

# Exponential speedup with ensemble computing

C. D'Helon, V. Protopopescu, and J. Barhen



Computing and Computational Sciences Directorate

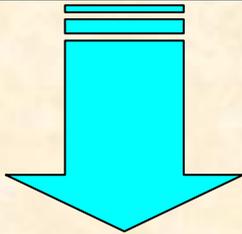
Oak Ridge National Laboratory

Oak Ridge, TN 37831-6355



# Objective: Provide an efficient approach to...

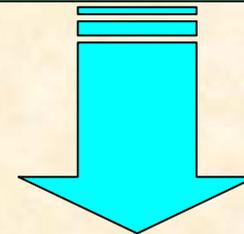
1. Evaluating **and** summing up a set of function samples



## APPLICATION

Evaluation of multi-dimensional integrals/sums and stochastic quantities

2. Solving the **continuous** Global Optimization Problem (GOP)



## APPLICATION

Global optimization problems occur in most scientific and technical fields

# Novelty of Proposed Approach

- Previous classical algorithms are inefficient, whereas quantum algorithms rely on the parallelism of entangled quantum states, which are difficult to maintain
- The proposed ensemble algorithms use the **parallelism of mixed states** in an ensemble of spins, not subject to decoherence
- A given function can be evaluated on all spins at once, and the result(s) are extracted by **ensemble average** measurements

# Summing Algorithm: Statement of the Problem

- Let  $f : \{1, 2, \dots, N\} \rightarrow [0, 1]$  be a real-valued function defined on a discrete set of  $N = 2^n$  samples
- We want to evaluate efficiently the sum  $S_N$ ,

$$S_N = \sum_i f(i)$$

where  $i = 1, 2, \dots, N$

- **Efficiency** is understood in relation to the **query complexity** of the algorithm, i.e., the number of times the function  $f$  is called to be evaluated

# Summing Algorithm: Finite-Precision Values

- Physical system:
  - **input register** with  $n$  two-level spins  $\rightarrow N = 2^n$  samples
  - **output register** with  $k$  two-level spins  $\rightarrow$  specifies the value precision  $\delta=2^{-k}$
- The function values  $f(i)$  are approximated by finite-precision values  $f_i$
- Therefore we are actually evaluating the sum  $S_{N,k}$

$$S_{N,k} = \sum_i f_i$$

which converges exponentially fast to the sum  $S_N$ , as the number of spins  $k$  increases

# Summing Algorithm: Outline of the Algorithm

1. Initialization of an ensemble mixture of input states representing all of the numbers  $i = 1, 2, \dots, N$  with equal weight
2. The function  $f$  is applied to the mixture, using a single unitary transformation  $U_f$  to perform the function evaluation for every input state  $i$  at once. This parallelism results in an ensemble mixture which contains all of the values  $f_i$  in the output register
3. Measurement of the output register averages the contributions from the entire ensemble, yielding a signal proportional to the sum  $S_{N,k}$

# Summing Algorithm:

## 1. Initialization

- The  $n$ -spin **input register** is initialized in an equally-weighted mixed state that represents all of the sample points  $i = 1, 2, \dots, N$

$$\rho_{in}^{(n)} = \frac{1}{N} \sum_{i=1}^N |i\rangle_n \langle i|_n$$

- The  $k$ -spin **output register** is initially set to zero

$$\rho_{out}^{(k)} = |0\rangle_k \langle 0|_k$$

# Summing Algorithm: Binary Encoding Scheme

- The function values  $f(i)$  can be approximately encoded in the output register using the following scheme:

$$|0_1 \rangle \otimes |0_2 \rangle \otimes \dots \otimes |0_k \rangle \leftrightarrow f(i) \in [0, \delta)$$

$$|1_1 \rangle \otimes |0_2 \rangle \otimes \dots \otimes |0_k \rangle \leftrightarrow f(i) \in [\delta, 2\delta)$$

$$|0_1 \rangle \otimes |1_2 \rangle \otimes \dots \otimes |0_k \rangle \leftrightarrow f(i) \in [2\delta, 3\delta)$$

⋮

$$|1_1 \rangle \otimes |1_2 \rangle \otimes \dots \otimes |1_k \rangle \leftrightarrow f(i) \in [1 - \delta, 1]$$

# Summing Algorithm:

## 2. Function Evaluation

- The function  $f$  is evaluated by applying a reversible unitary transformation  $U_f$

$$U_f |i\rangle_n \otimes |0\rangle_k \rightarrow |i\rangle_n \otimes |f_i\rangle_k$$

- This operation transforms the state of the output register to a mixture that represents all of the approximate function values  $f_i$ , **simultaneously**:

$$U_f(\rho_{in}^{(n)} \otimes \rho_{out}^{(k)})U_f^\dagger = \frac{1}{N} \sum_{i=1}^N |i\rangle_n \langle i|_n \otimes |f_i\rangle_k \langle f_i|_k$$

