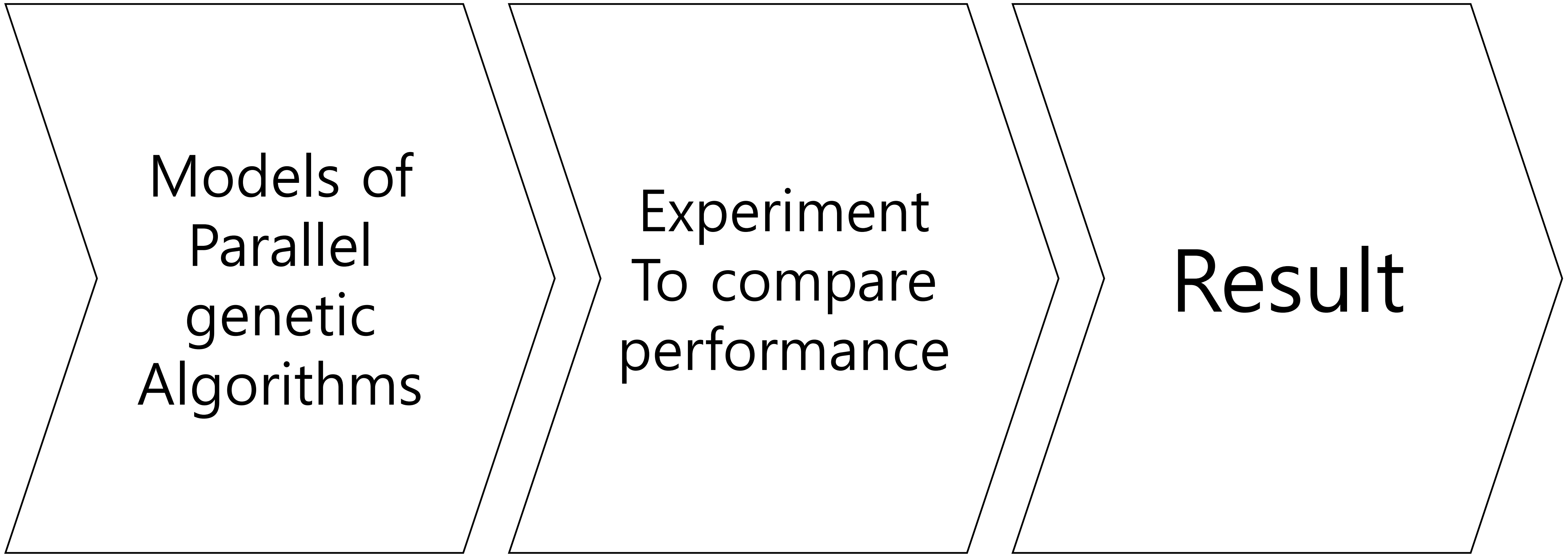
Serial and Parallel Genetic Algorithms as Function Optimizers

