optimizers

We compare SCOOP with two state-of-the-art, publicly available and well-maintained, repositories. We test REMBO (Wang et al. 2016), with code at https://github.com/ziyuw/rembo , and the Additive-GP (*Add-GP*)

algorithm with structured kernel learning presented in (Wang et al. 2017) with code at https://github.com/zi-w/Structural-Kernel-Learning-for-HDBBO. We test over the Rosenbrock function in 20, 50, and 100 dimensions, executing 20 replications of each algorithm in each dimension. To compare finite-time performance a wall-clock limit is enforced, allowing 2, 10, and 24 hours for the 20, 50, and 100 dimensional settings. SCOOP uses 2 dimensional subspaces for each case, resulting in 10, 25, and 50 for the three cases.

We note that all of these problems have full effective dimensionality, such that there exist no known low dimensional embedding. We take extra care in the treatment of REMBO and test over a suite of embedding dimensions (d_e) for all the experiments. Figure 8 plots the log of the average best observed function value gap against the different embeddings of REMBO. For comparison we include a bar across each plot of the average performance of SCOOP. Error bars for each REMBO experiment and the SCOOP comparison represent 2 standard errors of the mean over the 20 replications. Clearly, the impact of the embedding dimension is dramatic on REMBO's performance. Though REMBO performs best at the lowest dimensions in this problem this is not true in general, see Table 1 of (Wang et al. 2016). SCOOP avoids the assumption of low effective dimensionality and the need to specify an embedding dimension, a benefit which scales from Figure 8(a)-8(c) as problem dimension increases.

Table 1 presents the results for average optimality gap with respect to both function value and distance to the global minimum, SCOOP, ADD-GP, and a selection of the best performing embedding dimension for REMBO are presented for the 20, 50, and 100 dimensional cases. SCOOP statistically outperforms all algorithms in all cases with respect to distance to $\mathbf{x}^*$, as it is the only algorithm to converge towards the true global optimum at $(1,\ldots,1)$, rather than identifying the local optimum at $(0,\ldots,0)$.



(a) 20 dimensional Rosenbrock experimentation, 2 hours allowed per run.

(b) 50 dimensional Rosenbrock experimentation, 10 hours allowed per run.

(c) 100 dimensional Rosenbrock experimentation, 24 hours allowed per run.

Figure 8: Performance comparison of alternative REMBO embeddings (bars) with SCOOP (band). The log of average best observed function value (± 2 standard errors) over 20 replications reported.

Table 1: Average performance ($\pm$ two standard errors) of SCOOP, ADD-GP, and REMBO with alternative embedding dimension over 20 replications. 20, 50, and 100 dimensional Rosenbrock test cases.

20 dim			50 dim			100 dim		
Algorithm	$\overline{f(\mathbf{x}_*) - f^*}$	$\overline{\lvert\lvert\mathbf{x}_* - \mathbf{x}^*\rvert\rvert}$	Algorithm	$\overline{f(\mathbf{x}_*) - f^*}$	$\overline{\lvert\lvert\mathbf{x}_* - \mathbf{x}^*\rvert\rvert}$	Algorithm	$\overline{f(\mathbf{x}_*) - f^*}$	$\overline{\lvert\lvert\mathbf{x}_* - \mathbf{x}^*\rvert\rvert}$
SCOOP	238.60 ± 27.54	3.12 ± 0.14	SCOOP	434.10 ± 51.17	6.26 ± 0.12	SCOOP	$1.08 \times 10^3 \pm 169.40$	8.85 ± 0.25
ADD-GP	770.05 ± 23.47	3.96 ± 0.11	ADD-GP	$5.84 \times 10^3 \pm 241.81$	7.17 ± 0.11	ADD-GP	$2.52 \times 10^4 \pm 805.54$	12.35 ± 0.22
$d_e = 6$	478.31 ± 82.36	4.71 ± 0.08	$d_e = 8$	$2.42 \times 10^3 \pm 442.14$	7.80 ± 0.12	$d_e = 12$	$5.33 \times 10^4 \pm 6.93 \times 10^3$	14.44 ± 0.33
$d_e = 5$	488.22 ± 70.80	4.73 ± 0.06	$d_e = 6$	952.38 ± 131.93	7.39 ± 0.075	$d_e = 10$	$2.54 \times 10^4 \pm 4.92 \times 10^3$	12.87 ± 0.32
$d_e = 4$	185.95 ± 21.40	4.56 ± 0.06	$d_e = 4$	313.32 ± 40.79	7.30 ± 0.06	$d_e = 8$	$8.80 \times 10^3 \pm 1.54 \times 10^3$	11.66 ± 0.23
$d_e = 3$	19.50 ± 0.23	4.46 ± 0.004	$d_e = 3$	52.38 ± 0.97	7.07 ± 0.004	$d_e = 6$	$2.12 \times 10^3 \pm 368.50$	10.49 ± 0.07
$d_e = 2$	18.98 ± 0.003	4.47 ± 0.001	$d_e = 2$	48.98 ± 0.003	7.07 ± 0.001	$d_e = 2$	98.99 ± 0.003	9.99 ± 0.001
$d_e = 1$	18.99 ± 0.003	4.47 ± 0.001	$d_e = 1$	48.99 ± 0.003	7.07 ± 0.001	$d_e = 1$	98.99 ± 0.001	9.99 ± 0.001

