Parallel Hybrid Metaheuristics with Distributed Intensification and Diversification for Large-scale Optimization in Big Data Statistical Analysis

Wendy K. Tam Cho
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Urbana, IL, USA
wendycho@illinois.edu

Yan Y. Liu
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
Oak Ridge, TN, USA
yanliu@ornl.gov

***Abstract*—Important insights into many data science problems that are traditionally analyzed via statistical models can be obtained by re-formulating and evaluating within a large-scale optimization framework. However, the theoretical underpinnings of the statistical model may shift the goal of the decision space traversal from a traditional search for a single optimal solution to a traversal with the purpose of yielding a *set* of high quality, independent solutions. We examine statistical frameworks with astronomical decision spaces that translate to optimization problem but are challenging for standard optimization methodologies. We address the new challenges by designing a hybrid metaheuristic with specialized intensification and diversification protocols in the base search algorithm. Our algorithm is extended to the high performance computing realm using the Stampede2 supercomputer where we experimentally demonstrate the effectiveness of our algorithm to utilize multiple processors to collaboratively hill climb, broadcast messages to one another regarding landscape characteristics, diversify across the solution landscape, and request aid in climbing particularly difficult peaks.**

***Index Terms*—Optimization, Diversification and Intensification, Statistics, Causal Inference**

I. Introduction

Large-scale optimization problems, characterized by very large decision spaces, have become increasingly common with the rise in data availability. Applications of large-scale optimization abound across all areas of science, with many prominent applications in physics, biology, the social sciences, and engineering. The rise in data has been met with excitement for their enormous potential as well as a wariness for the daunting challenges that arise for their analysis. The ability to organize and analyze enormous stores of data becomes more difficult as the desired analyses increase in sophistication. Optimization techniques that were adequate for smaller data sets may not scale as complexity increases, rendering these standard techniques infeasible for large data sets.

We examine issues that arise with large-scale optimization applications for causal inference modeling of big non-experiment datasets, specifically at the intersection of computational and statistical models. We propose a hybrid metaheuristic approach tailored to augment the exploration capabilities necessary for a massive decision space and increased complexity. We develop intensification and diversification (I&D) protocols, two heuristic components that have been recognized as important in the design of global search methods [4]. Diversification refers to the search capability of visiting many different regions of the decision space while intensification refers to the ability to obtain high quality solutions within these search regions. The diversification portion enables a more global search of the landscape while the intensification component reinforces a convergence search in promising local regions [28].

In a parallel heuristic algorithm, an effective I&D strategy is able to catalyze global search efficiency through collective search effects. The intensification protocol enables the employment of additional computing power to accelerate a search and, in turn, propagates the outcome of the search more quickly, resulting in an improvement of the optimization across all processes. The role of the diversification protocol is to enlarge the scope of the search by maintaining information on active search regions and managing the effort of the processes with the aim of avoiding overlapping and thus wasteful effort.

We must be mindful that parallel I&D strategies usually exhibit highly irregular communication patterns. Consider a commonly employed loosely-coupled parallel heuristic model where a group of independent search processes are simultaneously employed on a single problem instance. Here, when I&D is implemented as a set of message passing protocols, most of the communication is collective by nature. However, conventional global barrier-based (blocking) collective com-

Yan Y. Liu's work in this paper is partly supported by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UTBattelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

Both authors contributed equally to this project.

978-1-7281-0858-219$31.00 ©2019 IEEE.

munication is ill-suited in this instance because there is no obvious critical section in the search logic that would serve as a synchronization communication rendezvous. Furthermore, the I&D notifications are conditioned dynamically on the local search status. Forcing a communication epoch is, thus, not only difficult to design, but inefficient. Moreover, since global synchronization is known to be a major scalability bottleneck, designing models with non-blocking and asynchronous protocols is critical.

We design a parallel I&D hybrid metaheuristic algorithm that implements application-level non-blocking collective communication. The implementation fully leverages asynchrony to enable efficient searching on extreme-scale computing platforms. Our hybrid metaheuristic includes an evolutionary algorithm (EA) as the baseline algorithm for the iterative search process and a tabu mechanism as a memory structure for diversification. The I&D protocols interact with the two heuristics locally and propagate information globally among all participating processes. We evaluate our implementation by solving an *NP*-Hard subset selection problem and measuring the solver's capability to identify not only one but many near-optimal solutions that satisfy a tight bound.

II. Literature Review

A number of algorithms have been proposed to extend basic heuristic models, termed metaheuristics (MHs) [12], including a set that is intended to augment decision space exploration capabilities. Classic MHs include, for example, simulated annealing (SA), tabu search (TS), evolutionary algorithms (EAs), which include genetic algorithms (GAs), and ant colony (AC). Extending the scope of the search space clearly becomes increasingly important and difficult as the size of the search space increases. One avenue for increasing search space capability that has been pursued is combining the favorable features of multiple MHs. The resulting combinations are termed hybrid metaheuristics (HMHs).

HMHs have focused on different aspects of the search process. To expand search exploration, intensification and diversification have been seen as important features [4]. In addition, because of its central stochastic component, EAs have been highlighted as a favored component of an HMH to facilitate search diversification [13]. Many different proposals have been made in this vein. For instance, [41] proposed an adaptive random search with intensification and diversification combined with a genetic algorithm (RasID-GA). [1] presented Meta-RaPS with Path Relinking, a metaheuristic that controls intensification and diversification strategies by adjusting randomness levels via the algorithm parameters which set the number of iterations, priority percentages, restriction percentage, and improvement percentage. [5] controlled diversity by eliminating duplication in the population and controlling the survival of non-elite solutions with deterministic and stochastic selection. [23] developed an algorithm that utilizes a variable population size and issues a periodic partial reinitialization of the population in the form of a saw-tooth function.