# Summing Algorithm:

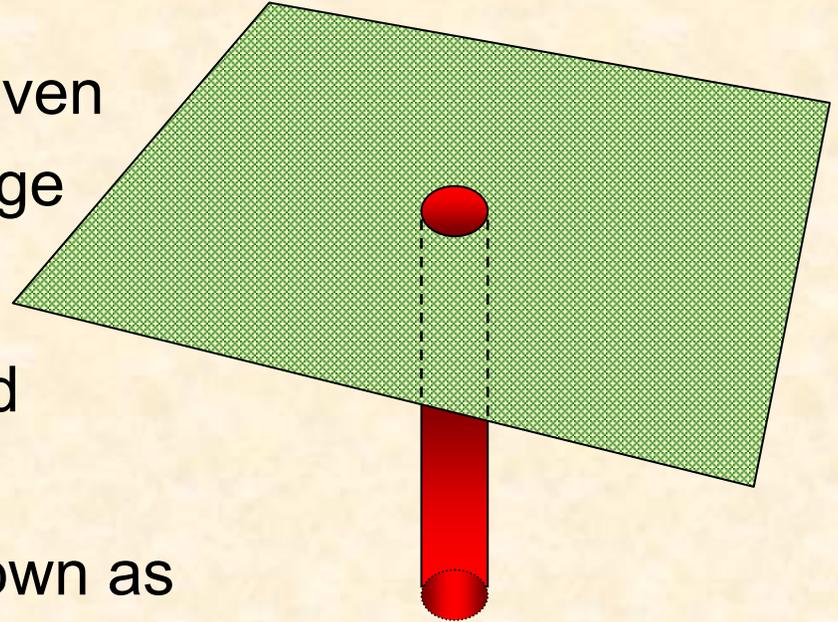
## 3. Measurement

- Each of the spins in the output register generates an **output signal**,  $\gamma_j$ , proportional to the number of spin sub-ensembles that have the  $j$ -th spin in the “up” state
- The sum of the normalized output signals, weighted by the binary factor  $2^{j-1}$ , is an **ensemble average of all the approximate function values  $f_i$**  in the output register:

$$\bar{f}_i = \frac{1}{2^k} \sum_{j=1}^k 2^{j-1} \bar{\gamma}_j$$

# GOP: Statement of the Problem

- **Deceptively simple**: find the absolute minimum,  $\mathbf{x}_{min}$ , of a given objective function  $f$  over the range of its variables
- The function may be specified **analytically**, or by a black-box process, a hidden algorithm known as an **oracle**
- Each function evaluation involves a **costly** computational sequence, therefore the number of function evaluations i.e., the **query complexity**, needs to be kept to a minimum



# GOP: Outline of the Algorithm

1. Map the GOP to a **discrete search problem**, assuming that we have additional information about the objective function  **$f$**
2. Apply Bruschiweiler's **ensemble search algorithm** to obtain a value belonging the **basin of attraction** of the global minimum, in a number of steps that grows **logarithmically** with the input size
3. Use **advanced descent techniques** to find the global minimum,  **$\mathbf{x}_{min}$** , within its basin of attraction

# GOP:

## 1. Map the GOP to a discrete search problem

- Additional information about the objective function is available for many classes of GOP, but cannot be exploited within classical optimization algorithms
- We have established a set of conditions that are reasonable and can be relaxed in principle:
  - (i) the function has a **unique** global minimum,  $f_{\min}=0$  at  $\mathbf{x}=\mathbf{x}_{\min}$
  - (ii) the values of all other minima are **greater** than a constant  $\delta > 0$ , and
  - (iii) the **size** of the basin of attraction for  $f_{\min}=0$ , measured at  $f(\mathbf{x})=\delta$ , is known
- Apply the transformation  $g=\text{ROUND}[f^{1/m}]$  ( $m \gg 1$ ),  $g \in \{0, 1\}$ , and discretize the domain  $[0, 1]^d$ , with a precision that gives only one output equal to zero,  $f(\mathbf{x}_0)=0$ , and all the other outputs equal to one

# GOP:

## 2. Apply Brüscheweiler's ensemble search algorithm

- “**Divide and conquer**” scheme to test whether the zero-valued output of the discretized GOP,  $\mathbf{x}_0$ , belongs to exponentially finer and finer partitions of the set of input values  $(\mathbf{x}_1, \dots, \mathbf{x}_N)$
- After each measurement, the partition containing  $\mathbf{x}_0$  is selected and then **subdivided** into another two equal partitions
- Each test to determine the partition that contains  $\mathbf{x}_0$ , **reduces the size** of the set of input values that contains  $\mathbf{x}_0$  by a factor of 2, therefore the query complexity of the algorithm is  **$O(\log_2 N)$**

# GOP:

## 3. Descent to the global minimum

- Return to the original function  $f$  and apply the descent technique of choice at the value  $\mathbf{x}_0$ , to obtain the actual global minimum  $\mathbf{x}_{min}$
- If the basin of attraction of the global minimum is narrow, the gradients of the function  $f$  may reach very large values that cause overshooting
- Once that phase of the algorithm is reached, one can proceed by applying a scaling (dilation) transformation that maintains the descent mode but moderates the gradients
- On the other hand, as one approaches the global minimum, the gradients become very small and certain acceleration techniques based on non-Lipschitzian dynamics may be required