4.3 Application of SCOOP in Neural Network Hyper-Parameter Search

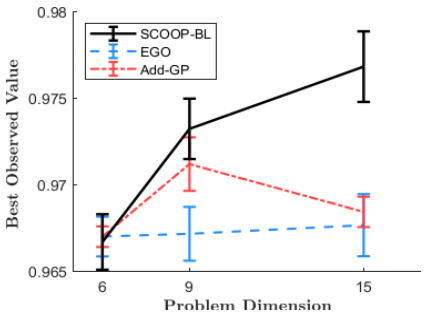


Figure 9: Mean best trained neural net performance, given wall clock of 600, 1800, and 3600 seconds in 6, 9 and 15 dimension search.

Instead of theoretical functions, we look in the problem of training neural nets to classify cancer data. The application description is found in (Wang et al. 2017) and the cancer data in their git repository. As competitors, SCOOP, EGO, and Add-GP were tested for varying dimensional versions of the neural net search problem, with decision variables being the number of nodes in each layer and the number of decision variables (dimensions) corresponding to the number of layers. A wall clock time constraint was implemented for each problem dimensionality, encompassing both neural net training and algorithm search times. Figure 9 shows the average best accuracy achieved by each algorithm across 20 replications. The results show that SCOOP significantly outperforms both EGO and Add-GP in terms of solution accuracy, as the number of problem dimensions increases. Note that Add-GP shows non-monotonic performance behavior in the number of dimensions (layers in the neural net). This is due to the exponential increase in the time Add-GP requires in structured kernel learning, which consumes significantly more time for learning the hyper-parameters of the neural than for training of the classifier.

5 Discussions and Conclusions

In this paper, we presented the *Subspace COmmunication OPtimization* (SCOOP) algorithm for global optimization over high dimensional problems. SCOOP explicitly models and optimizes over multiple complementary low dimensional subspaces and leverages information communication among these relatively easy subspace optimizations to navigate the hard to search full dimensional space. Consequently, SCOOP is specifically suited for problems such as inherently high dimensional machine learning hyper-parameter optimization, that *do not have* low effective dimensionality, and do not easily fit additive assumptions.

Experiment results with several problems with different characteristics have shown that SCOOP outperforms state-of-the-art Bayesian Optimization in quality of observed function values/solution locations, and wall clock time. Further, in a practical hyper-parameter search problem, SCOOP consistently found the best solutions. In general, SCOOP seems to be more scalable with less parameterization.

In this paper, we have shown that SCOOP offers a novel perspective of intelligently decomposing difficult optimization problems into complementary subspace optimizations to overcome the curse of dimensionality and tackle this notoriously hard problem. We finally note that SCOOP can also benefit from modern computational advances in the selection and in the optimization of the individual subspaces and these constitute our future research directions, along with investigating alternative information sharing strategies. Current research is being devoted to investigate alternative information theoretic approaches to address information sharing.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 026257-001. This research is also partially supported by NSF#1827757 “Data-Driven Services for High Performance and Sustainable Buildings”, NSF#1610282 “DataStorm: A Data

Enabled System for End-to-End Disaster Planning and Response", NSF#1633381 "Discovering Context-Sensitive Impact in Complex Systems", NSF#1633381 "Discovering Context-Sensitive Impact in Complex Systems", NSF#1909555 "pCAR: Discovering and Leveraging Plausibly Causal (p-causal) Relationships to Understand Complex Dynamic Systems" and "FourCmodeling": EUH2020 Marie Sklodowska-Curie grant agreement No 690817.