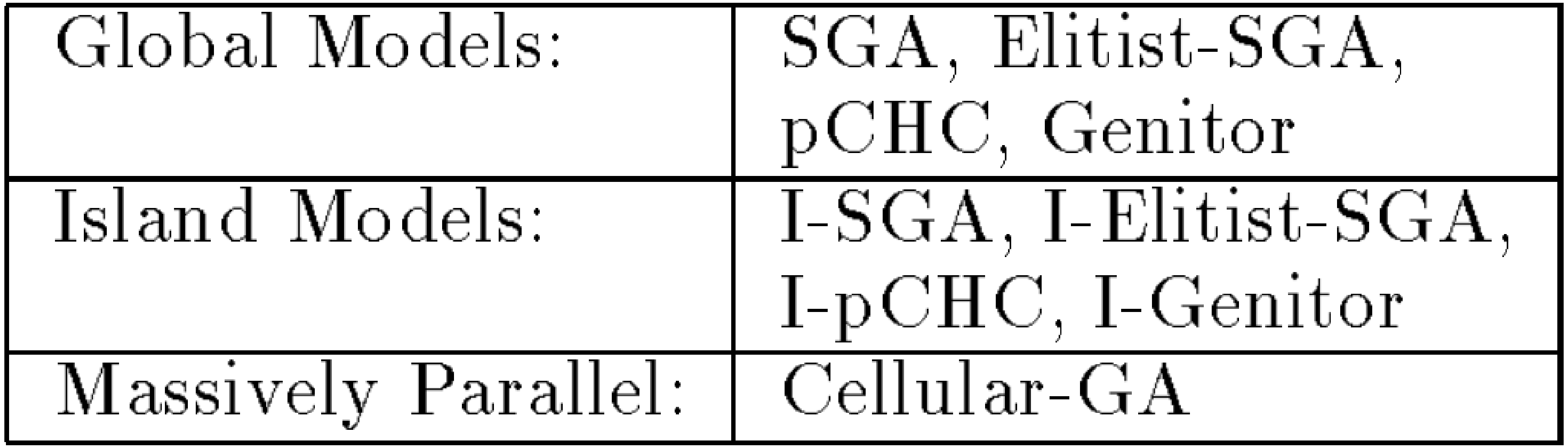
Introduction

- Overall Introduction
- Parallel genetic Algorithm model
- Performance measurement
- Test Suite
- Result and conclusion


Models of Parallel genetic Algorithms
Experiment To compare performance
Result

Parallel genetic Algorith?

- Different to serial genetic algorithm, each operation is shared by multiple processors and executed simultaneously
- Can separate to 3 categories



Global Models:	SGA, Elitist-SGA, pCHC, Genitor
Island Models:	I-SGA, I-Elitist-SGA, I-pCHC, I-Genitor
Massively Parallel:	Cellular-GA

SGA and Elitist SGA

- Simple genetic algorithm
- Use tornament selection to facilitate parallelism
- Every two slots of each new generation are filled by the offispring of two selected parents from the previous generation.
- In Elitist SGA, best individual is always placed in the next generation

pCHC

- Parallelized version of the CHC algorithm
- Similar to SGA, except best n strings are extracted from both generation t and t+1.
- Parents are paired through ‘incest prevention’.
- tournament selection is used, but to select pairs of individuals which are relatively dissimilar. After recombination, the offspring in generation t + 1 are compared against two particular elements from generation t, and the best two of the four are retained.
- This algorithm does not guarantee best n out of 2n individuals, but at least best two individuals will survive.

Genitor

- Rank based algorithm
- Two parents are selected and a single offspring is produced that displaces the worst member of the population.
- Population in sorted order. The winners of a small tournament recombine and the offspring replaces the loser of the tournament if the offspring has a higher fitness.

Island SGA and Elistist Island-SGA

- Island model involves running several single population genetic algorithms in parallel.
- Each Island is an SGA with its own subpopulation.
- Migration between islands uses a ring topology, and a single individual is chosen for migration by tournament selection, where the losing individual is replaced by the winning individual from the adjacent subpopulation.

Island-pCHC and Island-Genitor.

- These are straightforward insertions of pCHC and Genitor into the Island model.

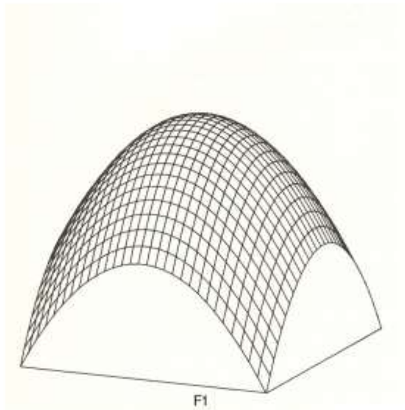
Cellular Genetic Algorithms.

