GPU Acceleration of Optimal Design Selection in Laboratory Experiments in Cognitive Science

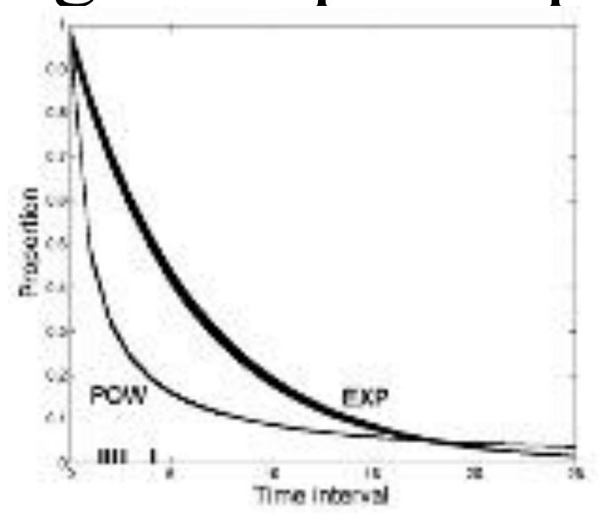
Di Zhao, Woojae Kim, Mark A. Pitt
Mathematical Modeling Lab, Department of Psychology, The Ohio State University, Columbus, OH 43210
zhao.1029@osu.edu

The Problem

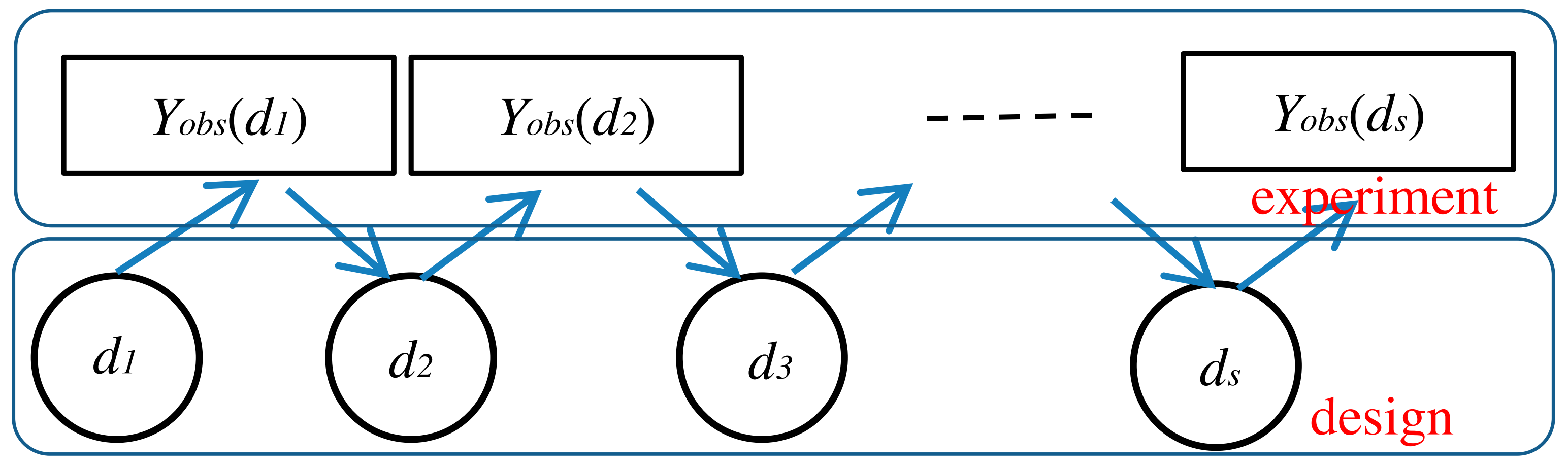
In cognitive science, competing computational models of a cognitive process (e.g., memory, categorization) are evaluated in laboratory experiments with human participants. Monte Carlo-based methods have been developed for identifying the most informative designs. One such method is Adaptive Design Optimization (ADO), in which the next trial is chosen adaptively depending on participants' responses in the preceding trial.