REFERENCES

Bergstra, J., and Y. Bengio. 2012. "Random Search for Hyper-Parameter Optimization". *Journal of Machine Learning Research* 13(2):281–305.

Binois, M., D. Ginsbourger, and O. Roustant. 2015. "A Warped Kernel Improving Robustness in Bayesian Optimization via Random Embeddings". In *International Conference on Learning and Intelligent Optimization*, 281–286. New York City, New York: Springer.

Binois, M., D. Ginsbourger, and O. Roustant. 2017. "On the Choice of the Low-Dimensional Domain for Global Optimization via Random Embeddings". *arXiv preprint arXiv:1704.05318*.

Brochu, E., V. M. Cora, and N. De Freitas. 2010. "A Tutorial On Bayesian Optimization Of Expensive Cost Functions, With Application To Active User Modeling And Hierarchical Reinforcement Learning". *arXiv preprint arXiv:1012.2599*.

Carpentier, A., and R. Munos. 2012. "Bandit Theory Meets Compressed Sensing for High Dimensional Stochastic Linear Bandit". In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*. April 21st – 23rd, La Palma, Canary Islands, 190–198.

Chen, B., R. Castro, and A. Krause. 2012. "Joint Optimization and Variable Selection of High-Dimensional Gaussian Processes". *arXiv preprint arXiv:1206.6396*.

Djolonga, J., A. Krause, and V. Cevher. 2013. "High-Dimensional Gaussian Process Bandits". In *Advances in Neural Information Processing Systems 26*, 1025–1033. Lake Tahoe, Nevada: Curran Associates, Inc.

Gardner, J., C. Guo, K. Weinberger, R. Garnett, and R. Grosse. 2017. "Discovering and Exploiting Additive Structure for Bayesian Optimization". In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. April 20th – 22nd, Fort Lauderdale, FL, 190–198.

Hoang, T. N., Q. M. Hoang, R. Ouyang, and K. H. Low. 2017. "Decentralized High-Dimensional Bayesian Optimization With Factor Graphs". *ArXiv* abs/1711.07033.

Hoffman, M., E. Brochu, and N. de Freitas. 2011. "Portfolio Allocation for Bayesian Optimization". In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 327–336. Arlington, Virginia: Association for Uncertainty in Aritficial Intelligence Press.

Hutter, F., H. H. Hoos, and K. Leyton-Brown. 2011. "Sequential Model-Based Optimization for General Algorithm Configuration". In *Learning and Intelligent Optimization*, 507–523. Berlin, Heidelberg: Springer.

Jones, D. R. 2001. "A Taxonomy of Global Optimization Methods Based on Response Surfaces". *Journal of Global Optimization* 21(4):345–383.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-Box Functions". *Journal of Global Optimization* 13(4):455–492.

Kandasamy, K., J. Schneider, and B. Póczos. 2015. "High Dimensional Bayesian Optimisation and Bandits via Additive Models". In *Proceedings of the 32nd International Conference on Machine Learning*. July 6th – 11th, Lille, France, 295–304.

Kirschner, J., M. Mutnỳ, N. Hiller, R. Ischebeck, and A. Krause. 2019. "Adaptive and Safe Bayesian Optimization in High Dimensions via One-Dimensional Subspaces". *arXiv preprint arXiv:1902.03229*.

Li, X., K. Candan, and M. Sapino. 2018. "M2TD: Multi-Task Tensor Decomposition for Sparse Ensemble Simulations". In *2018 IEEE 34th International Conference on Data Engineering*. April 16th – 19th, Paris, France, 1156–1167.

Marchant, R., and F. Ramos. 2012. "Bayesian Optimisation for Intelligent Environmental Monitoring". In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2242–2249. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Močkus, J. 1975. "On Bayesian Methods for Seeking the Extremum". In *Optimization Techniques IFIP Technical Conference Novosibirsk*, 400–404. Berlin, Heidelberg: Springer.

