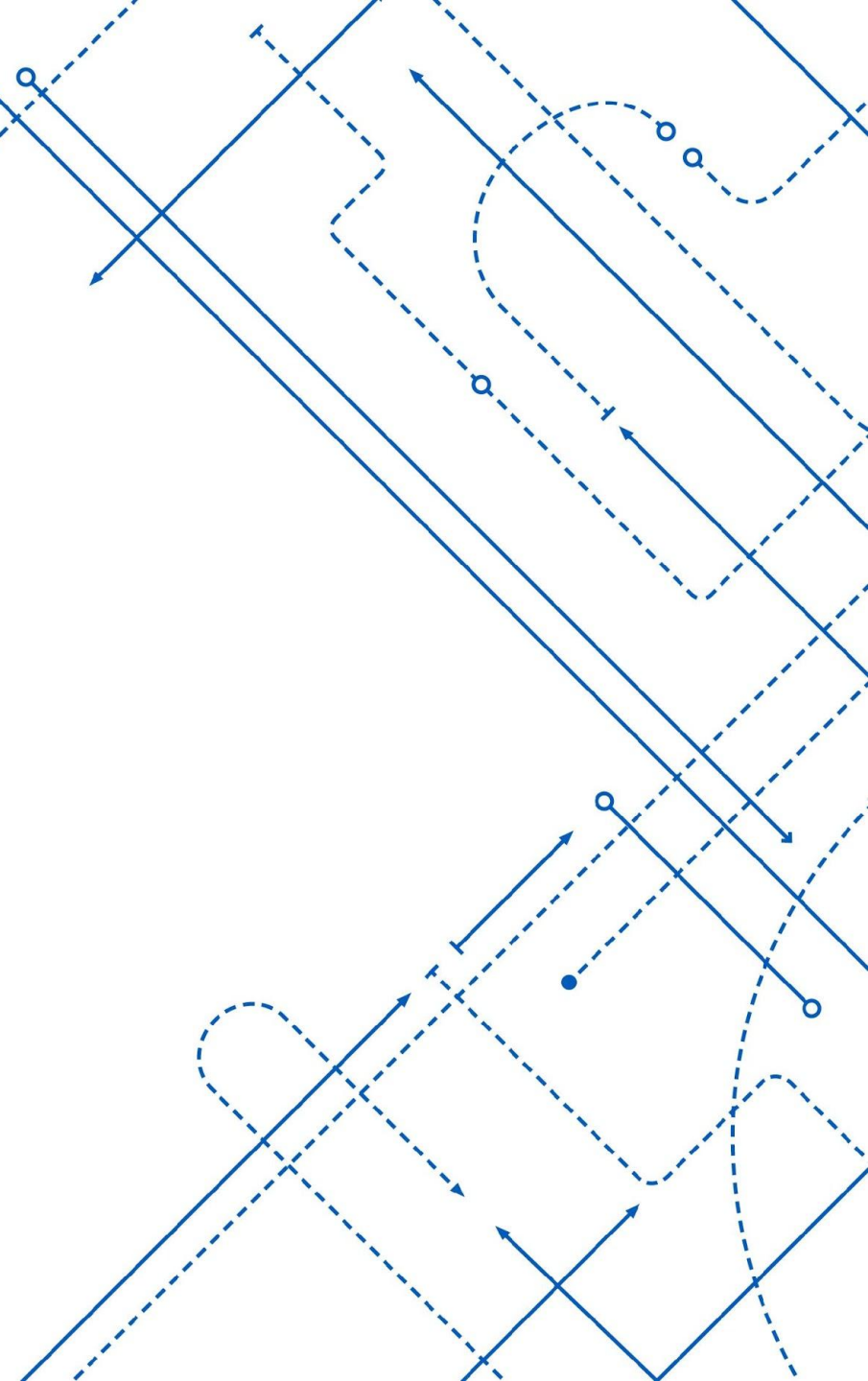
Parallel Particle Swarm Optimization

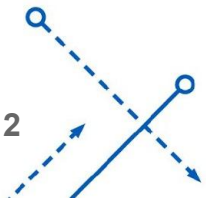
Sourabh Bhagat
CSE 633 : Parallel Algorithms
Instructor : Dr. Russ Miller



