# A COMPARISON OF PARTICLE SWARM OPTIMIZATION AND THE GENETIC ALGORITHM

Sal and Xianshun

## Overview

Particle swarm optimization (PSO)

Genetic algorithm (GA)

Comparison metrics and hypothesis-test

Result and conclusion

## Particle Swarm Optimization

- Invented in mid 1990s to simulate motions of bird swarms
- Randomly generated solutions act as a starting point, and the swarm moves by data that is shared between the swarm
- Inspired by the ability of bird flocks to survive predators and find food with information sharing.

## Genetic Algorithm

- Created mid 1970s
- Inspired by principles of genetics and evolution
- Uses principal of “survival of the fittest” to make an optimized swarm over generations

## Particle Swarm Optimization

- Three steps
- Step One:
  - Positions x and velocities k of each unit is generated in a bounds x(min)-x(max)

$$\mathbf{x}_0^i = \mathbf{x}_{\min} + rand\left(\mathbf{x}_{\max} - \mathbf{x}_{\min}\right)$$

$$\mathbf{v}_0^i = \frac{\mathbf{x}_{\min} + rand\left(\mathbf{x}_{\max} - \mathbf{x}_{\min}\right)}{\Delta t} = \frac{\text{position}}{\text{time}}$$

## Particle Swarm Optimization

- Step 2:
  - Update velocities of all particles
  - Determine particle with best global position, and best positions of all particles over time
  - These values, along with velocity, are used to calculate the direction each particle should move in

velocity of particle $i$ at time $k$+1

$$\mathbf{v}_{k+1}^i = \underbrace{w\,\mathbf{v}_k^i}_{\text{current motion}} + \underbrace{c_1\, rand \frac{\left(\mathbf{p}^i - \mathbf{x}_k^i\right)}{\Delta t}}_{\text{particle memory influence}} + \underbrace{c_2\, rand \frac{\left(\mathbf{p}_k^g - \mathbf{x}_k^i\right)}{\Delta t}}_{\text{swarm influence}}$$

inertia factor
range: 0.4 to 1.4

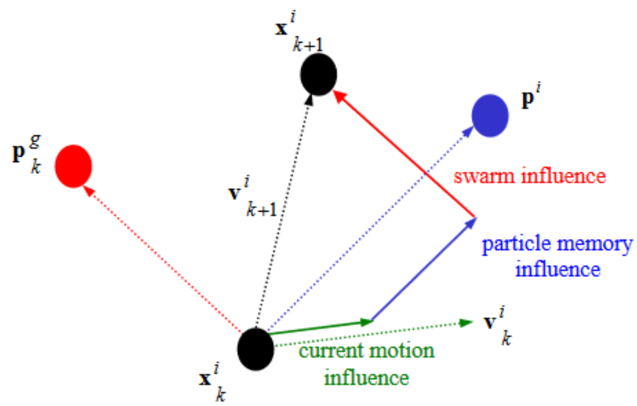
self confidence
range: 1.5 to 2

swarm confidence
range: 2 to 2.5

## Particle Swarm Optimization

- Step 3:
  - Update each particle's position

$$\mathbf{x}^i_{k+1} = \mathbf{x}^i_k + \mathbf{v}^i_{k+1} \Delta t$$




## The Genetic Algorithm

- Starts from randomly generated population that evolve over generations
- Uses three operators:
  - Selection operator acts a survival of the fittest
  - Crossover operator mimics propagation of of population
  - Mutation operator gives diversity to population and allows an effective search without getting stuck

## Comparison Metrics and Hypothesis Testing

T - test (Hypothesis test) with 5 steps:

| Action | The true situation may be | |
|---|---|---|
| | $H_\circ$ is true | $H_\circ$ is false |
| accept $H_\circ$, reject $H_a$ | $(1-\alpha)$ significance level | $\beta$ [type II error] |
| reject $H_\circ$, accept $H_a$ | $\alpha$ [type I error] | $(1-\beta)$ power of test |
| Sum | 1 | 1 |

## Comparison Metrics and Hypothesis Testing

Step 1: define H0 and H1

Step 2: decided on desired values for α, β and n to find out corresponding $t_{critical}$ ( t distribution)

Step 3: evaluate n random samples and evaluate statistical characteristics

Step 4: calculate the t-value of the null hypothesis

Step 5: compare the calculated t-value to the tabulated $t_{critical}$. If t <= $t_{critical}$, then accept null hypothesis with (1-α) confidence level