Molvalioglu, O., Z. B. Zabinsky, and W. Kohn. 2010. "Meta-Control of an Interacting-Particle Algorithm for Global Optimization". *Nonlinear Analysis: Hybrid Systems* 4(4):659–671.

Mutny, M., and A. Krause. 2018. "Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features". In *Advances in Neural Information Processing Systems 31*, 9005–9016. Dutchess County, New York: Curran Associates, Inc.

Oh, C., E. Gavves, and M. Welling. 2018. "BOCK: Bayesian Optimization with Cylindrical Kernels". *arXiv preprint arXiv:1806.01619*.

Rasmussen, C. E. 2004. "Gaussian Processes in Machine Learning". In *Advanced Lectures on Machine Learning*, 63–71. Berlin, Heidelberg: Springer.

Rolland, P., J. Scarlett, I. Bogunovic, and V. Cevher. 2018. "High-Dimensional Bayesian Optimization via Additive Models with Overlapping Groups". In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*. April 9th – 11th, Playa Blanca, Canary Islands, 298–307.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. 2016. "Taking the Human out of the Loop: A Review of Bayesian Optimization". *Proceedings of the IEEE* 104(1):148–175.

Snoek, J., H. Larochelle, and R. P. Adams. 2012. "Practical Bayesian Optimization of Machine Learning Algorithms". In *Advances in Neural Information Processing Systems 25*, 2951–2959. Dutchess County, New York: Curran Associates, Inc.

Snoek, J., O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. M. A. Patwary, P. Prabhat, and R. P. Adams. 2015. "Scalable Bayesian Optimization Using Deep Neural Networks". In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*. July 6th – 11th, Lille, France, 2171–2180.

Srinivas, N., A. Krause, S. M. Kakade, and M. Seeger. 2009. "Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design". *arXiv preprint arXiv:0912.3995*.

Swersky, K. J. 2017. *Improving Bayesian Optimization for Machine Learning using Expert Priors*. Ph. D. thesis, University of Toronto.

Wang, J., and N. Elia. 2012. "Distributed Averaging Under Constraints on Information Exchange: Emergence of Levy Flights". *IEEE Transactions on Automatic Control* 57(10):2435–2449.

Wang, Z., F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. 2016. "Bayesian Optimization in a Billion Dimensions via Random Embeddings". *Journal of Artificial Intelligence Research* 55:361–387.

Wang, Z., C. Li, S. Jegelka, and P. Kohli. 2017. "Batched High-dimensional Bayesian Optimization via Structural Kernel Learning". In *Proceedings of the 34th International Conference on Machine Learning*. August 6th – 11th, Sydney, Australia, 3656–3664.

Zabinsky, Z. B., R. L. Smith, J. F. McDonald, H. E. Romeijn, and D. E. Kaufman. 1993. "Improving Hit-and-Run for Global Optimization". *Journal of Global Optimization* 3(2):171–192.

AUTHOR BIOGRAPHIES

LOGAN MATHESEN is a PhD student in Industrial Engineering at Arizona State Univeristy. His research is in design and analysis of black-box optimization methods, with applications in cyber-physical system testing and falsification. His email is: logan.mathesen@asu.edu.

KAUSHIK KEEZHNAGAR CHANDRASEKAR is an Industrial Engineering graduate student at Arizona State University, specializing in Operations Research. He currently works on projects in deterministic, stochastic, and simulation based optimization. His email is: kkeezhna@asu.edu.

XINSHENG LI now works at Apple and graduate from Arizona State University. His research field is large scale data mining, Personalized Recommender Systems, and data management. His research focuses on density and noise challenges of tensor based analysis. His email address is: xinshing.li@apple.com

GIULIA PEDRIELLI is an Assistant Professor for the School of Computing, Informatics and Decision Sciences Engineering at Arizona State University. Her research is in learning for simulation and simulation optimization. Her email address is: giulia.pedrielli@asu.edu.

K. SELCUK CANDAN is a professor of computer science and engineering at Arizona State University and the director of ASUs Center for Assured and Scalable Data Engineering (CASCADE). His primary research interest is in the area of management and analysis of non-traditional, heterogeneous, and imprecise data. His email is: candan@asu.edu