While EAs have been primarily seen as contributing on the diversification front, they have also been combined with other heuristics in HMHs to augment the intensification side. [22] combined a microgenetic algorithm to conduct neighborhood generalized hill-climbing in areas identified by the EA. [27] followed a similar path, using a memetic algorithm to implement crossover hill-climbing from solutions found by the genetic operators. [34] similarly implemented a crossover-based adaptive local search (LS) operation to intensify searches in particular regions, and, thus, aid in the effectiveness of the EA for global optimization.

Some EA hybrid algorithms have been implemented on parallel architecture. [35] proposed a distributed GA that utilized four subpopulations or species. [15] proposed heterogeneous distributed GAs with different subpopulations, distinguished by their crossover probabilities and different degrees of exploration versus exploitation intentions. Within the heterogeneous distributed GA framework, [39] suggested varying population sizes with populations gaining and losing population based on their fitness. In addition to these types of teamwork hybridizations, [2] proposed to run collaborative GAs in parallel, each with an HMH integrating EA, TS, and LS, and communicating with an adaptive memory of the search history. Distributed I&D in parallel evolutionary algorithm (PEA) falls into biological distributed algorithms (BDA) [36], which has been an emerging research area in the distributed computing domain.

Indeed, intensification and diversification in evolutionary computation has been an active area of research. Our purpose is to adapt HMH for effective and efficient traversal of massive solution spaces with the goal of producing large sets of solutions that would be suitable for statistical modeling. Here, the need for solution independence highlights the role of the diversification component while the requirement for high quality or nearly optimal solutions harkens back to the aims of the intensification component.

III. Substantive Application: Optimal Subset Selection for Causal Inference

To study the effectiveness of our algorithm, we apply it to the problem of Optimal Subset Selection (OSS), an *NP*-Hard problem that has been shown to translate directly to causal inference models, which seek to establish causal effects from non-experimental data. Experimental research via randomized control trials has long been an important and central research tool across many fields. However, experiments, which require significant investment of both time and resources may not be possible [21]. Moreover, even if a researcher does possess the time and money to conduct a randomized experiment, some questions, such as whether smoking causes lung cancer or the effects of radiation, plainly do not lend themselves to this framework. These experiments, despite their value to society and science, simply cannot be conducted because they violates moral and ethical considerations.

When a randomized experiment is not possible, we may hope to gain some traction on obviously important questions

through observational data. When feasible, this route is appealing since observational data for a large number of phenomena often abound. We can, for instance, easily observe a large number of people who have chosen, on their own accord, to smoke. Hence, the ability to use observational data to make causal inferences would be highly valuable.

However, using observational data to make causal inferences is far from trivial. The lack of random assignment precludes the ability to ascribe differences in response to the treatment. Applying standard statistical models to observational data generally allows only associational inferences. If observational data can be used to successfully mimic experimental data, then we can theoretically derive causal inferences from observational data [19]. Although causal inference models have been developed primarily in the realm of statistical models, [8] demonstrate how an optimization/computational approach nicely dovetails with the theoretic statistical underpinnings and both extends and makes possible new insights from this type of statistical analysis.

Enhancing the ability to make causal inferences from observational data will stimulate research in a wide variety of fields and enhance our understanding of a broad array of phenomena. Existing studies examine, for example, the effects of Do Not Resuscitate (DNR) orders, the impact of utero exposure to phenobarbital on intelligence, and school choice programs, to name but a few [16], [17], [32], [37].

The OSS problem showcases both the enormous gain as well as the additional challenges brought forth from requiring the decision space traversal to produce a large set of identified solutions where the individual solutions are statistically independent from one another.

A. Solution Landscape

A key obstacle for optimization algorithms for the OSS problem is its large-scale and computational complexity. The solution landscape in the decision space for the subset selection problem is idiosyncratic. At a rough level of granularity, the solution landscape is rugged. At fine levels of granularity, it is essentially flat. That is, while the solution landscape is hilly in the sense that it has the usual peaks and valleys, these peaks and valleys are not a rapid succession of precipices, but instead, a series of vast plateaus, and hence, not rugged in the usual sense. For this particular problem, these expansive plateaus manifest themselves throughout the landscape because many subsets significantly overlap with one another. It is evident that, if one swaps out only a single observation from a subset, the new subset is substantially similar. Indeed, for any subset, there is a slew of such minor modifications.

In addition to the distinctive nature of the solution landscape, the decision space is heroically large. Consider that if there are 100 units from which to choose a subset of 10, there are $\binom{100}{10} = 1.73 \times 10^{13}$ possible subsets. More pointedly, substantive applications of the subset selection problem are much larger, often with tens of thousands of units from which one commonly wishes to choose subsets of a few hundred.

B. Problem Description of NP-Hard OSS Problem

We will now formally define the *Optimal Subset Selection* problem, a decision problem that has been shown to be *NP*-Complete [33].

Let $\mathbf{A} = \{A_1, A_2, \ldots, A_n\}$ be a given set of n units where each unit i has some set of k attributes, $\mathbf{X}_i = \{X_{i1}, X_{i2} \ldots, X_{ik}\}$.

Let $\mathbf{C} = \{C_1, C_2, \ldots, C_m\}$ be a set of m units where each unit i has the same set of k attributes, $\mathbf{X}_i = \{X_{i1}, X_{i2} \ldots, X_{ik}\}$. In addition, $m \gg n$.

For any $\mathbf{B} \subseteq \mathbf{C}$, define the function $h : 2^{\mathbf{A}} \times 2^{\mathbf{B}} \rightarrow [0, 1]$ to be a similarity measure between subsets $\mathbf{A}$ and $\mathbf{B}$.

Find the subset $\mathbf{B}$ such that $|\mathbf{B}| = |\mathbf{A}|$, and h is minimized.