University at Buffalo The State University of New York

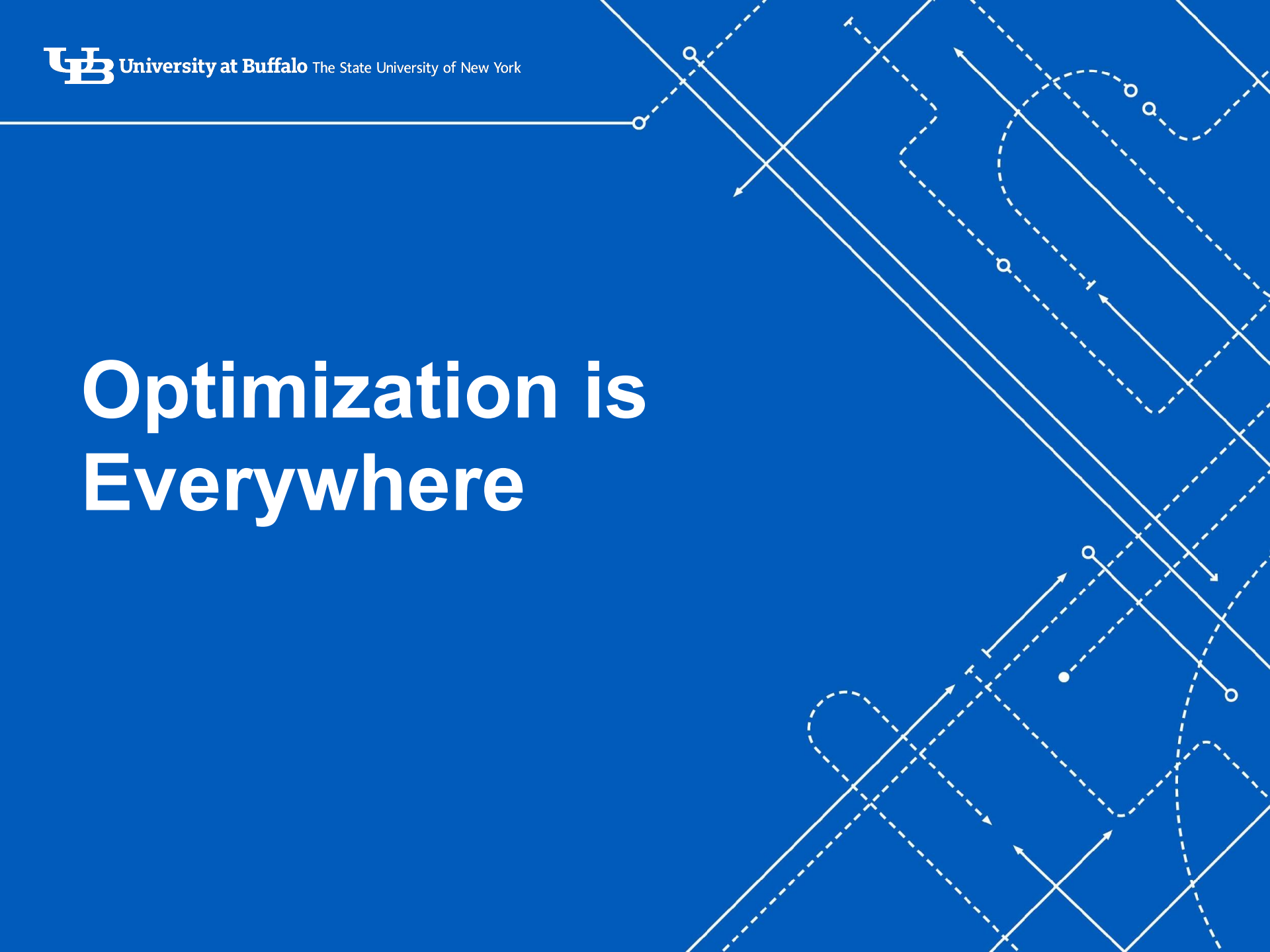
Contents

- Particle Swarm Optimization Algorithm
- How PSO works
- Parallel Approach to PSO
- Proposed Fitness Function
- Experimental Results



University at Buffalo The State University of New York

Optimization is Everywhere



Particle Swarm Optimization

Developed by Russell Eberhart & James Kennedy in 1995 [1]

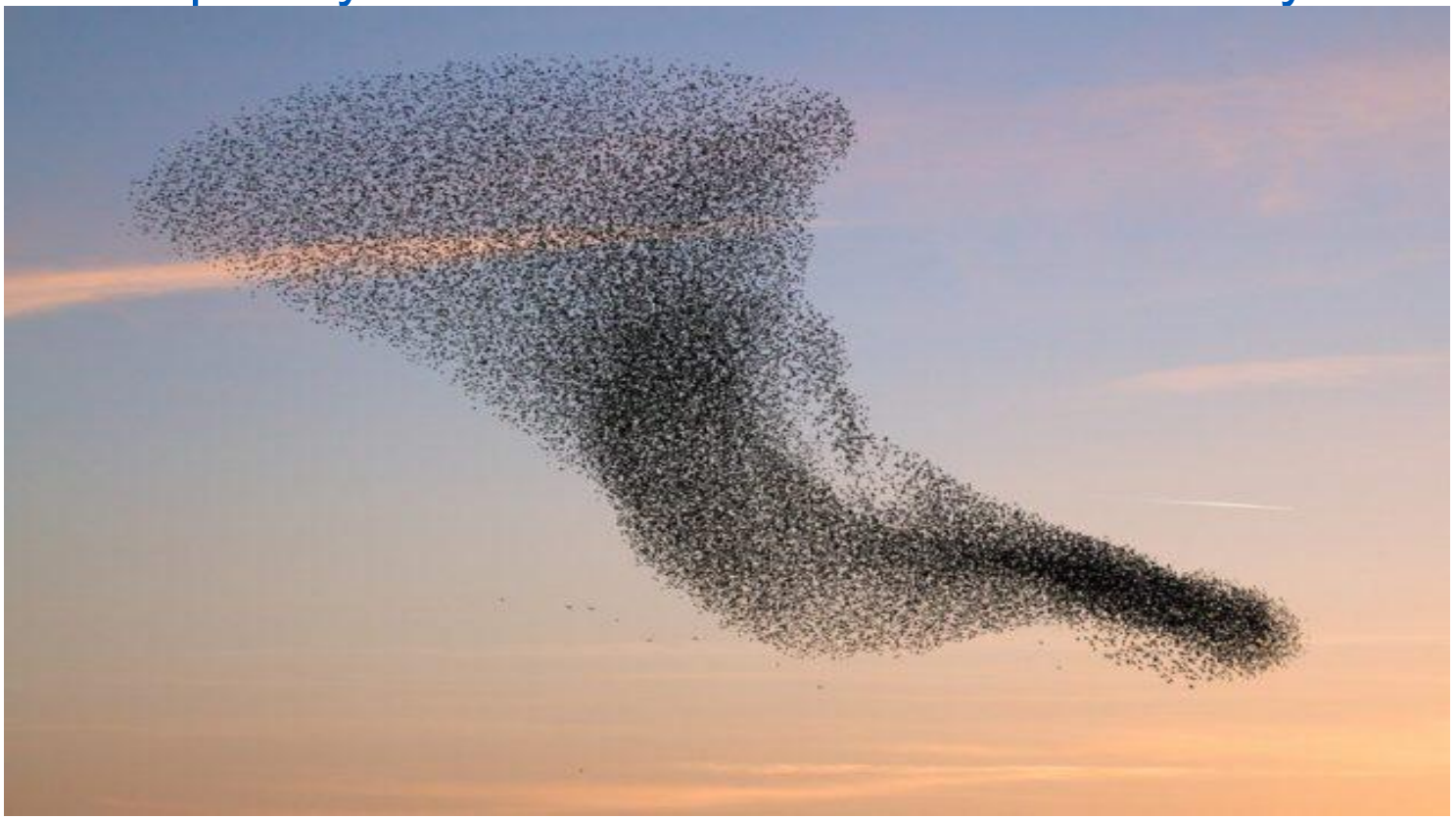
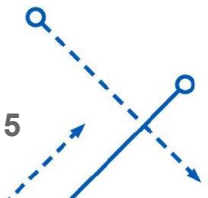


