"pVTDIRECT---A Massively Parallel Deterministic Global Optimization Algorithm"

Layne T. Watson
Virginia Polytechnic Institute and State University

Summary

Parallel stochastic global search algorithms readily scale, but they are also profligate wastrels of CPU cycles, in the sense of total flops per percent improvement in the objective function. Deterministic global optimization algorithms, in contrast, are very frugal users of CPU cycles, but difficult to parallelize and exceedingly difficult to scale massively. This project concerns a parallel deterministic global optimization algorithm, called pVTDIRECT, that has successfully scaled to the terascale System X at VPI&SU, and is being used on large scale systems biology problems here and solid state physics problems at Ames Laboratory..

Background. Global search algorithms have been increasingly applied to large scale optimization problems in many fields. Compared to local methods, these global approaches are more likely to discover the global optimum instead of being trapped at local minimum points for complex nonconvex or nonlinear problems with irregular design domain. The DIRECT algorithm (proposed by D. Jones in 1993) is one such global optimization algorithm that has been applied to large scale engineering design problems such as aircraft design, pipeline design, routing, surface optimization, wireless transmitter placement, molecular genetic mapping, and cell cycle modeling. The complexity of these applications ranges from low dimensional with 3--20 variables to high dimensional with up to 143 variables.

Compared to local methods, global optimization methods generally have higher computational cost and memory requirement, which often lead to solutions that involve data-distributed parallel computing techniques. In the past decade, the parallel schemes of DIRECT evolved from a classical master-slave paradigm, to a fully distributed version with dynamic load balancing, and recently to a multilevel scheme combining global addressing and message passing models. To further improve program portability, execution robustness, and parallel performance, a massively parallel version has been developed with several dynamic features, which have been evaluated on System X, a 2200-processor Apple G5 cluster. Its performance has been analyzed in terms of data structure efficiency and load balancing. The present work characterizes its performance sensitivity to problem dimension, task granularity, domain partition, and computing environment. The project goals are to (1) ensure the design effectiveness of pVTDIRECT on a variety

*540-231-7540, ltw@cs.vt.edu

of problems and systems, (2) guide the proper choice of optimization parameter inputs specified by users, and (3) describe the design considerations and analysis techniques that can be generalized to apply to parallelizing other global optimization algorithms challenged by large scale applications on massively parallel systems.

Overview of DIRECT. DIRECT is a deterministic global search algorithm for solving optimization problems subject to certain assumptions. The global convergence is contingent on the properties of the objective function and the nature of the constraints. When the objective function is Lipschitz continuous around the global optimum point, the global convergence is guaranteed. The general optimization problem considered here is to find the point $\bar{x} \in D$ that minimizes the given objective function f(x) defined in the N-dimensional domain $D = \{x \in E^N \mid \ell \le x \le u\}$, where ℓ and u are lower and upper bounds on x. The computed solution may or may not approximate a global minimum point depending on the specified stopping condition---a budget of computational cost (i.e., the maximum number of iterations or evaluations) or a semi-global optimization goal (i.e., the relative function value improvement between iterations or the minimum diameter of the subregion centered at $\bar{x}$).

Before the DIRECT search satisfies the stopping condition, it iterates through a few steps to select ``potentially optimal" subregions (SELECTION), to sample candidate points in these subregions (SAMPLING), and to subdivide D accordingly (DIVISION). If the objective function value within a subregion is potentially smaller than that in any other subregions for some Lipschitz constant, the subregion is deemed potentially optimal. This unique selection strategy explores the design space intelligently toward multiple promising subregions, thus avoiding being trapped by local minimum points.

Design of pVTDIRECT. The main high level steps of DIRECT for each iteration are:

1. SELECTION identifies a set of ``potentially optimal" boxes that represent subregions inside a normalized design domain. 2. SAMPLING evaluates new points sampled around the centers of all ``potentially optimal" boxes along their longest dimensions. 3. DIVISION subdivides ``potentially optimal" boxes according to the function values at the newly sampled points.

Note that the original DIRECT starts with a single domain, so there is only one center point for SAMPLING and DIVISION at the first iteration. Unavoidably, a load imbalance occurs at the early stage. To mitigate this problem, an optional step of domain decomposition can be specified by the user to create multiple staring points, one per subdomain. The performance analysis and results show that the multiple subdomain approach not only improves the load balancing, but also converges faster for real world applications.

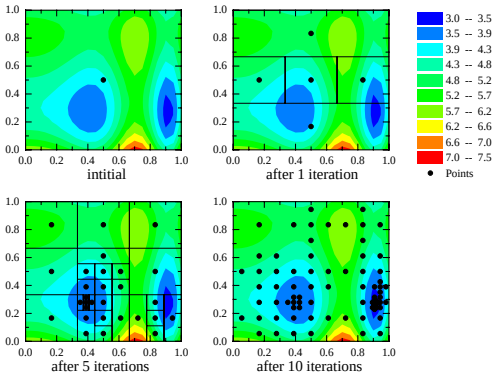
From the second iteration, SELECTION outputs multiple ``potentially optimal" boxes to SAMPLING, which in turn passes the function values of new sample points to DIVISION. The inherent concurrency gives rise to a natural task parallelism. Unfortunately, the same seemingly advantageous step also comes with a disadvantage---data dependency, since any step during an iteration needs to wait for the results of all the previous steps. This inherently sequential nature favors a parallel scheme that decouples SELECTION and SAMPLING

into two different roles---master and worker, respectively.

On a master processor, SELECTION involves convex hull computations because the ``potentially optimal'' boxes are on the convex hull of coordinate pairs containing box center function values and box diameters. When the amount of intermediate data may potentially grow beyond the memory capacity on a single machine, multiple masters should be used to share the data and collaborate on SELECTION in parallel. Also, the multiple masters update the intermediate results at the end of each iteration and check whether the stopping condition is met. The present work recommends that masters evaluate functions locally if the objective function cost is lower than the communication round trip cost between two nodes on the parallel system. This forms the horizontal 1-D scheme, in contrast with the vertical 1-D scheme with a single master distributing the function evaluation tasks to remote workers. Stacking function evaluations is another approach to reduce the communication overhead for distributing cheap objective function evaluations under the vertical scheme.

The overall parallel scheme consists of user configurable m subdomains (SDs), n subdomain masters (SMs), and k globally shared workers (Ws) that request tasks from randomly selected SMs. The above design is implemented purely in Fortran 95 to support high accuracy computation and dynamic data structures that expand at run time for rapidly growing intermediate data. It can be executed on either a single processor or multiple processors depending on the space and computation complexity of the optimization problem. If multiple processors are used, a small set of the MPI library functions are called to establish the interprocessor communication and synchronization. In addition, practical checkpointing methods were implemented to enhance fault tolerance in case of system failure. The key contributions of the massively parallel design are the important techniques developed to reduce the local memory requirement, minimize the network traffic, and balance the workload.

The figure shows the typical progression of the algorithm in two dimensions. It has already had an impact in systems biology, having found the best known 143 parameters for a budding yeast cell cycle model.



Progression of DIRECT sampling in two dimensions.

For further information on this subject contact:

Professor Layne T. Watson
Virginia Polytechnic Institute and State University
ltw@cs.vt.edu
540-231-7540

Or

Dr. Anil Deane
Applied Mathematics Research Program
Office of Advanced Scientific Computing
Phone: 301-903-1465
deane@ascr.doe.gov