IV. Parallel Hybrid Metaheuristic Algorithm

To tackle the OSS problem, we implement a hybrid optimization metaheuristic that incorporates features from evolutionary algorithms and tabu search. Evolutionary algorithms have been successfully deployed on a variety of problems [29], [42], [43]. Likewise, tabu search has also been a good general framework for many optimization problems [6], [24]. A known problem with EAs is premature convergence. TS can aid in preventing premature convergence by maintaining a memory of the search process so that the search engine can be diverted from areas that have already been searched. In this way, EAs and TS are natural partners in HMHs [11], [30], [46].

A sequential EA has been implemented for OSS [7]. For the benchmark LaLonde data set in this literature, this sequential EA identifies a better solution than that identified by any other proposed optimization method [9], [10], [40]. Whether there exists a better solution is an open question, since no one has conducted an exhaustive search of the approximately $\binom{15,000}{185} \gg 10^{300}$ possibilities. More interestingly, for the particular application to causal inference modeling, which is why it is particularly apt for our study here, is that the application requires not simply an optimal solution, but a set of independent solutions that surpasses a particular threshold of subset similarity.

The independence requirement behooves a diversified search as well as an intensified search in local areas. Hence, in this application, diversification plays a dual role. First, it fulfills its traditional role by likely resulting in better solutions since more of the decision space is being searched. Second, spanning out across the decision space is also essential to ensure that the identified solutions are independent of one another.

Figure 1 outlines our parallel hybrid algorithm in a flow chart format. The sequential part of the algorithm follows the "survival of the fittest" principle of EAs that conducts generational, instead of steady-state, update steps to the current population. Each step randomly picks two solutions in the population and probabilistically applies a *crossover* and a *mutation* operator to generate a new solution. If this new solution is an improvement, it is incorporated into the next generation of the population. This process terminates when the stopping criteria are met. We enhanced the baseline algorithm in [7] by adding a homogeneity check to detect the degree of

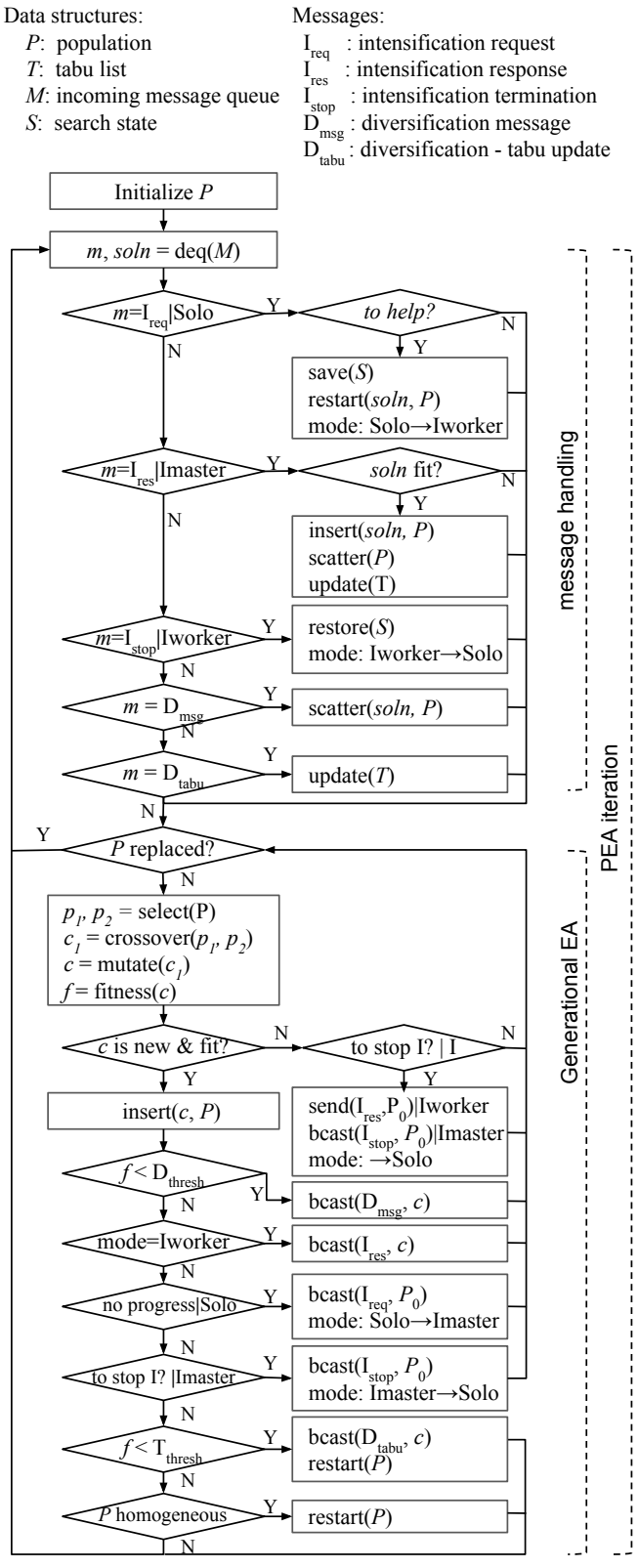

Fig. 1. The parallel EA+tabu algorithm with the I&D protocol.

solution similar in the population. If the current population is sufficiently homogeneous, a random start is triggered in the next iteration.

A. Asynchronous Communication for Distributed I&D

It has been demonstrated that the inter-process communication cost of a parallel EA (PEA) employing global synchronization can command more than 50% of the total execution time when more than 1000 processes are utilized. [26] implement an asynchronous migration strategy through MPI (message passing interface) non-blocking calls that eliminates the costly global synchronization. In their implementation, a grid topology was deployed whereby each EA process communicates with its four directly connected neighbors. Here, we employ a similar asynchronous communication strategy. In contrast to [26], however, the coordination of the I&D protocol must involve all EA processes and requires a scalable collective communication solution.