Image Ref : https://sifutrecht.nl/structureer-werk-mensen-organisatietrend-21e-eeuw/

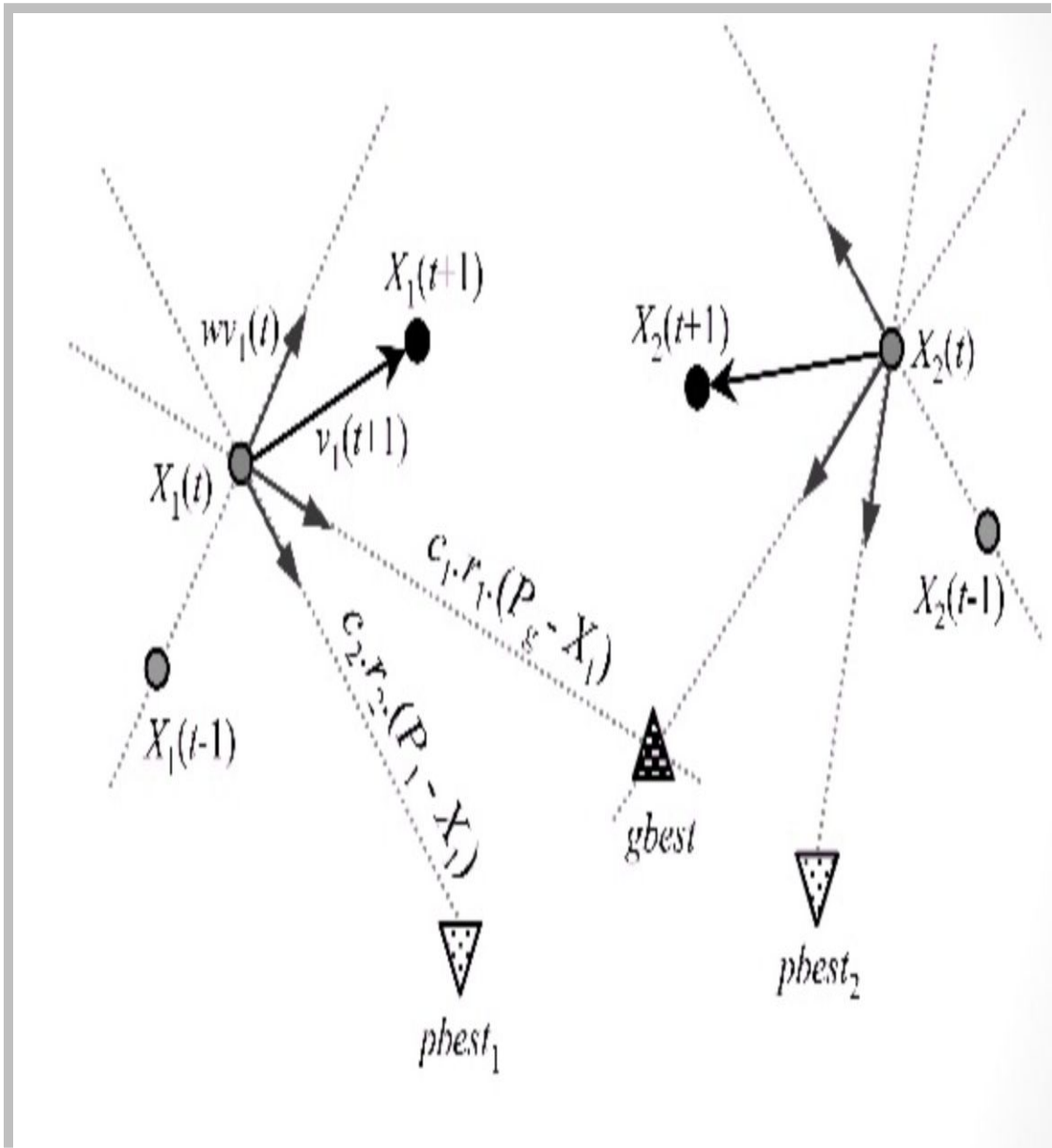
Particle Swarm Optimization (PSO)

- Population based global search algorithm.
- Each solution (particle) fly through search space with directed velocity vector to find better solution.
- Velocity vectors are adjusted based on the historical and inter-particle information.
- A particle adjusts its position according to its own experience as well as the experience of neighboring particles.
- Application : Path finding, Network Design, Clustering, etc.



How PSO works ?

- A particle status on the search space is characterized by two factors:
- - position (Xi)
- - velocity (Vi)
- Fitness function is used to evaluate particle position.