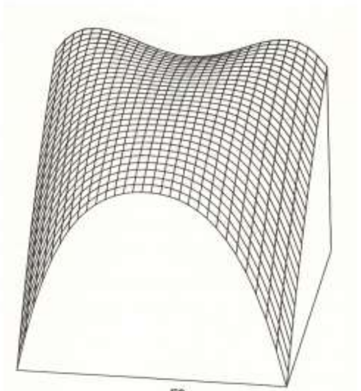
- Cellular Genetic Algorithms assign one individual per processor, and mating is limited to a deme (neighborhood) near the individual.
- strings reside on a two-dimensional grid, and demes consist of the four individuals directly above, below, left, and right of each individual.
- The best of these four is selected and crossover is performed with the individual.

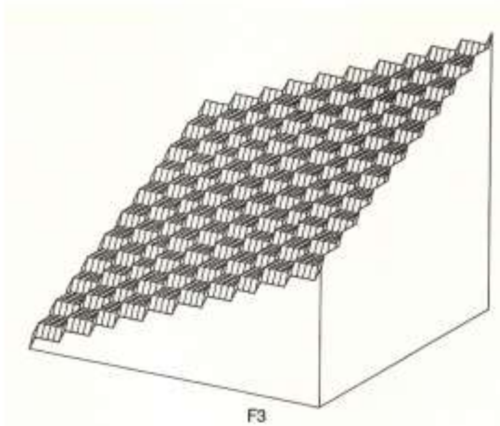
Performance Measurement

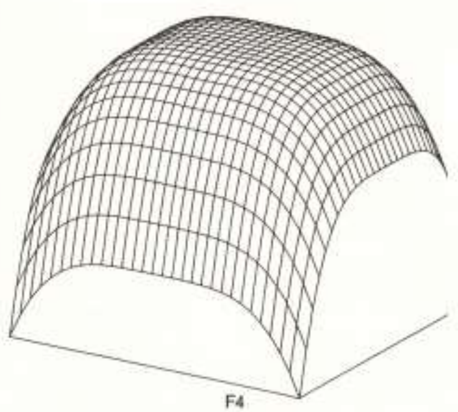
- We use number of function evaluations as the base work unit, and express all computation times in terms of SGA generations .
- Genitor performs two function evaluations per generation, it requires n/2 (where n is the population size) generations in Genitor to perform the same number of function evaluations as one SGA generation.
- cellular genetic algorithm performs two function evaluations for each location in the 2-D grid every generation. Thus in one generation it performs twice the number of function evaluations as SGA. Therefore we multiply the number of generations performed by the cellular genetic algorithm by two.