In the parallel component of our algorithm, we implement an asynchronous I&D design as a set of distributed computing protocols. An intensification request broadcasts a solution that represents a decision region for which an intensified search is requested to all processes. When an EA process receives this request, it probabilistically chooses whether to aid in the intensified search. A diversification request broadcasts a solution that represents a tabu decision region to all EA processes. In such a configuration, a non-blocking collective communication mechanism similar to the MPI non-blocking collective calls [18] is implemented at the application level in order to avoid the costly global barrier in collective communications.

The distributed I&D protocols are implemented using non-blocking MPI calls. *MPI_Isend* and *MPI_Testall* are used to send an I&D message. A parameter, *send_parallelism*, controls the number of send operations that can be initiated before *MPI_Testall* is called to complete them. *MPI_Iprobe* and *MPI_Recv* are used to receive an I&D message. An EA process relays messages to its directly connected neighbors. To avoid duplicate messages, we employ a balanced spanning tree topology to define the connectivity of the EA processes. The height of the tree is $\log_d np$, where np is the number of EA processes, and d is the connectivity degree or the number of directed neighbors. A message reaches all of processes in this topology in at most $2\log_d np$ hops. Compare this with the $\sqrt{np}/2$ hops, which would be required in a grid topology. The number of hops for broadcasting thus becomes shorter than the grid topology when np is sufficiently large. Although it takes a few hops for a message to reach all EA processes, PEA is known to be resilient to such delays because the delays introduce additional randomness in the search [14].

We considered using MPI non-blocking collectives (*nbcolls*). Although the message handling part of the algorithm provides a rendezvous for receiving messages, the stochastic nature of the EA search makes it extremely difficult, if not impossible, to identify a sending rendezvous because I&D notifications are conditioned on an uncertain

timing of satisfying various I&D thresholds. Implementing non-blocking collective communications at the application level gives us explicit control over the degree of parallelism exhibited in our hybrid metaheuristic algorithm. For example, our algorithm is able to immediately switch to computing after an I&D call is returned. The search logic does not block on broadcast completion. This type of computing and communication overlap occurs at a finer grained level than that supported by MPI non-blocking collectives since MPI non-blocking collectives still assume pseudo synchronization that arises from data dependencies at each process on the broadcast topology (e.g., a binomial tree).

B. Distributed Diversification and Intensification

Although the decision space for a large optimization is enormous, we often observe that multiple processes search in similar regions that may be represented by overlapping alleles. Our distributed diversification is designed to maintain various search efforts in distinct parts of the decision space. It is implemented via a distributed tabu list. In particular, a tabu list is maintained at each process to direct search effort, preventing overlapping searches. When a process identifies a new solution satisfying the goodness threshold, which is a tight solution bound, the solution is saved and outputted, and its alleles are added to the tabu list. The alleles are then broadcast as a D_{tabu} message. Other processes, upon receiving this message, then update their tabu list to register those alleles as "tabu." Since only alleles are registered, the tabu list does not introduce a significant memory requirement. When a process detects that it is searching in similar regions, identified by the tabu list, a random start is initiated.

An additional diversification feature is introduced at regular intervals once a process has exceeded a particular solution quality threshold. In this situation, a function, *scatter()*, is invoked to diversify the local population away from portions of the solution space that exceed a particular threshold even if a solution in this area has not yet been identified. In this case, a D_{msg} message is also broadcast to other processes, which then invoke *scatter()* on their populations.

The distributed intensification protocol enables collaborative hill climbing at difficult points in the optimization. When a process is stuck in a search region for too long without making progress, it broadcasts an intensification request, I_{req}. It then also alters its search mode, *Solo*, into the *Imaster* mode. The *Solo* mode indicates that a process is operating independently of all other processes. The *Imaster* mode indicates that a process has invoked an intensification request and is now potentially searching in conjunction with other processes in the same region. When another process receives an intensification request, it makes a probabilistic decision to aid or not in the intensified search. This probability is proportional to the tightness of the solution bound in the request as well as the total number of processes. Requests from processes in more difficult regions invoke a larger number of helpers, proportional to the total number of processes. A process that chooses to help then saves its current state and turns into the

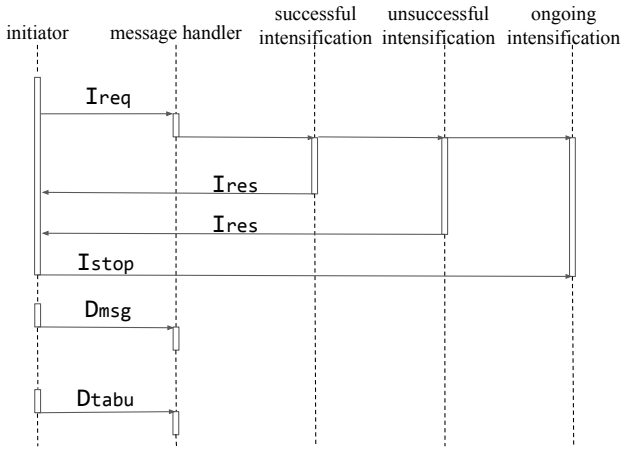

Fig. 2. Sequence diagram for I&D message passing.

Iworker mode. Once the multi-process search surpasses set optimization thresholds, indicating a sufficient progress in the search, the processes resume with their normal independent search effort. The thresholds are set adaptively and respond to rises and falls in the number of iterations that are required before solution improvement is observed. If progress is made by the *Imaster*, an I_{stop} message is broadcast. If progress is made by an *Iworker*, an I_{res} message that includes the new solution is sent back to the *Imaster*, followed by an I_{stop} broadcast by the *Imaster*. Participants then all return to normal search mode. I_{res} messages may still arrive after an *Imaster* sends I_{stop}. These are handled by simply extracting the solutions into the local population.