Two models of memory retention

$p = a(t+1)^{-b}\ (POW)$ and $p = ae^{-bt} (EXP)$.



ADO methodology



which leads to an optimization problem:

$$\max_d \sum_m p(m) \iint u(d,\theta_m,y_m)p(y_m|\theta_m,d)p(\theta_m)dy_m d\theta_m, (1)$$

where $p_{s+1}(m)$ and $p_{s+1}(\theta_m)$ are updated by:

$$p_{s+1}(m) = \frac{p_{0(m)}}{\sum_{k=1}^{K} p_0(k)BF_{(k,m)}(z_s|d_s)}\ (2),$$

$$p_{s+1}(\theta_m) = \frac{p(z_s|\theta_{m,}d_s)p_s(\theta_m)}{\int p(z_s|\theta_m,d_s)p_s(\theta_m)d\theta_m}\ (3).$$

$p_{s+1}(m)$ of Equ. (2) is model posterior probability which is a indicator of the performance of ADO. Since Equ. (1) is difficult to solve, we transform Equ. (1) to:

$$h(d,\{m\},\{y_m\},\{\theta_m\}) = \alpha\left[\sum_{m=1}^{K} p(m)u(d,\theta_m,y_m)\right]\left[\prod_{m=1}^{K} p(y_m|\theta_m,d)\,p(\theta_m)\right].(4)$$

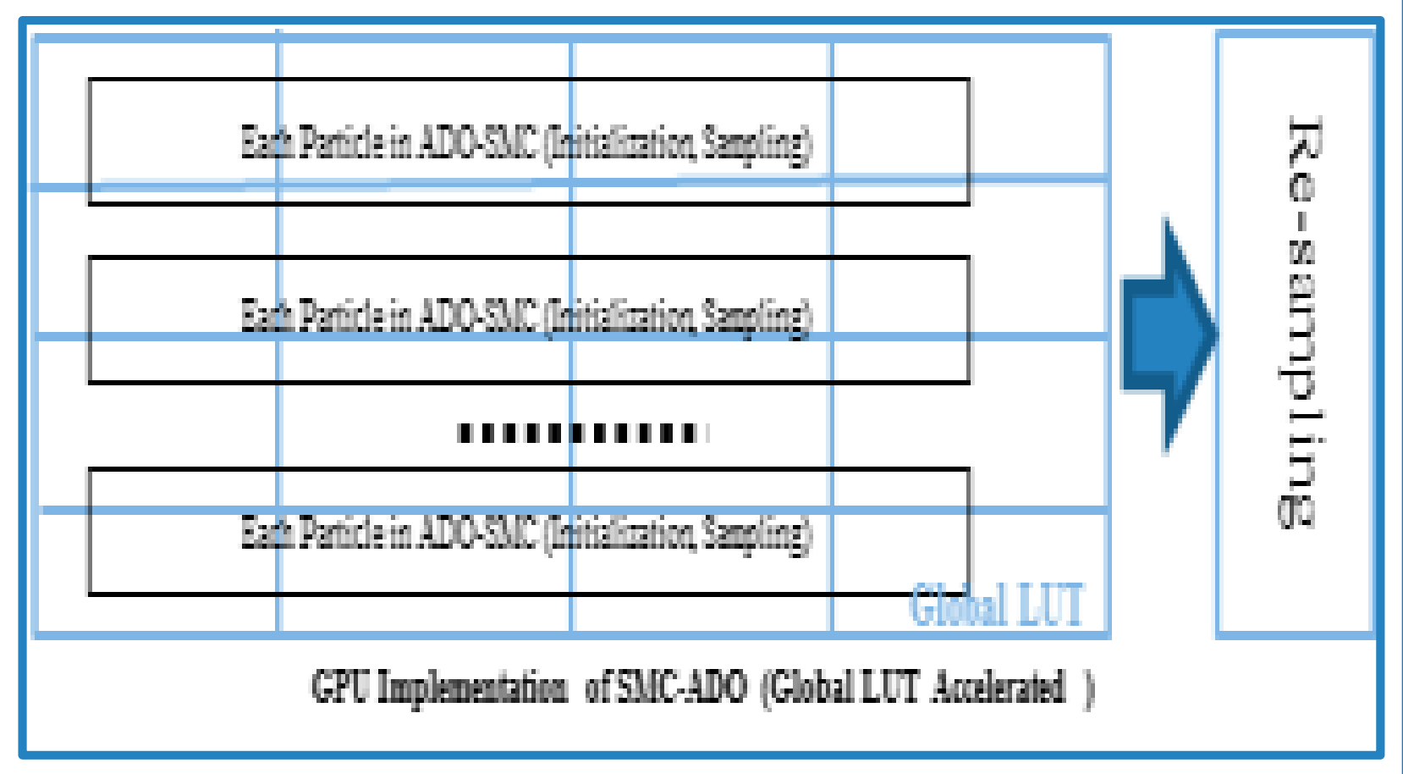
The computational time of Equ. (4) of ADO is too great to be used in real-time experiments.