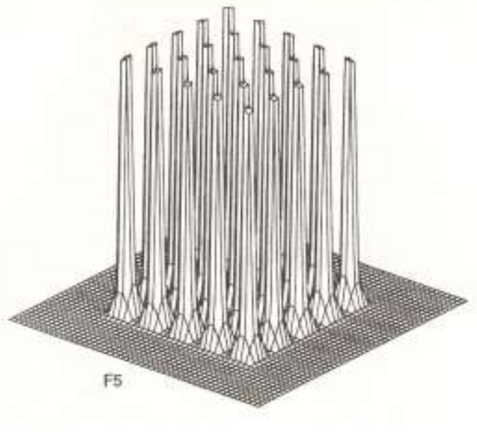
DeJONG TEST SUITE


F1

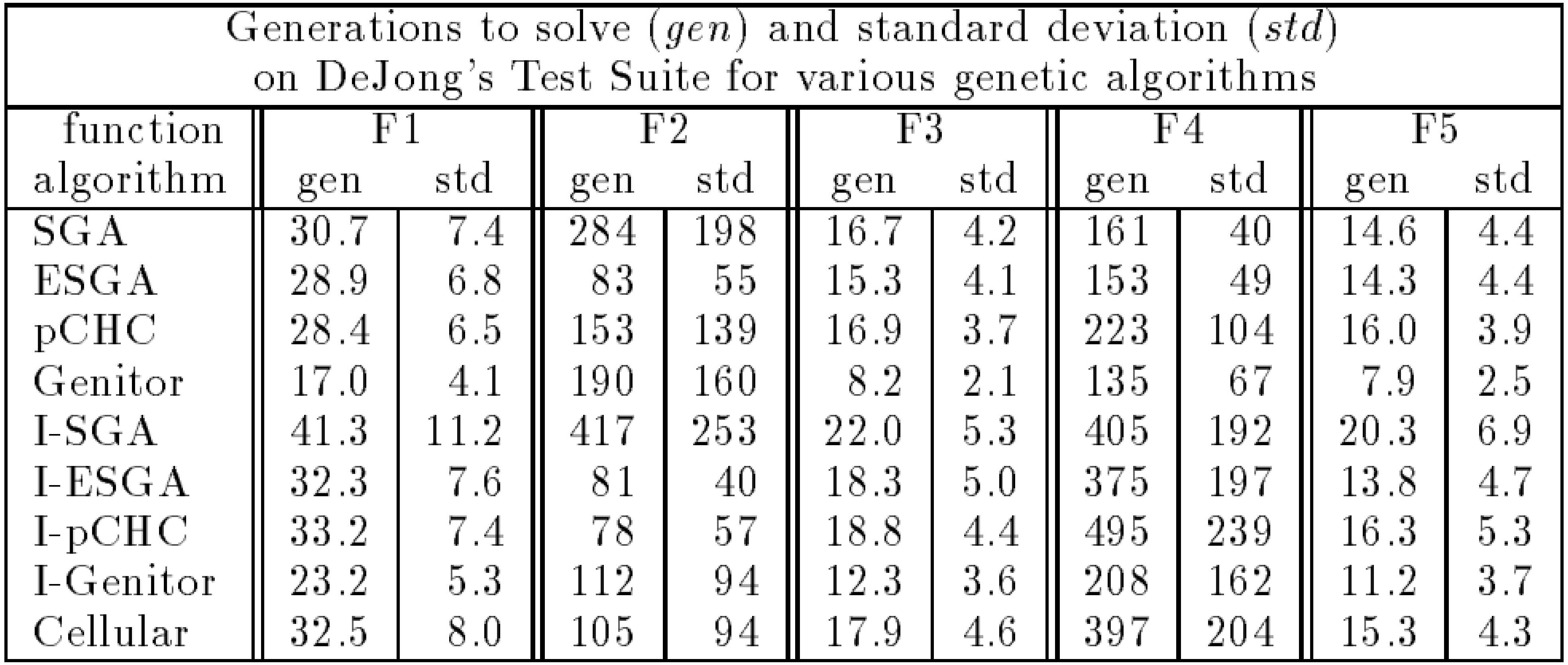

F2


F3


F4


F5

DeJONG TEST SUITE



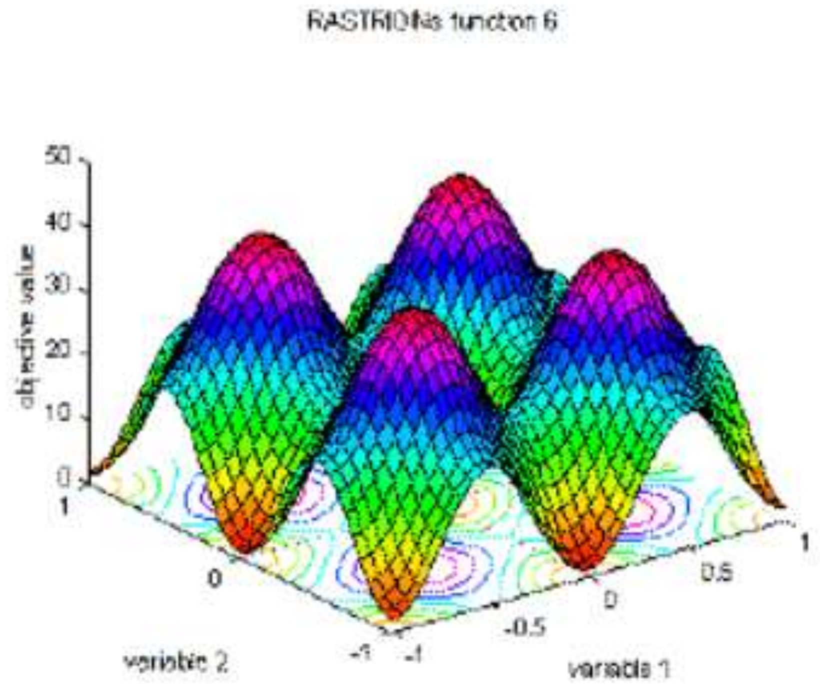
Generations to solve (*gen*) and standard deviation (*std*) on DeJong's Test Suite for various genetic algorithms

