# Parameter Adaptive Differential Evolution for Multi-modal Function Optimization


Jingqiao Zhang and Arthur C. Sanderson
Center for Automation Technologies and Systems


## Parameter Adaptive Differential Evolution for Multi-modal Function Optimization


Jingqiao Zhang
Arthur C. Sanderson

Center for Automation Technologies and Systems
Rensselaer Polytechnic Institute
April, 2008

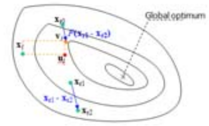

## Evolutionary algorithms

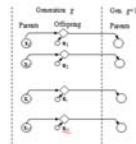
- Darwinian Evolution
  - The principle of natural selection: survival of the fittest
  - Given a population of individuals which have experienced
  - Operate on a population of solutions simultaneously random or probabilistic perturbation (mutation and crossover), the nature favors those individuals with better fitness for survival and further evolution.

- Evolutionary algorithms mimic Darwinian Evolution
  - Operate on a population of solutions simultaneously
  - In each iteration (called a generation in EA), it undergoes a loop of mutation, crossover, and selection operations
- EA operations
  - **Mutation**: Add random perturbation to current solutions
  - **Crossover**: Exchange information between solutions
  - **Selection**: Prefer the solutions with better fitness values (e.g. the value of the objective function) and select them for further evolution.


## Differential Evolution

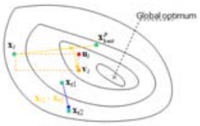
- Differential evolution is a recent branch of EA
  - It has been shown as a simple yet powerful approach for global optimization for many real-world applications such as in engineering, financial, and scientific areas.
- Start from a population of random solutions (vectors) in the search space: $\{x_1, x_2, \ldots, x_N\}$. Then, the algorithm enters a loop of evolutionary operations
  - mutation $v_i = x_{r0} + F(x_{r1} - x_{r2})$ where r0, r1 and r2 are three random indices
  - crossover $u_{j,i} = \begin{cases} v_{j,i} & \text{if } \mathrm{rand}_j(0,1) \le CR \text{ or } j = j_{rand} \\ x_{j,i} & \text{otherwise,} \end{cases}$
  - selection $x_i = \begin{cases} u_i & \text{if } f(u_i) < f(x_i) \\ x_i & \text{otherwise,} \end{cases}$



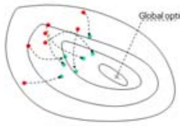




## JADE: Parameter Adaptive Differential Evolution

- JADE (greedy mutation strategy + parameter adaptation)
  - Greedy mutation strategy (the utilization of direction information)
    - Best-solution information: DE/current-to-p-best/1
    - Inferior-solution information: archive the inferior solutions that were previously searched
  - Parameter adaptation
    - Update parameter values according to the feedback from the evolutionary search
    - Principle: Better parameter values tend to generate individuals that are more likely to survive, and thus can be propagated to more offspring.




**Best-solution** information




- Current solutions
- Archived inferior solutions

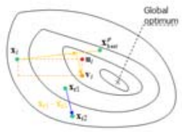
## JADE: Parameter Adaptive Differential Evolution

- JADE: greedy DE (Differential Evolution) + PA (Parameter Adaptation)
  - DE: differential Mutation + Crossover + one-to-one Selection
  - PA: Adapt control parameters ($F$ and $CR$) along with the evolutionary search
- At each generation, for each parent vector $x_i$ in the population
  - $x_{r1}$ and $x_{r2}$: two other vectors in the population; $x^p_{best}$: one of the top 100p% vectors
    - Mutation $v_i = x_i + F_i(x^p_{best} - x_i) + F_i \cdot (x_{r1} - x_{r2})$
    - Crossover $u_{j,i} = \begin{cases} v_{j,i} & \text{if } \mathrm{rand}_j(0,1) \le CR_i \text{ or } j = j_{rand} \\ x_{j,i} & \text{otherwise,} \end{cases}$
    - Selection $x_i = \begin{cases} u_i & \text{if } f(u_i) < f(x_i) \\ x_i & \text{otherwise.} \end{cases}$ Successful update: $S_F = S_F \cup \{F_i\}$ & $S_{CR} = S_{CR} \cup \{CR_i\}$; $S_F$ and $S_{CR}$ are two sets initialized to be empty at the beginning of each generation
- Self-adaptation of control parameters

$$\begin{cases} \mu_{CR} = (1-c)\mu_{CR} + c \cdot \mathrm{mean}_A(S_{CR}) \\ \mu_F = (1-c)\mu_F + c \cdot \mathrm{mean}_L(S_F) \end{cases} \qquad \begin{cases} CR_i = \mathrm{randn}_i(\mu_{CR}, 0.1) \\ F_i = \mathrm{randc}_i(\mu_F, 0.1) \end{cases}$$

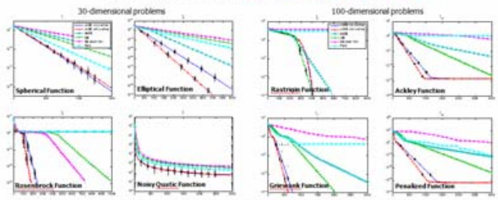
mean$_A$: arithmetic mean; mean$_L$: Lehmer mean $\mathrm{mean}_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}$

- $1/c$: the rate of the self-adaptation. Suggested value $1/c \in [5, 20]$
- $p$: the greediness of the algorithm. Suggested value $p \in [5\%, 20\%]$




## Results for 30D or 100D problems


**Performance Comparison** among JADE with/without archive, SaDE, jDE, DE/rand/1/bin, PSO




(x axis: number of generations; y axis: the value of the best solution obtained so far)