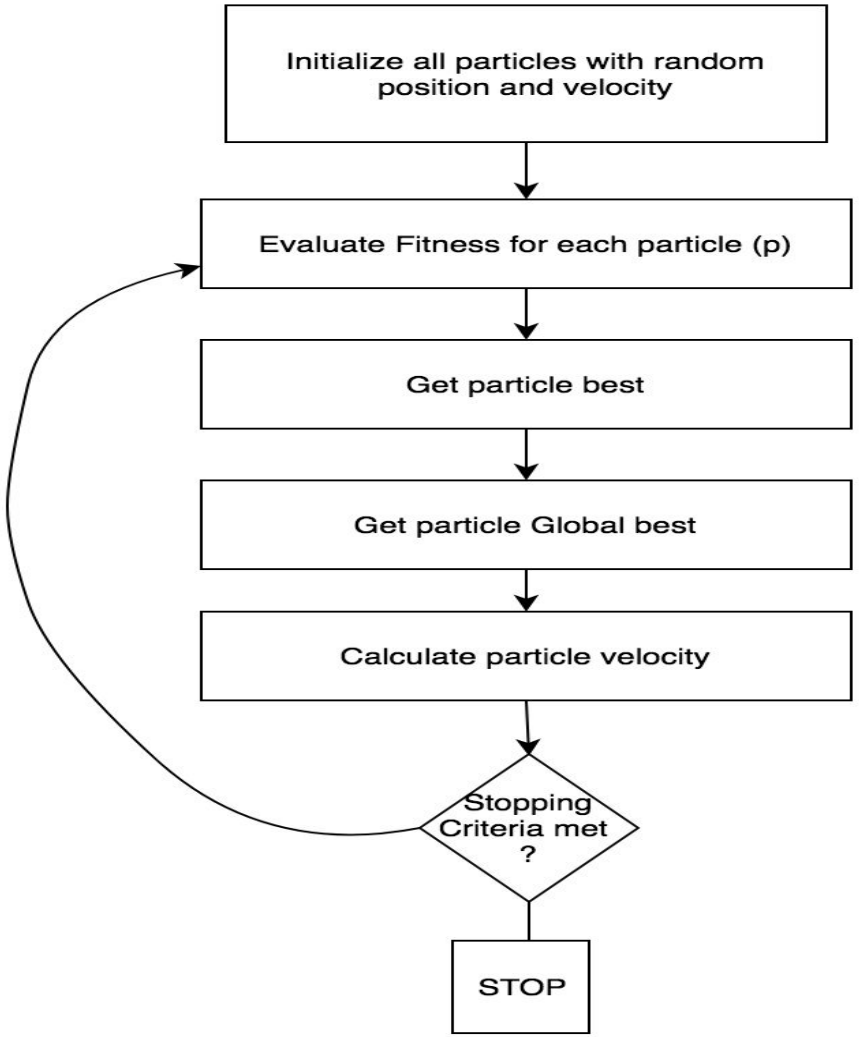
Position and Velocity Update

$$x_{k+1}^i = x_k^i + v_{k+1}^i$$

$$v_{k+1}^i = w_k v_k^i + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i)$$

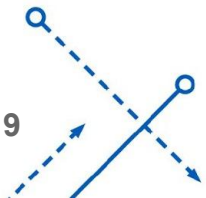
- x_k^i represents the current position of particle i & k represents pseudo-time increment
- p_k^i is the best-found position of particle i upto time steps k
- v_k^i represents particle velocity
- p_k^g represents global-best position among all particle upto time step k
- **c1, c2, r1, r2**, **wk** are constants

Algorithm


Initialize all particles with random position and velocity
Evaluate Fitness for each particle (p)
Get particle best
Get particle Global best
Calculate particle velocity
Stopping Criteria met ?
STOP

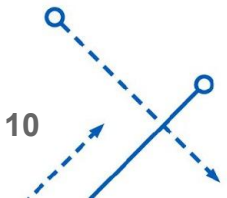
Synchronous Design

- Most of the algorithms given in literature are synchronous in nature.
- They require synchronization point at the end of each iteration before moving to next iteration.
- Generally results in poor parallel efficiency [2]
- Practically impossible for a processor wait for end of the iteration.



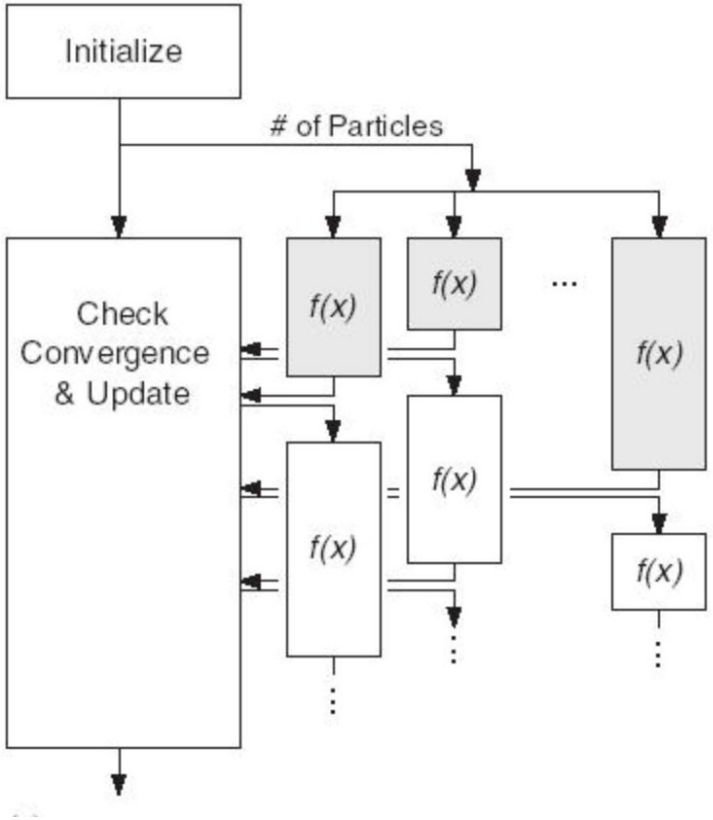
Asynchronous Design

- Don’t require synchronization point
- Next design iteration are analyzed before the current design iteration is completed.
- Optimization can proceed to next iteration without waiting for completion of all function evaluation from current iterations.
- Advantage : No idle processors



Parallel Implementation

- Calculate fitness function in parallel
- Parallel scheme based on MPI
- Master-worker implementation
- Provides dynamic load balancing between processors.