function / algorithm	F1		F2		F3		F4		F5	
	gen	std	gen	std	gen	std	gen	std	gen	std
SGA	30.7	7.4	284	198	16.7	4.2	161	40	14.6	4.4
ESGA	28.9	6.8	83	55	15.3	4.1	153	49	14.3	4.4
pCHC	28.4	6.5	153	139	16.9	3.7	223	104	16.0	3.9
Genitor	17.0	4.1	190	160	8.2	2.1	135	67	7.9	2.5
I-SGA	41.3	11.2	417	253	22.0	5.3	405	192	20.3	6.9
I-ESGA	32.3	7.6	81	40	18.3	5.0	375	197	13.8	4.7
I-pCHC	33.2	7.4	78	57	18.8	4.4	495	239	16.3	5.3
I-Genitor	23.2	5.3	112	94	12.3	3.6	208	162	11.2	3.7
Cellular	32.5	8.0	105	94	17.9	4.6	397	204	15.3	4.3

Table 1: Performance of nine GAs on DeJong's test suite

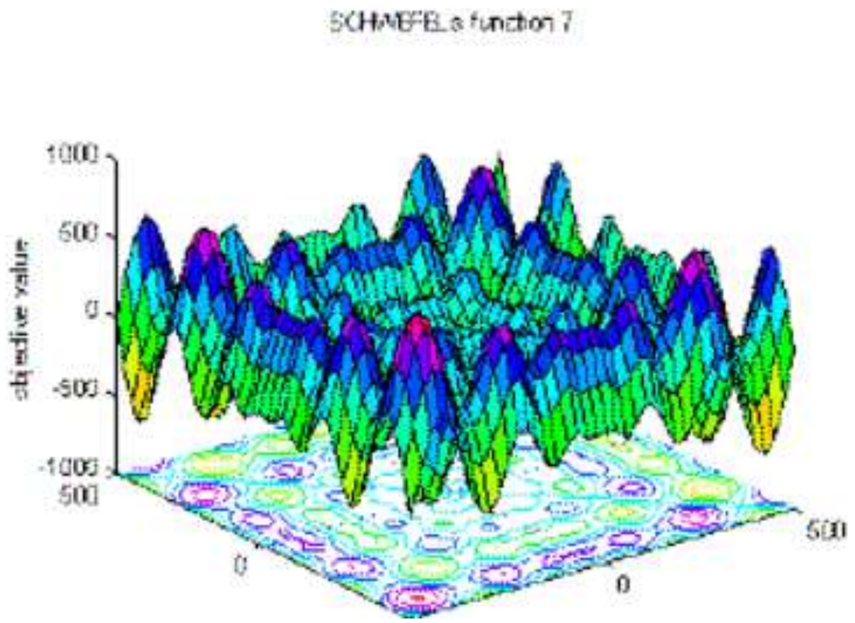
RASTRIGIN(f6), SCHWEFEL(f7), AND GRIEWANGK(f8) FUNCTION

- Rastrigin's function is based on function 1 with the addition of cosine modulation to produce many local minima. Thus, the test function is highly multimodal. However, the location of the minima are regularly distributed.