Figure 2 presets a sequence diagram of the I&D messages. All message communications except I_{res} are broadcast.

V. Empirical Evaluation

We conducted a series of experiments with our search protocols using the seminal Lalonde CPS data set [25], which is often used to benchmark causal inference models. Within this data set, we randomly inserted 100 non-overlapping subsets of size 25 into the first 15,000 observations of the data set. Each of these subsets has a different outcome value and provides a well-matched subset to a simulated treatment group.

We define well-matched with a balance or subset similarity measure, b,

$$b = \sum_{i=1}^{C} w_i \left(KS_i + \mid t_i \mid + \left| \frac{\sigma_{ti}^2}{\sigma_{ci}^2} - \frac{\sigma_{ci}^2}{\sigma_{ti}^2} \right| \right) . \qquad (1)$$

where i indexes the attribute variable, $\mathbf{X}_i$, C is the number of attribute variables, w is a weight, KS_i is the Kolmogorov-Smirnov statistic, t is the t-statistic for the difference of means, and σ_t^2 and σ_c^2 indicate the variance of the treatment and control groups, respectively.

For the optimization to be effective in this setting, it needs to identify 100 distinct subsets. While this example is smaller than many common applications of OSS to identify causal inferences, it is large enough to require significant computation

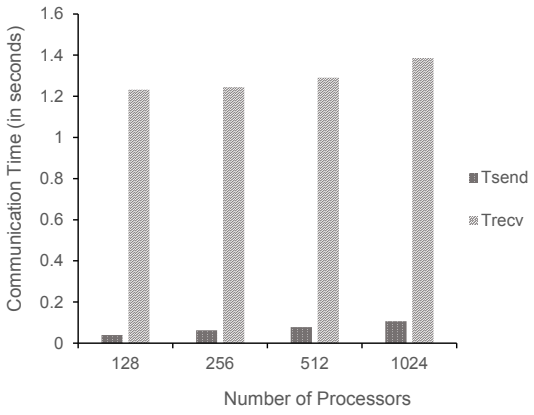

Fig. 3. Sending and receiving time in one-hour runs.

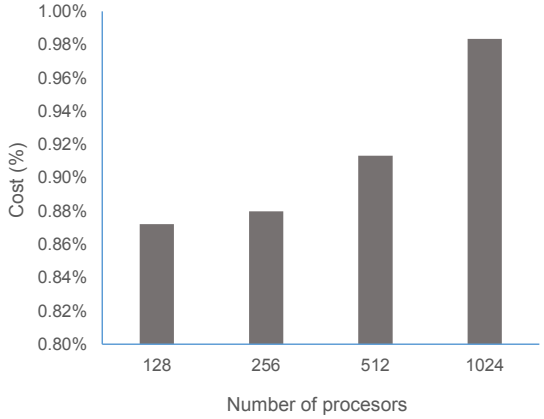

Fig. 4. Receiving cost.

from which we can derive insights into how to design I&D protocols in a parallel computing environment.

Our algorithm is coded in ANSI C and MPI. Experiments are conducted using the Knights Landing computing nodes (Intel Xeon Phi 7250 CPU. 68 cores per node) on the Stampede2 supercomputer at the Texas Advanced Computing Center (TACC). To generate a unique random number sequence on each of the separate processors, we utilize the Scalable Parallel Random Number Generators Library, SPRNG 2.0 [31] to ensure that no two processes repeat the same random number sequence. Notice that starting two processes with different seeds is not sufficient because it does not preclude using the same random number sequence, which would have the undesirable result of similar search paths for processors running independently even without external random noise. When we initialized random number sequences on multiple MPI processes with unique random seeds, we noticed difficulty in diversifying across the decision space.

For all of our experiments, the size of a local EA popluation is 100. To avoid buffer overflow issues in the asynchronous communication [26], *MPI_Iprobe* is invoked four times in an iteration. The topology of the broadcast is a balanced spanning tree.

A. Communication Cost

Since a heuristic search is memory- and compute-intensive, we examine the communication cost and scalability. The algorithm is run with diversification only because D_{msg} is sent on regular basis (every 300 iterations) with dynamic notifications conditioned on *scatter()* and tabu update. Each scenario is run for one hour. Summary statistics from 5 separate runs (one set for each of 128, 256, 512, and 1024 processors) are calculated. Per-processor measures are reported.

Figure 3 shows the distribution of sending and receiving time during each of the one-hour runs. Message receiving time dominates the communication because message probing (*MPI_Iprobe*) occurs more often than sending completion test (*MPI_Testall*). Figure 4 depicts the proportion of receiving operation time in iterations that involve receiving and computing

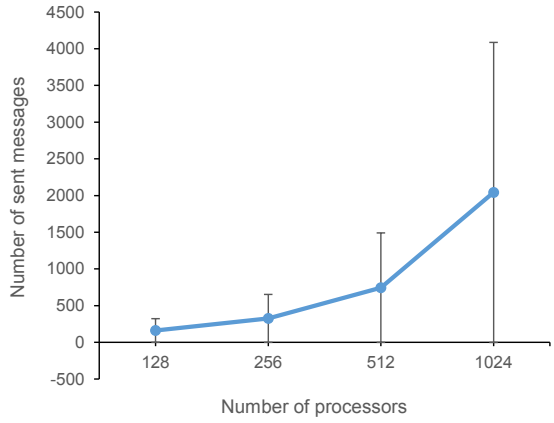

Fig. 5. Message count on sending (mean and standard deviation).

(instead of counting all iterations). Even in such iterations, the communication cost is low. Overall, the communication cost increases only slightly as more processors lead to more broadcasts, but remains under 1% with 1024 cores.