Proposed Fitness Function

- Maximization of Throughput of the network.

$$T(d) = \frac{L - C}{L} R f(g(d)) \qquad [3]$$

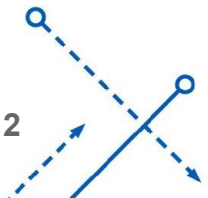
$$g(d) = A \frac{P_t}{N_0 R d^{\alpha}}$$

$$Fitness(i,j) = dist(i,j) + dist(j,BS) \qquad [4]$$

where

$$dist(i,j) = \sqrt{(x1 - x)^2 + (y1 - y)^2}$$

- Objective : **Max {throughput(i,j)}**
- Use case : DDoS Attack Mitigation

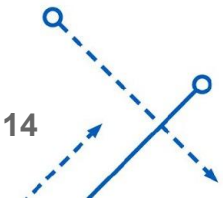


University at Buffalo The State University of New York

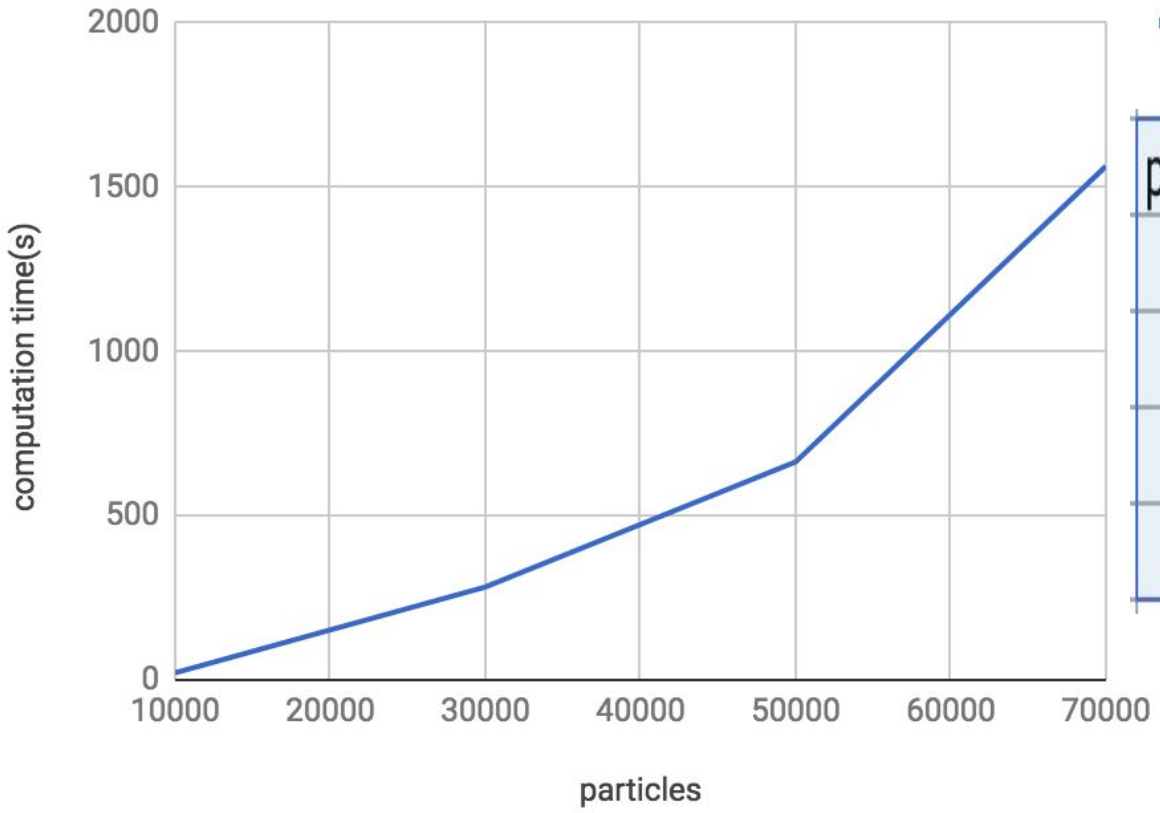
Experimental Results

Parameters

- Number of iterations = 500
- Problem dimension = 2
- Wk = 0.5
- c1, c2 = 1
- L = 108, C = 1, R = 1(Throughput constants)