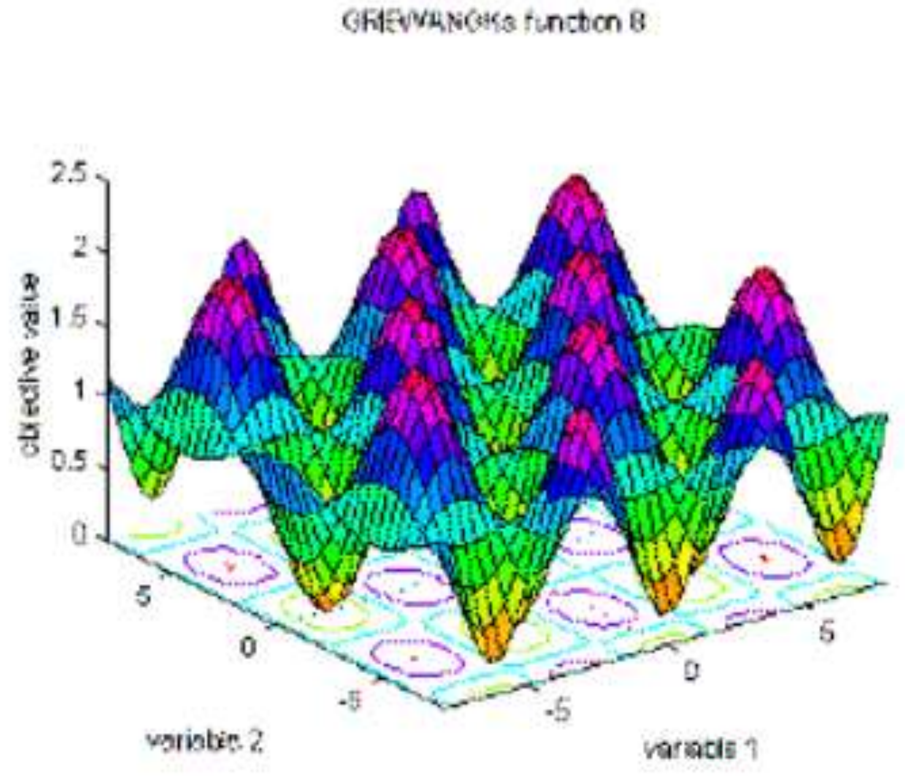
RASTRIGIN(f6), SCHWEFEL(f7), AND GRIEWANGK(f8) FUNCTION

- deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction.



RASTRIGIN(f6), SCHWEFEL(f7), AND GRIEWANGK(f8) FUNCTION

- Griewangk's function is similar to Rastrigin's function. It has many widespread local minima. However, the location of the minima are regularly distributed.



UGLY 3 AND 4-BIT DECEPTIVE FUNCTIONS

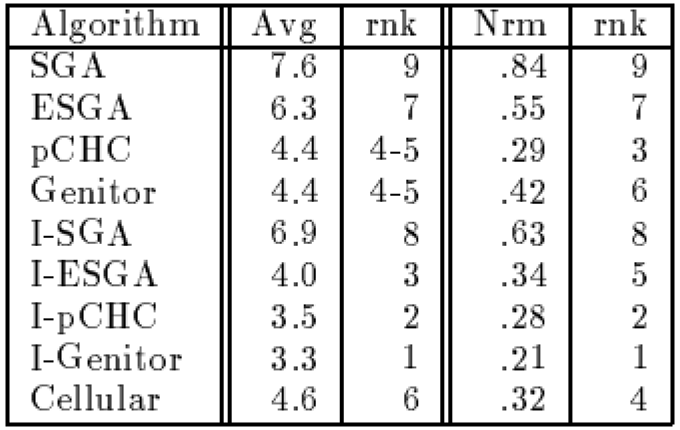
- The ugly 3-bit problem (D3) is a 30-bit artificially constructed problem
- ugly 4-bit problem (D4) is a similarly constructed 40-bit problem in which ten fully-deceptive 4-bit subproblems are interleaved
- These problems isolate interactions in the hyperplane sampling abilities of a genetic algorithm as well as the linkage between bits
- We execute each run for 5000 generations and report the number of runs in which the global optimum was found

ZERO-ONE KNAPSACK PROBLEMS

- The zero-one knapsack problem is defined as follows. Given n objects with positive weights Wi and positive profits Pi, and a knapsack capacity M, determine a subset of the objects represented by a bit vector X
- We use 20 object and 80 object problem