Figure 5 counts the total number of messages sent by each process. High variation is observed since leaf nodes on the broadcast tree forward messages less frequently than non-leaf nodes. Receiving is balanced because most of messages are broadcasts that reach every processor.

B. Intensification and Diversification Protocol Run Separately

Table I shows the result of our diversification and intensification protocols when they are run separately. We ran experiments utilizing 136, 272, 544, and 1088 processor cores and recorded the time (in seconds) that it took to recover 100 solutions. The first through third columns show the results when only the diversification protocol is invoked. The difference in the columns is that we vary the fitness threshold at which we invoke the *scatter()* feature. As the fitness value approaches zero, the quality of the solution increases. For all of the threshold values tested, it appears that the diversification protocol is effective and significantly improves the

TABLE I
Results for Diversification and Intensification Protocol Run Separately

	Diversification			Intensification
Processors	Threshold 0.02	Threshold 0.05	Threshold 0.10	
136	1592.82	1388.66	1738.47	2912.08
272	923.83	684.86	749.26	1567.44
544	396.87	446.04	451.74	675.79
1088	253.64	263.32	258.78	308.51

Time shown in seconds

performance of the search. Moreover, the scalability of the algorithm is maintained as the number of processes increases. However, it does appear that the level of effectiveness of the diversification protocol is related to the point in the optimization at which it is evoked. In particular, while setting the threshold at 0.02 is more effective than not invoking the diversification protocol at all, invoking the protocol a bit further from an optimal solution (at a threshold of 0.05) produces an even better result. At the same time, when the threshold is set even further, the performance is still better than without the protocol, but degrades from the threshold level of 0.05.

Our intensification protocol is adaptive, invoked when no new elite has been identified for some number of iterations. Processes choose probabilistically to provide aid, with the probability increasing as the number of iterations without the identification of a new elite chromosome increases. The results from our intensification protocol are shown in the last column in Table I. Here, as when the diversification protocol was invoked alone, invoking just the intensification protocol also produces an improvement over when no intensification protocol is used. The improvement, however, is not as large as when the diversification protocol is invoked.

The key to efficiency in an intensification effort is to balance the need for computational effort and communication overhead. When the number of processes is lower, the intensification protocol utilizes a larger proportion of the resources, which may not be ideal. Ideally, this is tuned to the difficulty of the search and the available resources. In addition, if a search is intensified when less effort is needed for search progress, then the communication cost is purely a cost without a counterbalancing substantial benefit. As such, identifying a useful point at which to invoke the intensification protocol, which must be application specific, seems critical for performance. We note that we also observed with the diversification protocol that there appears to be a "sweet spot" at which invoking these protocols results in the greatest performance gain.

C. Intensification and Diversification Protocols Run Together

Figure 6 shows our results when we run the intensification and diversification protocols simultaneously, with diversification threshold 0.05. The boxplot produces a graphic display of a set of 5 runs for the various numbers of processors. There is the most variance with 136 processes, with increasingly less variance as the number of processes increases. When the protocols were run simultaneously, the average amount of time required was lower than either diversification or intensification alone for the 544 and 1088 processor runs. Different thresholds on diversification were also tested. Using both protocols was better in all instances when the Threshold was 0.02 or 0.10, but the values when the Threshold was 0.05 with 136 and 272 processes were slightly better with just the diversification protocol.

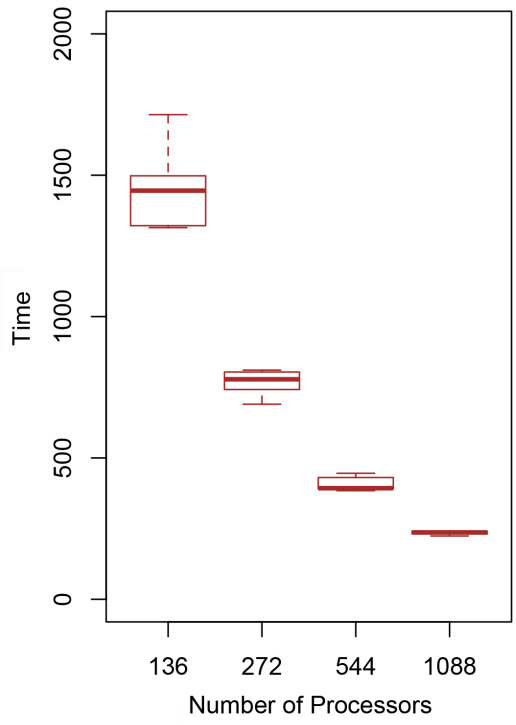

Fig. 6. Intensification and Diversification Protocols Run Simultaneously

VI. Discussion

To harness massively parallel computing power for big data statistical analysis, an efficient and effective I&D strategy becomes more pertinent. Inefficiencies in individual search processes must be minimized as more computing resources are used. In addition, the benefits of collaborative search can proliferate in such parallel environment, perhaps improving performance at greater than linear speeds. We have explored

asynchrony in parallel EA computing and communication and developed a parallel intensification and diversification strategy in a hybrid metaheuristic to address both needs.

Our work shows that our protocol parameters affect the efficiency of our algorithm. In future work, we intend to dynamically tune the optimal problem-specific thresholds and parameters of our protocols within the algorithm itself. This would involve some data keeping to identify, for any particular application, problem regions that require an intensified search effort. As well, we would employ a similar process to identify when the fitness value achieves a level at which diversification would be helpful.

In the exascale computing era, asynchronous communication and communication avoidance/reduction are major components in the development of highly scalable algorithms. The algorithms and heuristics that we develop exhibit desirable scalability and have a general framework for straightforward adaptation to other large-scale optimization problems. Our work lays the foundation for a parallel EA library that provides an application programming interface for leveraging fine-grained computing and communication overlap for highly scalable evolutionary computation. We plan to employ this library to generate a large number of samples from massive observational datasets for causal inference modeling in real-world applications.