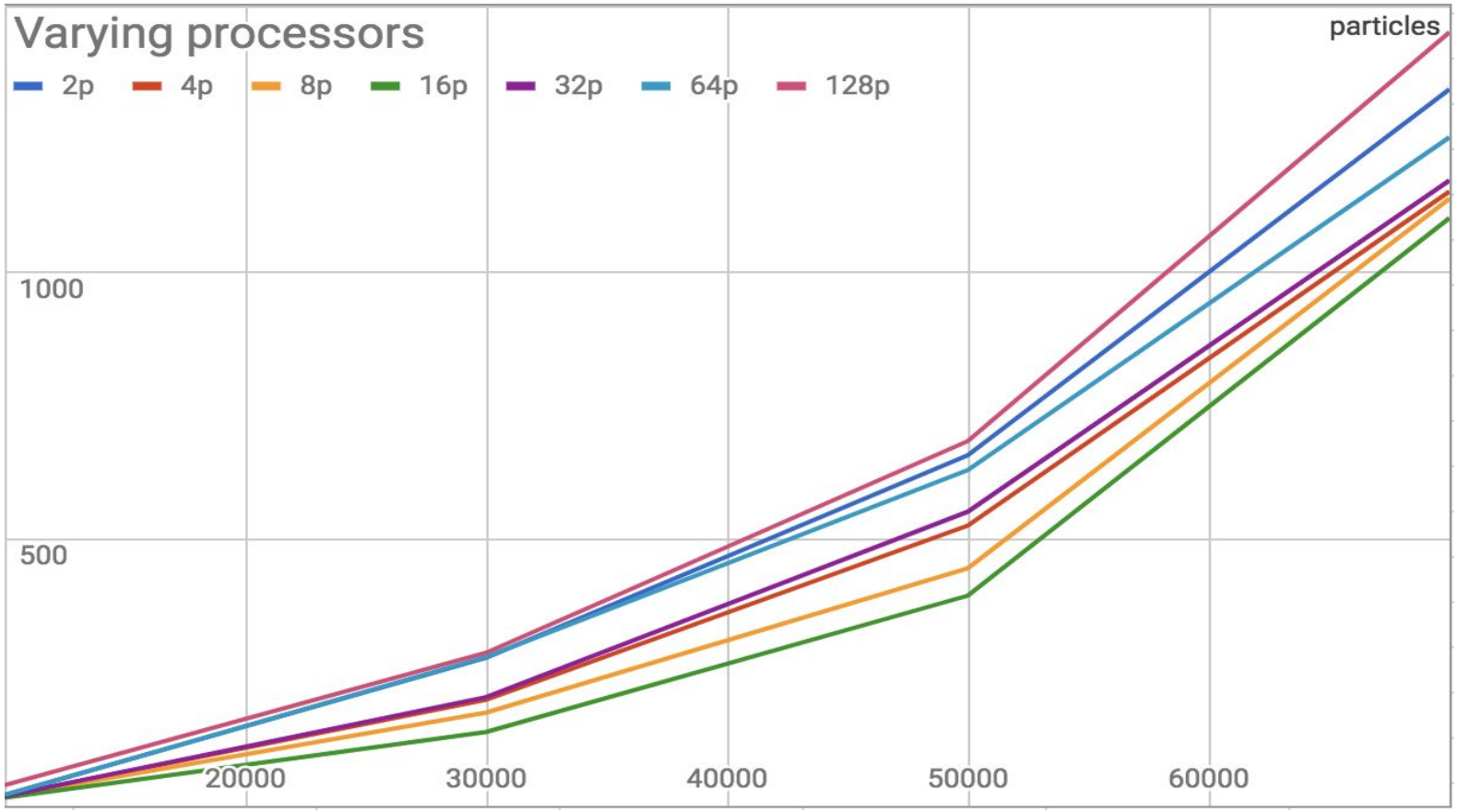
Sequential



particles	computation time(s)
10000	21.88242245
30000	282.5355
50000	663.53444
70000	1564

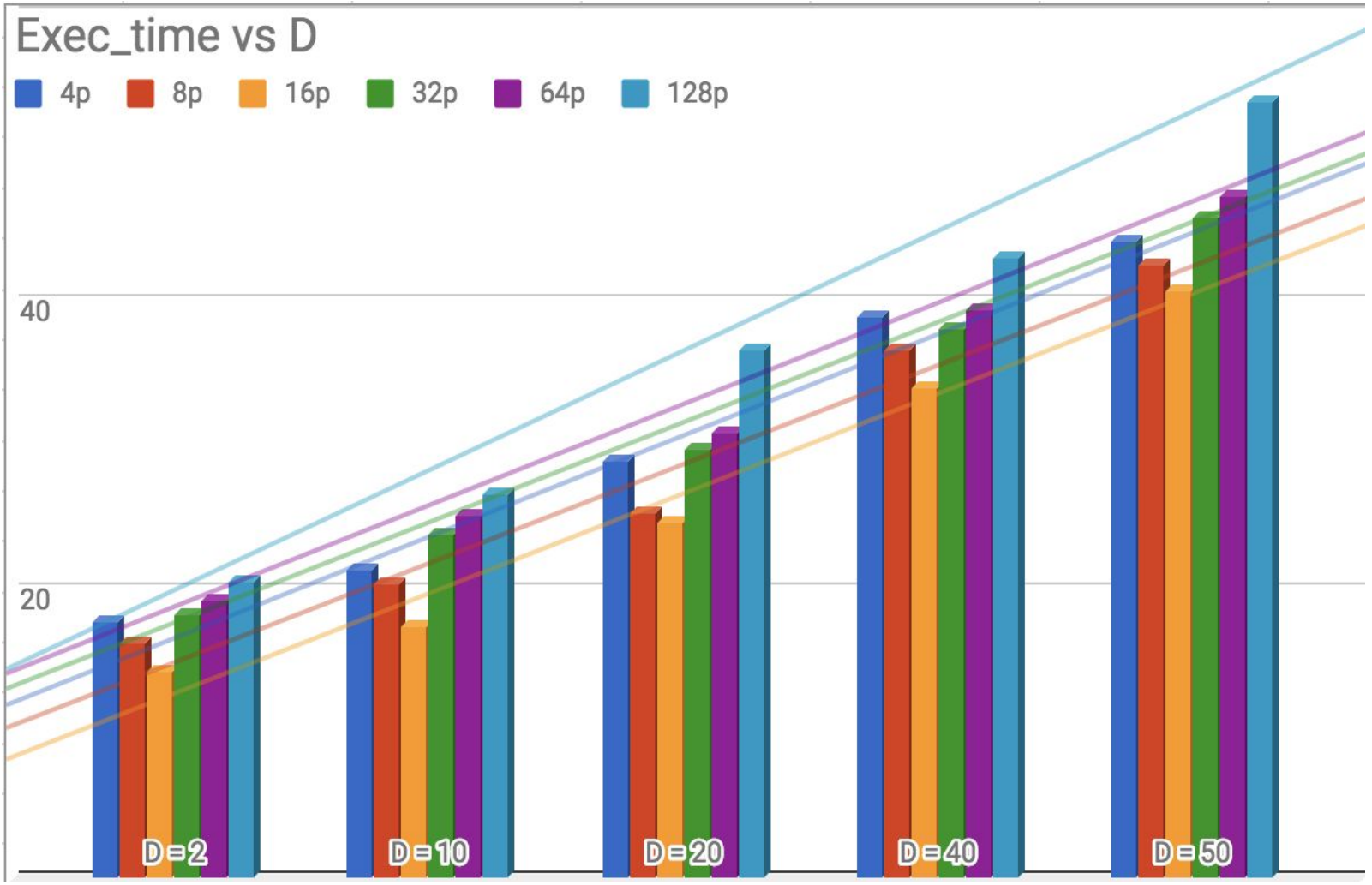
Running time (in seconds) of Parallel approach