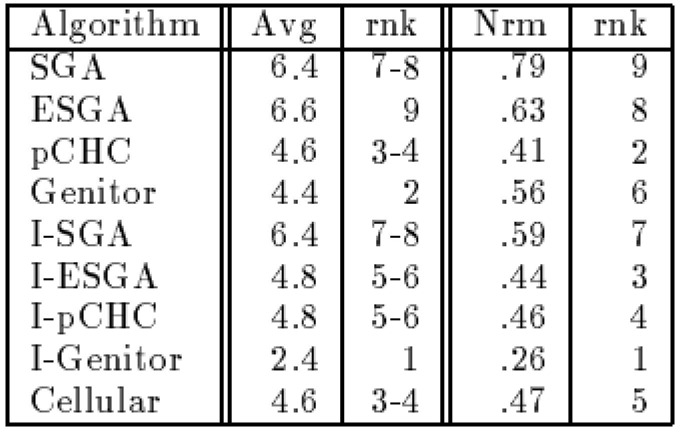
$$\sum_{i=1}^{n} X_i W_i \leq M \quad and \quad \sum_{i=1}^{n} X_i P_i \quad maximal.$$

- Non elitist algorithm perform worth(SGA and I-SGA)
- Parallel algorithm peform better in harder function



Algorithm	Avg	rnk	Nrm	rnk
SGA	7.6	9	.84	9
ESGA	6.3	7	.55	7
pCHC	4.4	4-5	.29	3
Genitor	4.4	4-5	.42	6
I-SGA	6.9	8	.63	8
I-ESGA	4.0	3	.34	5
I-pCHC	3.5	2	.28	2
I-Genitor	3.3	1	.21	1
Cellular	4.6	6	.32	4

Table 6: Performance of nine GAs on hard problems
F2, F4, Rastrigin, schwelfel, Griewangk, 20 object knapsack



Algorithm	Avg	rnk	Nrm	rnk
SGA	6.4	7-8	.79	9
ESGA	6.6	9	.63	8
pCHC	4.6	3-4	.41	2
Genitor	4.4	2	.56	6
I-SGA	6.4	7-8	.59	7
I-ESGA	4.8	5-6	.44	3
I-pCHC	4.8	5-6	.46	4
I-Genitor	2.4	1	.26	1
Cellular	4.6	3-4	.47	5

Table 7: Performance on long bitstring problems
F4, Rastrgin, Schwelfel, Griewangk, and 80 knapsack

Algorithm	F1	F2	F3	F4	F5	F6	F7	F8	D3	D4	K20	K80	Avg	rnk
SGA	5	8	4	3	5	8	9	9	8	8	8	3	6.5	8
ESGA	4	3	3	2	4	6	8	8	9	7	7	9	5.8	7
pCHC	3	6	5	5	7	3	5	6	5	2	3	4	4.5	4
Genitor	1	7	1	1	1	9	7	4	3	3	1	1	3.3	2
I-SGA	9	9	9	8	9	7	6	3	7	9	6	8	7.5	9
I-ESGA	6	2	7	6	3	4	4	5	1	1	9	5	4.4	3
I-pCHC	8	1	8	9	8	5	1	2	2	4	5	7	5.0	5
I-Genitor	2	5	2	4	2	2	3	1	4	5	2	2	2.8	1
Cellular	7	4	6	7	6	1	2	7	6	6	4	6	5.2	6

Table 4: Ranking of performance of nine GAs on test suite

ZERO-ONE KNAPSACK PROBLEMS

- overall, elitist strategies perform better than non-elitist ones.
- cellular, steadystate (i.e., Genitor), and CHC approaches are at least as effective as elitist versions of the standard genetic algorithm
- The performance of SGA is relatively poor compared to the other alternative algorithms examined in this study.
- parallel genetic algorithm using some form of restricted selection and mating based on locality that are executed serially often yield better performance than single population implementations with global “panmictic" mating.