## Results for higher-dimensional problems

**Table 1: Comparison of algorithms for a set of 512-D problems**

| Algorithm | JADE with archive | DE | PSO | [illegible] | [illegible] | [illegible] |
|---|---|---|---|---|---|---|
| Sphere | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Rosenbrock | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Rastrigin | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Griewank | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Ackley | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |

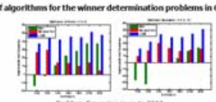
The minimum values obtained

**Table 2: Comparison of algorithms for a set of 1000-D problems**

| Algorithm | JADE with archive | DE | [illegible] | [illegible] |
|---|---|---|---|---|
| Shifted Sphere Function | [illegible] | [illegible] | [illegible] | [illegible] |
| Shifted Rotated Elliptic Function | [illegible] | [illegible] | [illegible] | [illegible] |
| Schwefel Problem 2.6 | [illegible] | [illegible] | [illegible] | [illegible] |
| Shifted Rosenbrock Function | [illegible] | [illegible] | [illegible] | [illegible] |
| Shifted Rotated Ackley Function | [illegible] | [illegible] | [illegible] | [illegible] |
| Shifted Rastrigin Function | [illegible] | [illegible] | [illegible] | [illegible] |
| Shifted Rotated Rastrigin Function | [illegible] | [illegible] | [illegible] | [illegible] |
| Expanded Extended Griewank + Rosenbrock Function | [illegible] | [illegible] | [illegible] | [illegible] |

The minimum values obtained

**Figure: Comparison of algorithms for the winner determination problems in Combinatorial Auction**

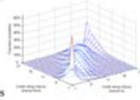


Problem dimension is up to 2200.

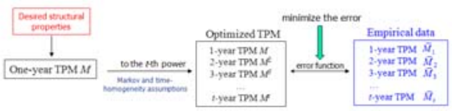
Objective: Find the best combination of bids that can maximize the outcome in an auction

## Application 1 -- TPM optimization

- Transition probability matrix (TPM)
  - TPM indicates the likelihood of obligor's credit rating migrating from one state to another over a given time horizon.
  - It has been used in various high-dollar-value credit decision-making applications ranging from the pricing of financial instruments, loan evaluation, portfolio risk analysis, and economic capital assessment.
- Problem: the empirical TPM does not satisfy the following properties
  - Structural stability: the TPM should show the natural tendency of obligors to maintain status quo and to have a greater probability of migrating to a nearer credit rating than a farther one.
  - Default probability constraints: the probability of default is expected to satisfy monotonic properties: the default probability increases over time for each credit rating, and in each year a higher rating tends to have a lower probability of default than a lower rating.



Surface plot of a TPM

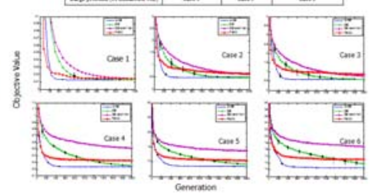



## Problem Complexity & Results

- Problem Complexity
  - Highly nonlinear (due to the non-linear nature of error functions and the matrix exponential operations)
  - High-dimensional (up to 400+ decision variables)
  - Highly constrained (up to 900+ inequality constraints)

| Problem Instances | Square Function | Exp Function | Probit Function |
|---|---|---|---|
| Small problem (of dimension 38) | Case 1 | Case 2 | Case 3 |
| Large problem (of dimension 462) | Case 4 | Case 5 | Case 6 |

Convergence graphs of JADE, jDE, DE/rand/1/bin and PSO



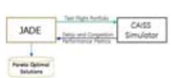

## Application 2 – Flight Planning & Optimization

- Background: Air traffic demand is expected to double or triple in the next 20 years.
  - It will worsen the current U.S. and European National air traffic systems that they are currently operating at the edge of their capabilities.
- The objective of automated flight planning and optimization
  - is to produce a set of air traffic operations over a certain period (e.g. an entire day) that is compatible with the available capacity and that respect multiple stakeholder preferences.
- For example, given a set of flights each of which are assigned multiple alterative paths by its stakeholder,
  - the objective could be to select a route for each flight so as to minimize both the congestion level of the air traffic system and the total flight distance .

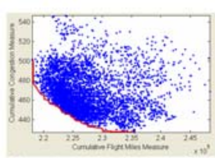

## Results

- Problem Complexity
  - Assume there are $n$ flights in the system and each has two route options.
  - If $n$ = 3, there would be 23 = 8 possible solutions
  - If $n$ = 40, there would be $2^{40}$ (more than **a trillion**) possible solutions
  - In general, $n$ > 30000 and so it is impossible to explore all possible sols.

8 possible solutions of $n$ = 3




JADE is used to search over the solution space (the performance of each solution is evaluated using an air-traffic simulator -- CAISS)




Pareto-optimal and dominated flight path portfolios

## Conclusion

- Evolutionary algorithms are capable of solving nonlinear, non-convex, multi-modal problems which usually cause great difficulty to traditional nonlinear programming techniques.
- Parameter adaptive differential evolution (JADE) is easy to use, reliable and efficient to achieve satisfactory solutions. Compared to many other EAs, JADE
  - improves the convergence performance of the optimization , and
  - avoids users' knowledge of the problem's characteristics and the theory of optimization by adapting control parameter automatically according to the feedback from the evolutionary search.
- JADE has been tested on a variety of multi-modal optimization problems with up to thousands of decision variables.
- Its application to real-world problems has contributed better results than other state-of-the-art algorithms.

Rensselaer

CATS Center for Automation Technologies and Systems

NYSTAR