VII. Acknowledgements

The experiments conducted in this paper used the Extreme Science and Engineering Discovery Environment (XSEDE) resources, which are supported by National Science Foundation grant number ACI-1548562. Specifically, the authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources, i.e., the Stampede2 system, that have contributed to the research results reported within this paper.

References

[1] Arif Arin and Ghaith Rabadi. 2016. "Performance of an Intensification Strategy Based on Learning in a Metaheuristic: Meta-RaPS with Path Relinking." In *Heuristics, Metaheuristics and Approximate Methods in Planning and Scheduling*, Ghaith Rabadi (Ed.). Springer.

[2] Vincent Bachelet and El-Ghazali Talbi. 2000. "COSEARCH: a Co-Evolutionary Metaheuristic." In *Proceedings of the 2000 Congress on Evolutionary Computation*, Vol. 2. 1550–1557.

[3] Thomas Back, David B. Fogel, and Zbigniew Michalewicz (Eds.). 1997.*Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, UK.

[4] Christian Blum and Andrea Roli. 2003. "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison." *Comput. Surveys* 35, 3 (September 2003), 268–308.

[5] Nachol Chaiyaratana, Theera Piroonratana, and Nuntapon Sangkawelert. 2007. "Effects of Diversity Control in Single-Objective and Multi-Objective Genetic Algorithms." *Journal of Heuristics* 13, 1 (February 2007), 1–34.

[6] J. Chakrapani and J. Skorin-Kapov. 1993. "Connection Machine Implementation of a Tabu Search Algorithm for the Traveling Salesman Problem." *Journal of Computing and Information Technology* 1, 1 (29–36 1993).

[7] Wendy K. Tam Cho. 2018. "An Evolutionary Algorithm for Subset Selection in Causal Inference Models." *Journal of the Operational Research Society* 69, 4 (2018), 630–644.

[8] Wendy K. Tam Cho, Jason J. Sauppe, Alexander G. Nikolaev, Sheldon H. Jacobson, and Edward C. Sewell. 2013. "An Optimization Approach for Making Causal Inferences." *Statistica Neerlandica* 67, 2 (May 2013), 211–226.

[9] Rajeev H. Dehejia and Sadek Wahba. 1999. "Causal Effects in Non-Experimental Studies: Re-Evaluating the Evaluation of Training Programs." *J. Amer. Statist. Assoc.* 94, 448 (1999), 1053–1062.

[10] Alexis Diamond and Jasjeet S. Sekhon. 2013. "Genetic Matching for Estimating Causal Effects: A General Multivariate Matching Method for Achieving Balance in Observational Studies." *Review of Economics and Statistics* 95, 3 (2013), 932–945.

[11] Gautam Garai and B.B. Chaudhurii. 2013. "A Novel Hybrid Genetic Algorithm with Tabu Search for Optimizing Multi-dimensional Functions and Point Pattern Recognition." *Information Sciences* 221 (February 2013), 28–48.

[12] Fred W. Glover and Gary A. Kochenberger (Eds.). 2003. *Handbook of Metaheuristics*. Kluwer Academic Publishers, Massachusetts.

[13] Crina Grosan and Ajith Abraham. 2007. "Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews." In *Hybrid Evolutionary Algorithms*, Ajith Abraham, Crina Grosan, and Hisao Ishibuchi (Eds.). Studies in Computational Intelligence, Vol. 75. Springer Berlin Heidelberg, 1–17.

[14] William E. Hart, Scott B. Baden, Richard K. Belew, and Scott R. Kohn. 1996. "Analysis of the Numerical Effects of Parallelism on a Parallel Genetic Algorithm." In *IPPS '96: Proceedings of the 10th International Parallel Processing Symposium*. IEEE Computer Society, Washington, DC, USA, 606–612.

[15] Francisco Herrera and Manuel Lozano Lozano. 2000. "Gradual Distributed Real-Coded Genetic Algorithms." *IEEE Transactions in Evolutionary Computation* 4, 1 (2000), 43–63.

[16] Jennifer L. Hill, Donald B. Rubin, and Neal Thomas. 2000. "The Design of the New York School Choice Scholarships Program Evaluation." In *Research Designs: Donald Campbell's Legacy*, Leonard Bickman (Ed.). Sage, 155–180.

[17] Daniel E. Ho. 2005. "Affirmative Action's Affirmative Actions: A Reply to Sander." *Yale Law Journal* 114 2011–2016.

[18] Torsten Hoefler, Andrew Lumsdaine, and Wolfgang Rehm. 2007. "Implementation and performance analysis of non-blocking collective operations for MPI." In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*. ACM, 52.

[19] Paul W. Holland. 1986. "Statistics and Causal Inference." *J. Amer. Statist. Assoc.* 81, 396 (1986), 945–960.

[20] Kosuke Imai. 2005. "Do Get-Out-The-Vote Calls Reduce Turnout? The Importance of Statistical Methods for Field Experiments." *American Political Science Review* 99, 2 (2005), 283–300.

[21] S.C. Johnston, J.D. Rootenberg, S. Katrak, W.S. Smith, and J.S. Elkins. 2006. "Effect of a U.S. National Institutes of Health Programme of Clinical Trials on Public Health and Costs." *Lancet* 367, 9519 (April 2006), 1319–1327.

[22] S. A. Kazarlis, S. E. Papadakis, J. B. Theocharis, and V. Petridis. 2001. "Microgenetic Algorithms as Generalized Hill-Climbing Operators for GA Optimization." *IEEE Transactions in Evolutionary Computation* 5, 3 (June 2001), 204–217.