## Comparison Metrics and Hypothesis Testing

Effectiveness test (finding the true global optimum):

**Effectiveness Test**

Objective to test whether $H_a: \ \mu_{Q_{sol}} > 99\%$

$H_\circ: \ \mu_{Q_{sol}} \le 99\%$

$$t = \frac{\overline{Q}_{sol} - 99\%}{s(\overline{Q}_{sol})} \qquad Q_{sol} = \left|\frac{\text{solution - known solution}}{\text{known solution}}\right|\%$$

taking $\alpha = 1\%$, $\beta = 1\%$, and $n = 10$

This is a one sided test of significance of a mean (table 6.10, Reference 7) $\rightarrow t_{critical} = 2.0$

## Comparison Metrics and Hypothesis Testing

Efficiency (computational cost) result:

**Efficiency Test**

Objective to test whether $H_a: \ \text{PSO } \mu_{N_{eval}} < \text{GA } \mu_{N_{eval}}$

$H_\circ: \ \text{PSO } \mu_{N_{eval}} \ge \text{GA } \mu_{N_{eval}}$

$$t = \frac{\text{GA } \overline{\mu}_{N_{feval}} - \text{PSO } \overline{\mu}_{N_{feval}}}{\overline{s}(x)\sqrt{1/n_{GA} + 1/n_{PSO}}} \quad \text{where} \quad \overline{s}(x) = \sqrt{\frac{(n_{GA}-1)s_{GA}^2 + (n_{PSO}-1)s_{PSO}^2}{n_{GA} + n_{PSO} - 2}}$$

taking $\alpha = 1\%$, $\beta = 1\%$, and $n_{PSO} = n_{GA} = 10$

This is a one sided test of significance of comparison of two means (table 6.11, Reference 7) $\rightarrow t_{critical} = 2.5$

Result: PSO is significantly more efficient than the GA

## Benchmark Test Problems

The Banana (Rosebrock) Function (has a known global minimum at [1,1]:

$$\text{Minimize } f(\mathbf{x}) = 100\left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2$$

The Eggcrate Function (has lower and upper bounds of [-2π,2π], and global minimum at [0,0]:

$$\text{Minimize } f(\mathbf{x}) = x_1^2 + x_2^2 + 25\left(\sin^2 x_1 + \sin^2 x_2\right)$$

Golinski's Speed Reducer:

(Details are in a reference link: http://mdob.larc.nasa.gov/mdo.test/class2prob4.htm, which is broken

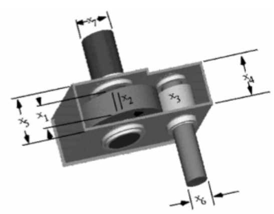


**Figure 2. Golinski's Speed Reducer.**

## Space System Problems

Telescope Array Configuration (Scientific Application):

Array of small distributed telescope to form a large single costly telescope:

Degree of cover in the uv plane(x, y are positions) : $u_{ij} = x_j - x_i \quad , \quad v_{ij} = y_j - y_i$

Number of visibility point: $N_{uv} = N_{stations}\left(N_{stations} - 1\right)$

Metric for the uv plane coverage as the distance between all uv points:

$$\text{Maximize } uv \text{ coverage} = \sum_{i,j,k,l} \sqrt{\left(u_{ij} - u_{kl}\right)^2 + \left(v_{ij} - v_{kl}\right)^2}$$

## Space System Problems

Communication Satellite Reliability-Based Design:

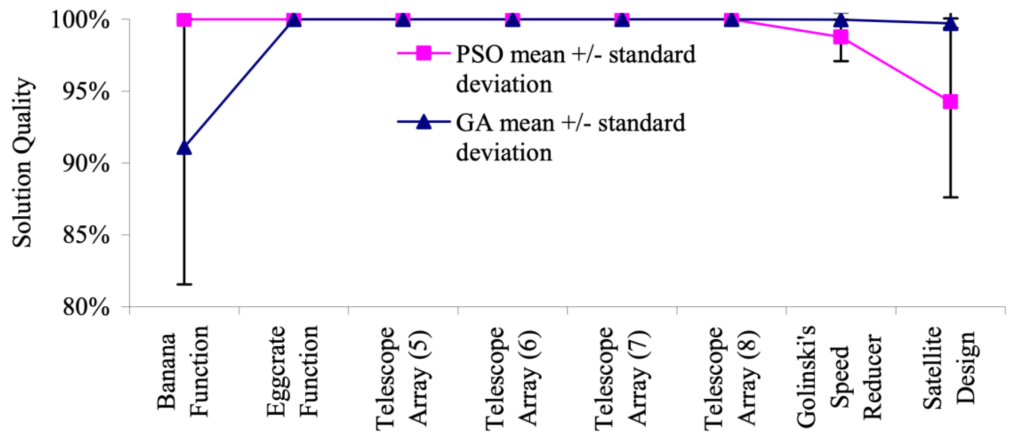
Designing the payload and bus subsystems of a commercial communication Geosynchronous satellite with given payload requirements. The goal is to minimize the spacecraft overall launch mass.