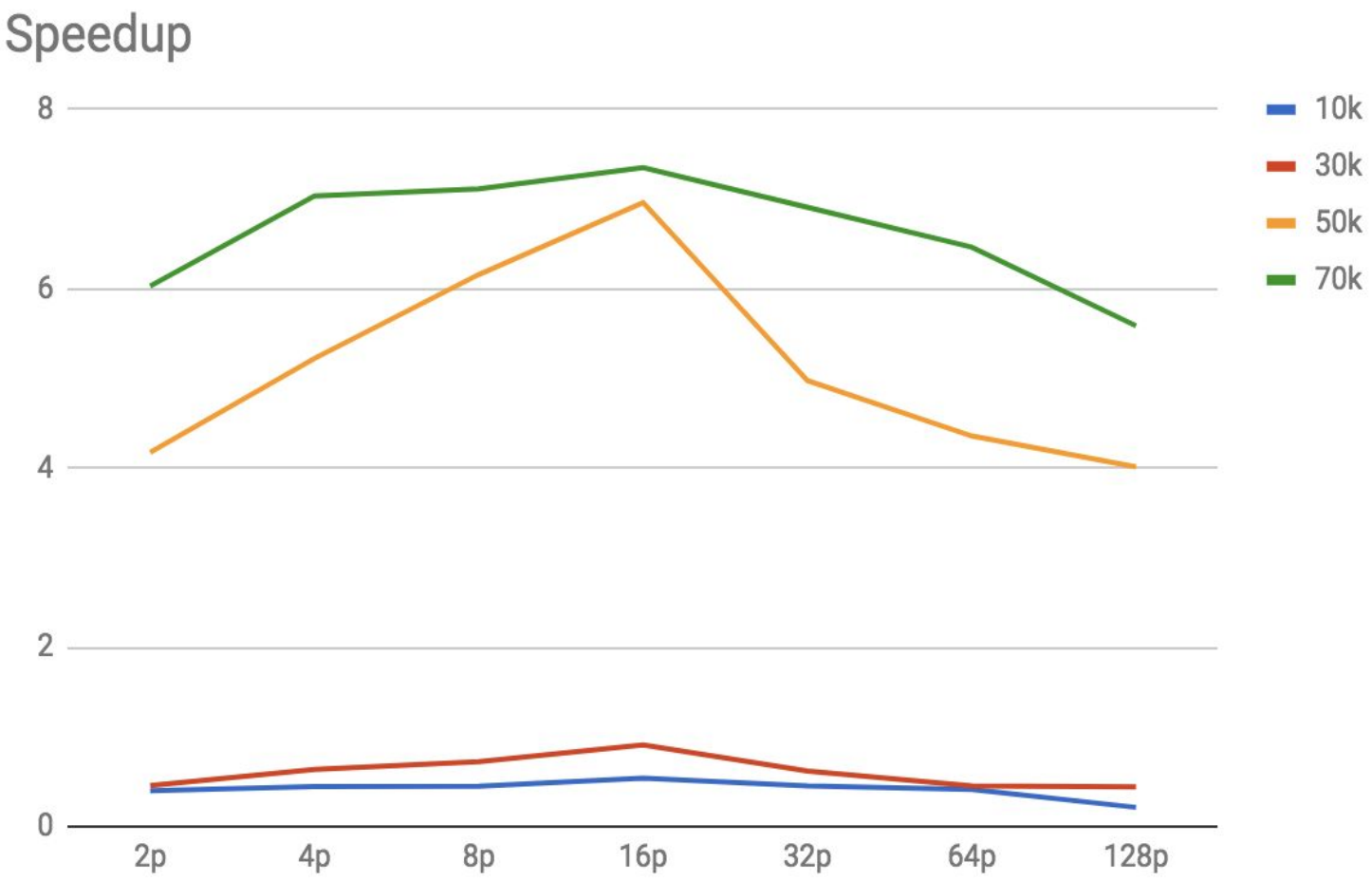
particles	2p	4p	8p	16p	32p	64p	128p
10000	20.7460062	18.5492422	18.2915851	15.2957604	18.2008698	19.9455821	38.7066314
30000	277.862277	199.08043	174.944094	138.599893	204.009193	279.394358	287.109496
50000	657.274744	525.407571	445.488469	394.111021	551.52699	629.748321	683.916605
70000	1343.82011	1151.88548	1138.64525	1102.11094	1172.64207	1253.60185	1450.48639