[23] V. Koumousis and C. Katsaras. 2006. "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance." *IEEE Transactions in Evolutionary Computation* 10, 1 (2006), 19–28.

[24] Manuel Laguna, J.P. Kelly, J. L. Gonzalez-Velarde, and Fred Glover. 1995. "Tabu Search for the Multilevel Generalized Assignment Problem." *European Journal of Operational Research* 82 (1995), 176–189.

[25] Robert LaLonde. 1986. "Evaluating the Econometric Evaluations of Training Programs with Experimental Data." *American Economic Review* 76 (September 1986), 604–20.

[26] Yan Y. Liu and Shaowen Wang. 2015. "A Scalable Parallel Genetic Algorithm for the Generalized Assignment Problem." *Parallel Comput.* 46 (July 2015), 98–119.

[27] Manuel Lozano, Francisco Herrera, Natalio Krasnogor, and Daniel Molina. 2004. "Real-Coded Memetic Algorithms with Crossover Hill-Climbing." *Evolutionary Computation* 12, 3 (September 2004), 273–302.

[28] M. Lozano and C. García-Martínez 2010. "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report." Computers & Operations Research 37: 481–497.

[29] O.Z. Maimon and D. Braha. 1998. “A Genetic Algorithm Approach to Scheduling PCBs on a Single Machine.” *International Journal of Production Research* 36, 3 (1998), 761–784.

[30] K.L. Mak and D. Sun. 2009. “A New Hybrid Genetic Algorithm and Tabu Search Method for Yard Cranes Scheduling with Inter-crane Interference.” In *Proceedings of the World Congress on Engineering (WCE 2009)*. London, UK.

[31] Michael Mascagni and Ashok Srinivasan. 2000. “Algorithm 806: SPRNG: A Scalable Library for Pseudorandom Number Generation.” *ACM Trans. Math. Software* 26, 3 (2000), 436–461.

[32] Martin W. McIntosh and Donald B. Rubin. 1999. “On Estimating the Causal Effects of DNR Orders.” *Medical Care* 37, 8 (1999), 722–726.

[33] Alexander G. Nikolaev, Sheldon H. Jacobson, Wendy K. Tam Cho, Jason J. Sauppe, and Edward C. Sewell. 2013. “Balance Optimization Subset Selection (BOSS): An Alternative Approach for Causal Inference with Observational Data.” *Operations Research* 61 (March/April 2013), 398–412.

[34] Naimul Noman and Hitoshi Iba. 2008. “Accelerating Differential Evolution Using an Adaptive Local Search.” *IEEE Transactions in Evolutionary Computation* 12, 1 (2008), 107–125.

[35] J.C. Potts, T.D. Giddens, and S.B. Yadav. 1994. “The Development and Evaluation of an Improved Genetic Algorithm based on Migration and Artificial Selection.” *IEEE Transactions on Systems, Man and Cybernetics* 24, 1 (January 1994), 73–86.

[36] Mira Radeva. 2014. “Review of BDA Workshop 2014.” *SIGACT News* 45 (2014), 100–104.

[37] Lune Machover Reinisch, Stephanie A. Sanders, Erik Lykke Mortensen, and Donald B. Rubin. 1995. “In Utero Exposure to Phenobarbital and Intelligence Deficits in Adult Men.” *The Journal of the American Medical Association* 274 (1995), 1518–1525.

[38] Donald B. Rubin. 2001. “Using Propensity Scores to Help Design Observational Studies: Application to the Tobacco Litigation.” *Health Services & Outcomes Research Methodology* 2, 1 (2001), 169–188.

[39] Dirk Schlierkamp-Voosen and Heinz Mühlenbein. 1994. “Strategy Adaptation by Competing Subpopulations.” In *Parallel Problem Solving from Nature (PPSN III)*. Springer-Verlag, 199–208.

[40] Jeffrey A. Smith and Petra E. Todd. 2005. “Does Matching Overcome LaLonde’s Critique of Nonexperimental Estimators?” *Journal of Econometrics* 125, 1–2 (2005), 305–353.

[41] Dongkyu Sohn, K. Hirasawa, and Jinglu Hu. 2005. “Adaptive Random Search with Intensification and Diversification Combined with Genetic Algorithm.” In *The 2005 IEEE Congress on Evolutionary Computation*, Vol. 2. 1462–1469.

[42] Cuong C. To and Jiri Vohradsky. 2007. “A Parallel Genetic Algorithm for SIngle Class Pattern Classification and its Application for Gene Expression Profiling in Streptomyces Coelicolor.” *BMC Genomics* 8, 49 (2007), 1–13.

[43] Shuqin Wang, Yan Wang, Wei Du, Fangxun Sun, Xiumei Wang, Chunguang Zhou, and Yanchun Liang. 2007. “A Multi-Approaches-Guided Genetic Algorithm with Application to Operon Prediction.” *Artificial Intelligence in Medicine* 41, 2 (2007), 151–159.

[44] Christopher Winship and Stephen Morgan. 1999. “The estimation of causal effects from observational data.” *Annual Review of Sociology* 25 (1999), 659–707.

[45] Herman A. Witkin, Sarnoff A. Mednick, Fini Schulsinger, Eskild Bakkestrom, Karl O. Christiansen, Donald R. Goodenough, Kurt Hirschhorn, Claes Lundsteen, David R. Owen, John Philip, Donald B. Rubin, and Martha Stocking. 1976. “Criminality in XYY and XXY Men.” *Science* 193 (1976), 547–555.

[46] Q. Zhang, H. Manier, and M.-A. Manier. 2012. “A Genetic Algorithm with Tabu Search Procedure for Flexible Job Shop Scheduling with Transportation Constraints and Bounded Processing Times.” *Computers & Operations Research* 39, 7 (2012), 1713–1723.