## Results

**Table 2: Calculated *t*-values for the effectiveness and efficiency hypotheses tests**

| | Effectiveness Test, $t_{critical} = 2.0$ Calculated *t*-value | | Efficiency Test, $t_{critical} = 2.5$ Calculated *t*-value |
|---|---|---|---|
| | PSO | GA | |
| 1- Banana Function | 8404.05 | -2.49 | 1.9980 |
| 2- Eggcrate Function | 312420.20 | 5214.34 | 7.2743 |
| 3- Telescope Array (5 Stations) | ∞ | ∞ | 19.2995 |
| 4- Telescope Array (6 Stations) | ∞ | ∞ | 16.8674 |
| 5- Telescope Array (7 Stations) | ∞ | ∞ | 20.9451 |
| 6- Telescope Array (8 Stations) | ∞ | ∞ | 23.7893 |
| 7- Golinski's Speed Reducer | -0.39 | 1251.28 | 10.3079 |
| 8- Satellite Design | -2.13 | 6.47 | 19.6993 |

## Results




## Results

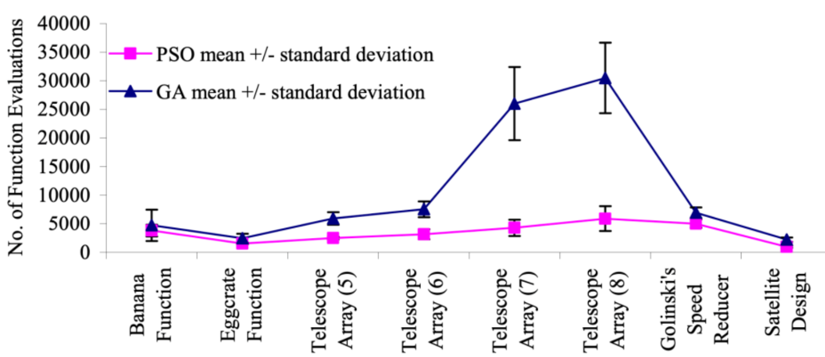
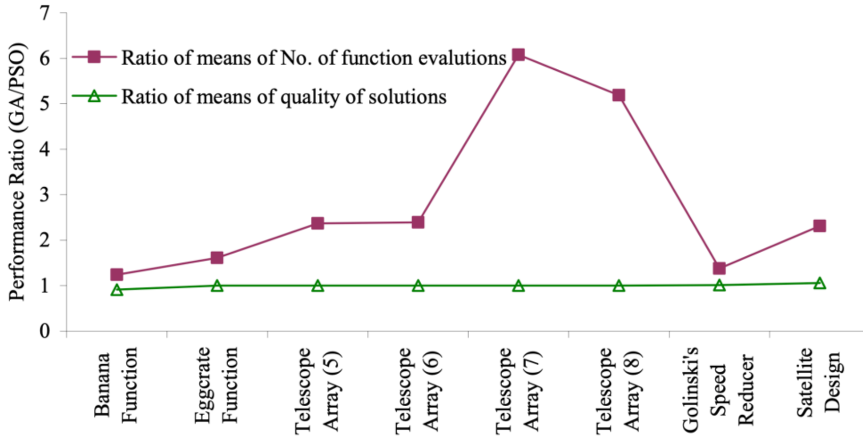



**Figure 5. The mean computational effort of the solutions obtained by PSO and the GA using 10 run samples for eight test problems.**

## Results




## Conclusion

PSO is similar to GA:

- They are both population-based search approach
- They both depend on information sharing among the population members

PSO is more efficient than GA based on the results from t-tests

Questions?