To solve the conflict between choosing an optimal design and the real-time needs of clinical experiments, we developed five GPU based algorithms for ADO in the styles of Single Instruction Multiple Data: Sequential Monte Carlo (GPU-SMC), Grid Search (GPU-GS), Grid Search in One-dimension (GPU-SMC-1D), Differential Evolution (GPU-DE) and Lookup Table (LUT) accelerated DE (GPU-LUT-DE) against an existing code of CPU based SMC (CPU-SMC).

Algorithm Implementation

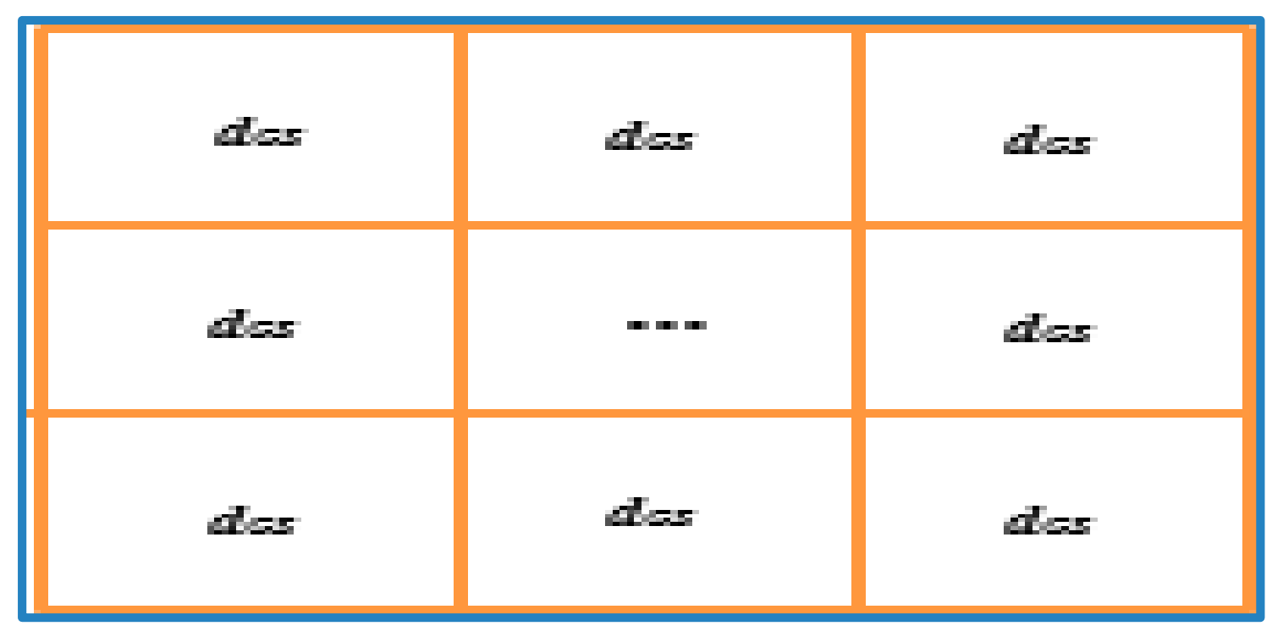
	CPU-SMC	GPU-SMC	GPU-GS	GPU-GS-1D	GPU-DE	GPU-LUT-DE
Algorithm Complexity	High	High	Low	Low	Middle	Middle
GPU Compatibility (1-5, 5=very compatible)	NA	3	4	5	4	4
Acceleration	Vectorization, Global LUT	Vectorization, Global LUT	Vectorization, Global LUT	Vectorization, Global LUT	Vectorization, Global LUT	Vectorization, Global LUT, Dynamic LUT

```
Algorithm: SMC (LUT Accelerated )
1. Building LUT:
•    Sample $d$ from proposal distribution;
•    Sample $\theta$ from $p(\theta|d)$;
•    Sample $y$ form $p(y|\theta,d)$;
•    Calculate all combination $d$, $\theta$ and $y$ of local utility
     function $u(d,\theta,y)$ in Equ. (1);
2. Initialization and Sampling:
•    Sample $d$, $\theta$ and $y$;
•    Fetch the local utility $u$ from LUT;
3. Resampling;
4. Calculating $h(d)$ in Equ. (4) and find the optimal $d$;
```