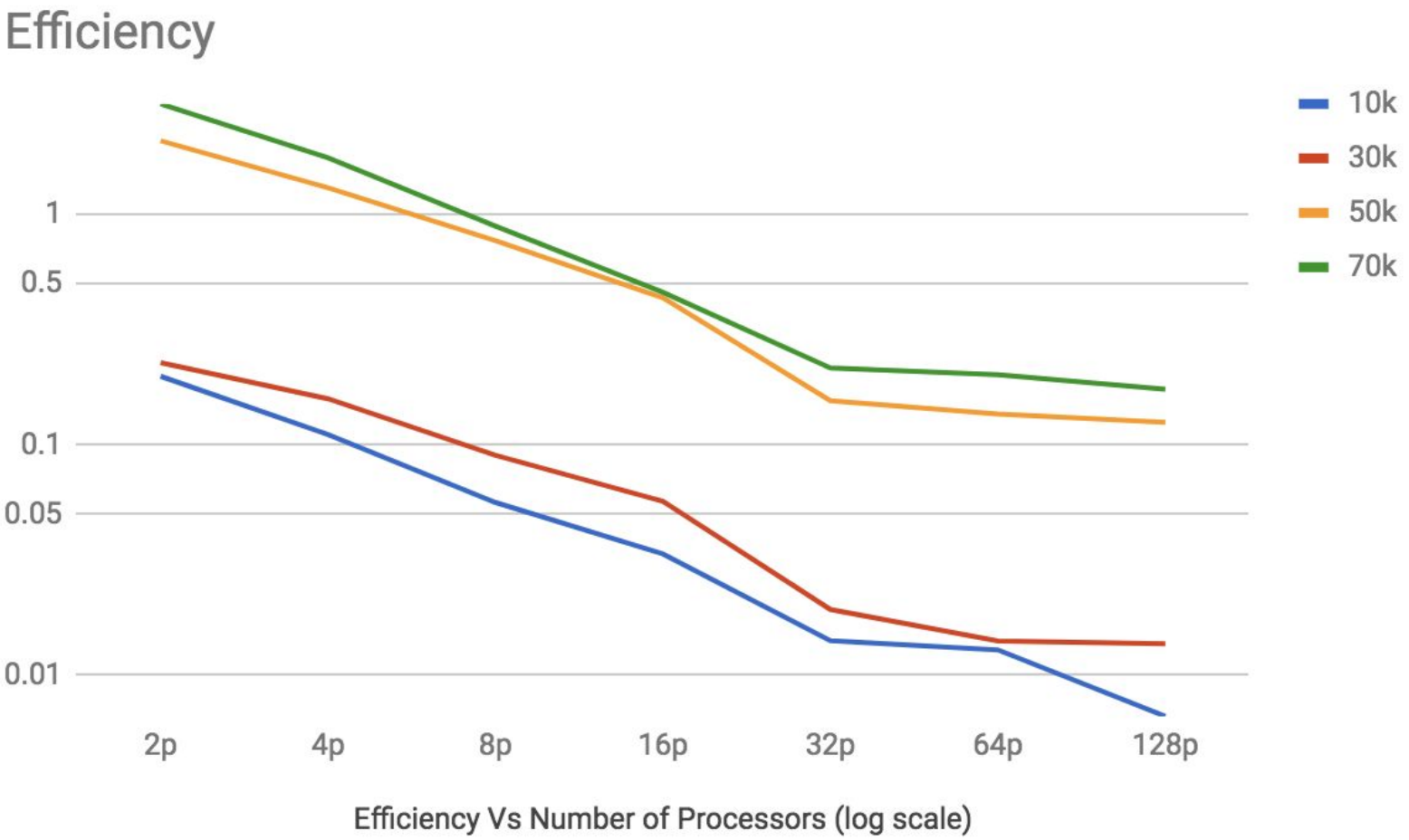

Varying processors
particles
2p
4p
8p
16p
32p
64p
128p
1000
500
20000
30000
40000
50000
60000

Variation of Dimension [5]

	4p	8p	16p	32p	64p	128p
D = 2	17.7460062	16.2915851	14.2957604	18.2008698	19.2008698	20.4253534
D = 10	21.3555522	20.3423927	17.3201395	23.7950679	25.1438536	26.6358335
D = 20	28.8620587	25.2132479	24.5571379	29.6636317	30.8706279	36.5882903
D = 40	38.8073485	36.5390352	34.0007512	38.0445791	39.4065166	42.8957243
D = 50	44.1421378	42.5964413	40.5907378	45.7988242	47.1784433	53.7874253

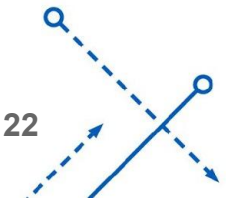

Exec_time vs D
4p
8p
16p
32p
64p
128p
40
20
D = 2
D = 10
D = 20
D = 40
D = 50


Speedup
8
6
4
2
0
2p
4p
8p
16p
32p
64p
128p
10k
30k
50k
70k


Efficiency
10k
30k
50k
70k
1
0.5
0.1
0.05
0.01
2p
4p
8p
16p
32p
64p
128p
Efficiency Vs Number of Processors (log scale)

Observation/Learnings/Future Scope

- Working with 16 processors was best.
- Increase on processor decrease running time but only upto certain number processors.
- Hands on parallel computing, MPI programming (debugging)
- OpenMP, CUDA implementation
- MPI Gather, Scatter, Allgather instead of MPI_Recv, MPI_Send functions
- Optimize implementation



References

- [1] : J. Kennedy and R. Eberhart, “Particle swarm optimization.” Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995 (in press).
- [2] : Schutte, J. F., Reinbolt, J. A., Fregly, B. J., Haftka, R. T., and George, A. D., Parallel Global Optimization with the Particle Swarm Algorithm, International Journal of Numerical Methods in Engineering (accepted), 2003
- [3] : R. J. Lavery, “Throughput optimization for wireless data transmission,” in M. S. Thesis, Polytechnic University,2001.
- [4] : A. Balamurugan, “Efficient fitness based routing protocol in wireless sensor networks,” in ICTACT Journal on Communication Technology,vol. 05,2014.
- [5] : T. Hendtlass, “Particle Swarm Optimisation and high dimensional problem spaces,” IEEE Congress on Evolutionary Computation, pp. 1988-1994, 2009
- https://ubccr.freshdesk.com/support/solutions/articles/13000026245-tutorials-and-training-documents

University at Buffalo The State University of New York

Thank you