# Summing Algorithm: Query Complexity

- If the **measurement sensitivity** is adequate to distinguish between distinct normalized output signals with a **precision equal to or better than  $1/N$** , the query complexity is  **$O(1)$** , i.e., the function  **$f$**  is called to be evaluated once
- For large enough  **$N$** , **significant differences** between normalized output signals, differences larger than  **$1/N$** , will **not** be detectable in a single experimental trial
- To enhance the **measurement sensitivity**, the algorithm has to be **repeated a number of times**, which in turn increases the query complexity

# GOP: Query Complexity

- If the **measurement sensitivity** is adequate to distinguish between distinct output signals with a **precision equal to or better than  $1/N$** , the query complexity is determined by the ensemble search algorithm:  **$O(\log_2 N)$**
- The precision requirement is necessary for the first partition test, but may be relaxed for subsequent tests
- For large enough  **$N$** , **significant differences** between output signals, differences larger than  **$1/N$** , will **not** be detectable in a single experimental trial
- To enhance the **measurement sensitivity**, the algorithm has to be **repeated a number of times**, which in turn increases the query complexity

# NMR Implementation

- In an NMR implementation, the signal-to-noise ratio  $S$  scales with the **square-root** of the number of experimental trials,  $N_e$ ,  $\therefore$  both algorithms have to be repeated at least  $N^2$  times, to achieve an adequate **measurement sensitivity**
- The ensemble summing and search algorithms are **more efficient** than their quantum counterparts, using pure states, for an input size  $N$  given by  $N < S^{4/3}$  and  $N^{3/2} \log_2 N < S^2$ , respectively
- The best available signal-to-noise ratio in present NMR technology is  $S \sim 10^4$

# Comparison Table

	Ensemble summing	Ensemble search	Grover's search (pseudo-pure state)	Grover's search (pure state)
measurement sensitivity scaling	$1/N$	$1/N$	$1/N$	1
no. of NMR experimental trials	$N^2$	$N^2$	$N^2$	1
query complexity (single-run)	$O(1)$	$O(\log N)$	$O(\sqrt{N})$	$O(\sqrt{N})$
query complexity (overall)	$O(N^2)$	$O(N^2 \log N)$	$O(N^2 \sqrt{N})$	$O(\sqrt{N})$

- The measurement sensitivity scaling, no. of experimental trials required in an NMR implementation, the single-run query complexity, and the **overall query complexity** for:
1. the **ensemble summing algorithm** we have proposed,
  2. the **ensemble search algorithm** proposed by Bruschiweiler, and
  3. **Grover's search algorithm**, which provides the critical speedup in existing quantum algorithms

if the measurement sensitivity is inadequate i.e., for large  $N$

# Advantages of Ensemble Computing

1. No need to maintain quantum coherences for ensemble algorithms, so they are **easier to implement** than their quantum counterparts
2. Ensemble algorithms may give an **exponential speedup**, for a number of function samples  $N < N_{\max}$ , determined by the measurement sensitivity. In this regime, the proposed summing algorithm requires only a **single** invocation of the function  $f$ , and the solution of the GOP requires a number of evaluations that scales **logarithmically** with the input size
3. **Sheer numbers**: an ensemble of molecules with twenty spins would generate an ensemble space of  $10^6$  states, the equivalent of having a **million processors** available for the classical computation. Each sub-ensemble representing one of these states could be made up of a million identical molecules, to provide a high degree of **fault tolerance**, since the output signal is generated by all of these molecules

# Acknowledgments and References

This research was supported in part by the U.S. Department of Energy, Office of Basic Energy Sciences. The Oak Ridge National Laboratory is managed for the U.S. DOE by UT-Battelle, LLC, under contract No. DE-AC05-00OR22725

- C. D'Helon and V. Protopopescu, *New summing algorithm using ensemble computing*, J. Phys. A, **35**, L1, (2002)
- V. Protopopescu, C. D'Helon and J. Barhen, *Efficient solution to the Global Optimization Problem using an ensemble search algorithm*; Physical Review A.
- V. Protopopescu and J. Barhen, *Solving a class of continuous global optimization problems using quantum algorithms*, Phys. Lett. A, **296**, 9 (2002)
- R. Brüschweiler, *Novel Strategy for Database Searching in Spin Liouville Space by NMR Ensemble Computing*, Phys. Rev. Lett. **85**, 4815 (2000)
- L. Xiao, G. L. Long, H-Y. Yan and Y. Sun, *Experimental realization of Brüschweiler's algorithm in a homonuclear system*, J. Chem. Phys. **117**, 3310 (2002); X. Yang, D. Wei and X. Miao, *Modification and realization of Brüschweiler's search*, Phys. Rev. A, **66**, #042305 (2002)