Each Particle in ADO-SMC (Initialization, Sampling)
Each Particle in ADO-SMC (Initialization, Sampling)
Each Particle in ADO-SMC (Initialization, Sampling)
Global LUT
Re-sampling
GPU Implementation of SMC-ADO (Global LUT Accelerated)

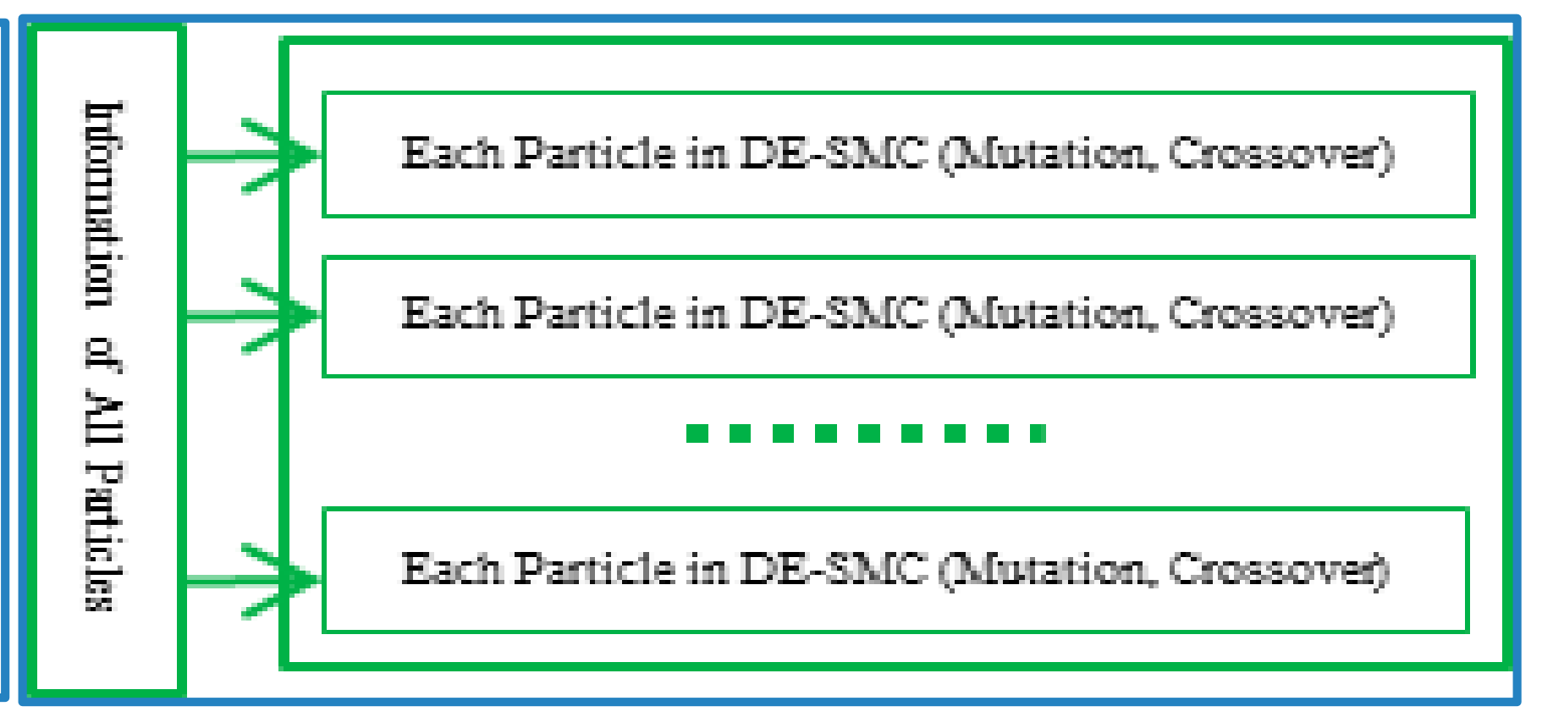
- **SMC**, the distribution of $h(d)$ in Equ. (4) is approximated;
- The solution of Equ. (1) is obtained by finding the optimal from the approximation set of d;
- Pre-calculating all possible combination of d, θ_m and y_m, and stored these values into the global LUT;

```
Algorithm: GS
1. For all discretized $d$ in proposal distribution:
•    Sample $\theta$ from $p(\theta|d)$;
•    Sample $y$ form $p(y|\theta,d)$;
•    Calculate the local utility function $u(d,\theta,y)$ in Equ.
     (1);
2. Calculate $h(d)$ in Equ. (4) and find the optimal $d$ ;
```



- **GS**, no approximation: all discrete points of d are computed;
- All time points d in Equ. (4) in the algorithm distributes well in GPU threads;
- No information exchange is necessary;

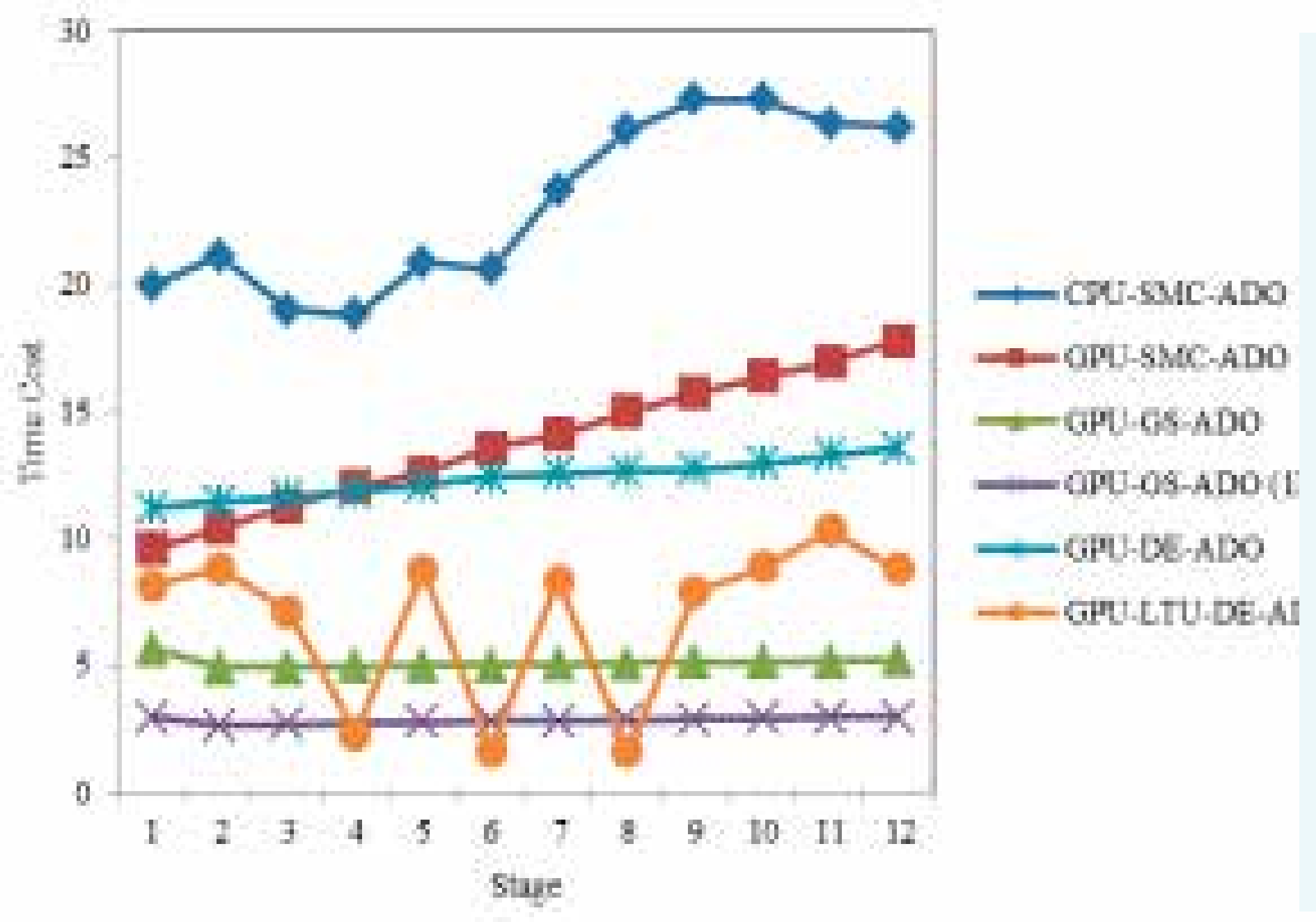
```
Algorithm: DE
1. Initialization of multiple $d$;
2. For each $d$:
•    Generate new $d'$;
•    Sample $\theta$ and $y$;
•    Calculate $u(d,\theta,y)$ in Equ. (1);
•    Calculate $h(d)$ in Equ. (4) and
     decide new $d$;
```


Information of All Particles
Each Particle in DE-SMC (Mutation, Crossover)
Each Particle in DE-SMC (Mutation, Crossover)
Each Particle in DE-SMC (Mutation, Crossover)

- **DE**, time point d is continuously selected to calculate $h(d)$ in Equ. (4);
- Each particle of DE is responsible for a time point;
- The algorithm is accelerated by global LUT and dynamic LUT;

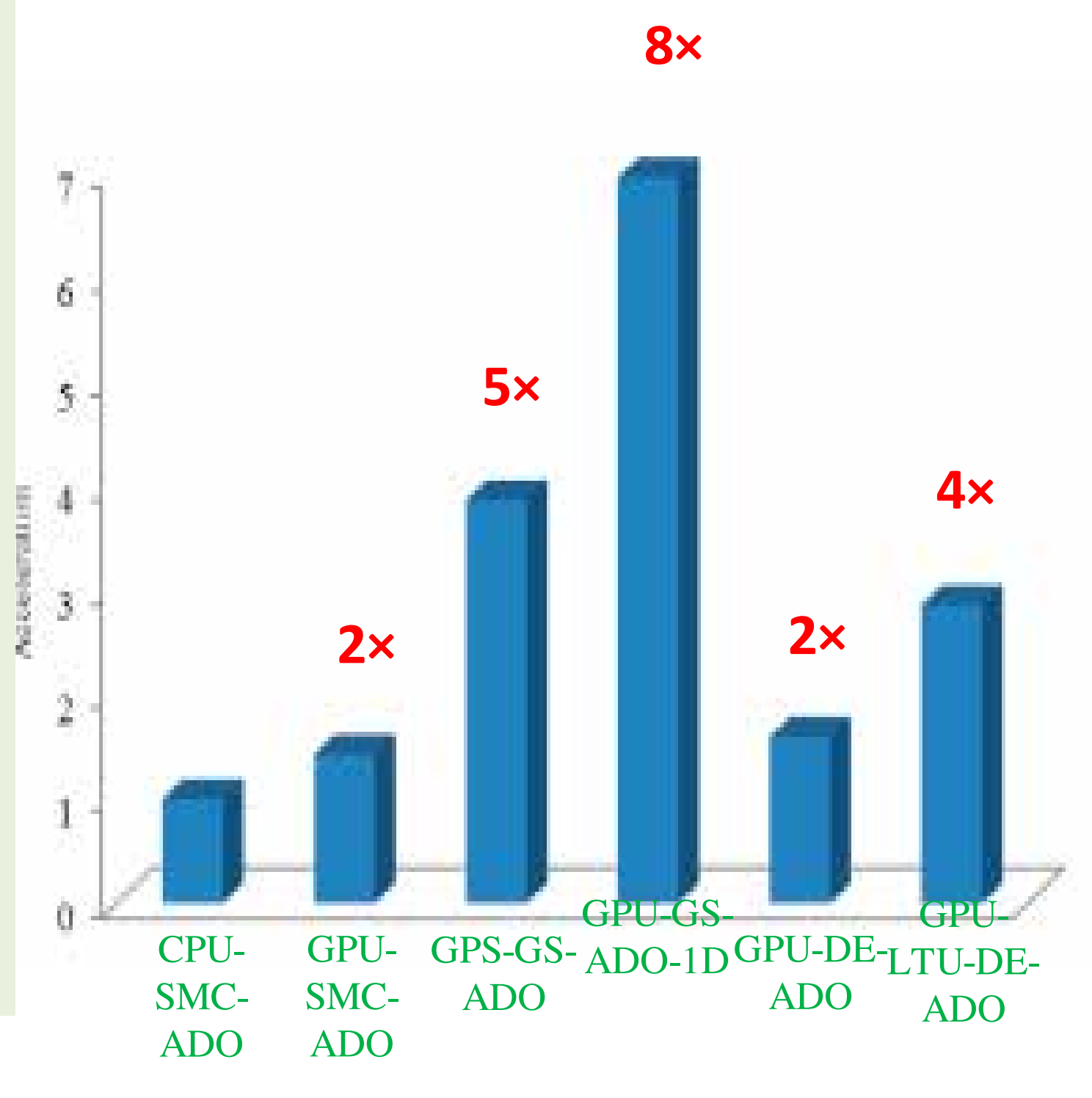
Results

Computational facility is a GPU server with two of Intel Xeon E5620, 72 Gb ECC memory and Tesla C2075 with 4Gb memory.



- All GPU implementations are faster than our existing CPU-SMC code;
- As the progress of the experiment, GPU-SMC spends more and more time, while GPU-GS, GPU-GS-1D, GPU-DE and GPU-LUT-DE keep stable;

- Comparing with the CPU-SMC, GPU-GS-1D is the fastest: accelerates about eight times;
- Algorithms with low complexity is faster than that with high complexity;
- Algorithms with high GPU compatibility is faster than that with low compatibility;
- LUT significantly speeds up these algorithms;



Conclusions

In this poster, we did GPU implementation of SMC, GS and DE for ADO. Computational results shows that:

- GPU significantly accelerates the solution speed of ADO;
- For target distribution without complex shape, GPU-GS-1D is the fastest solver for ADO;
- The fastest algorithm comes from low complexity, high GPU compatibility and acceleration techniques;

References

1. Myung, J.I. and M.A. Pitt, *Optimal Experimental Design for Model Discrimination*. Psychological Review, 2009. **116**(3): p. 499-518.
2. Cavagnaro, D.R., et al., *Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science*. Neural Computing, 2010. **22**(4): p. 887-905.
3. Cavagnaro, D., M. Pitt, and J. Myung, *Model discrimination through adaptive experimentation*. Psychonomic Bulletin & Review, 2011. **18**(1): p. 204-210.

Acknowledgement

This project is supported by National Institute of Health Grant R01-MH57472.