Adaptive Strategy Selection in Differential Evolution

Wenyin Gong[1] Álvaro Fialho[2] Zhihua Cai[1]

[1]School of Computer Science, China University of Geosciences
Wuhan, China

[2]Microsoft Research – INRIA Joint Centre
Orsay, France

July 10, 2010

Outline

Differential Evolution

Differential Evolution

DE (Storn & Price, 1997) is a simple yet powerful population-based, direct search, self-adaptive algorithm for global optimization mainly using real-valued parameters.

Advantages

Among DE's advantages are its simple structure, ease of use, speed, and robustness.

Differential Evolution

Differential Evolution

DE (Storn & Price, 1997) is a simple yet powerful population-based, direct search, self-adaptive algorithm for global optimization mainly using real-valued parameters.

Advantages

Among DE's advantages are its simple structure, ease of use, speed, and robustness.

Applications

Currently, DE has many applications, such as data mining, engineering design, pattern recognition, power systems, etc.

Differential Evolution

Differential Evolution

DE (Storn & Price, 1997) is a simple yet powerful population-based, direct search, self-adaptive algorithm for global optimization mainly using real-valued parameters.

Advantages

Among DE's advantages are its simple structure, ease of use, speed, and robustness.

Applications

Currently, DE has many applications, such as data mining, engineering design, pattern recognition, power systems, etc.

Mutation Strategies of DE

In DE there are many mutation strategies, and different strategy has different characteristics.

- "DE/rand/1/bin": classic DE strategy, less greedy, slower convergence speed, and more reliable, more suitable for multi-modal problems.
- The best-so-far solution based strategies: converge faster, more suitable for unimodal functions.

Mutation Strategies of DE

In DE there are many mutation strategies, and different strategy has different characteristics.

- "DE/rand/1/bin": classic DE strategy, less greedy, slower convergence speed, and more reliable, more suitable for multi-modal problems.
- The best-so-far solution based strategies: converge faster, more suitable for unimodal functions.
- The strategies based on the two difference vectors: provide better perturbation.

Mutation Strategies of DE

In DE there are many mutation strategies, and different strategy has different characteristics.

- "DE/rand/1/bin": classic DE strategy, less greedy, slower convergence speed, and more reliable, more suitable for multi-modal problems.
- The best-so-far solution based strategies: converge faster, more suitable for unimodal functions.
- The strategies based on the two difference vectors: provide better perturbation.
- The current solution (base vector) based strategies: perform local search, converge faster, stagnate quickly.

Mutation Strategies of DE

In DE there are many mutation strategies, and different strategy has different characteristics.

- "DE/rand/1/bin": classic DE strategy, less greedy, slower convergence speed, and more reliable, more suitable for multi-modal problems.
- The best-so-far solution based strategies: converge faster, more suitable for unimodal functions.
- The strategies based on the two difference vectors: provide better perturbation.
- The current solution (base vector) based strategies: perform local search, converge faster, stagnate quickly.

Strategy Adaptation of DE

One drawback: strategy problem-dependent

Choosing the best among different mutation strategies available for DE is not easy for a specific problem (Qin, *et al.*, 2005, 2009).

- **SWAF** (Xie & Zhang, 2004): neural network based strategy weights updating method;
- **DEwSAcc** (Zamuda, *et al.*, 2008): fixed strategy weights;
- **SaDE** (Qin, *et al.*, 2009): adaptively updates the weights in the search based on their previous success rate.

Strategy Adaptation of DE

One drawback: strategy problem-dependent

Choosing the best among different mutation strategies available for DE is not easy for a specific problem (Qin, *et al.*, 2005, 2009).

- **SWAF** (Xie & Zhang, 2004): neural network based strategy weights updating method;
- **DEwSAcc** (Zamuda, *et al.*, 2008): fixed strategy weights;
- **SaDE** (Qin, *et al.*, 2009): adaptively updates the weights in the search based on their previous success rate.

Motivations

Motivation 1

- In DE, there are many mutation strategies; different strategy has different characteristics and is suitable for different problems.
- However, the selection of which of such strategies should be used is a difficult and crucial task for the performance of the DE, besides being problem-dependent (Qin, *et al.*, 2009).

Motivations

Motivation 1

- In DE, there are many mutation strategies; different strategy has different characteristics and is suitable for different problems.
- However, the selection of which of such strategies should be used is a difficult and crucial task for the performance of the DE, besides being problem-dependent (Qin, *et al.*, 2009).
- There are only a few works to adopt multi-strategies in the DE literature.

Motivations

Motivation 1

- In DE, there are many mutation strategies; different strategy has different characteristics and is suitable for different problems.
- However, the selection of which of such strategies should be used is a difficult and crucial task for the performance of the DE, besides being problem-dependent (Qin, *et al.*, 2009).
- There are only a few works to adopt multi-strategies in the DE literature.

Motivations

Motivation 2

- Adaptive strategy selection (AdapSS) is able to choose the suitable strategy for different problems adaptively.
- Thus, DE with adaptive strategy selection might enhance its performance and remedy the difficulties to choose different strategies for different problems.

Motivations

Motivation 2

- Adaptive strategy selection (AdapSS) is able to choose the suitable strategy for different problems adaptively.
- Thus, DE with adaptive strategy selection might enhance its performance and remedy the difficulties to choose different strategies for different problems.

Main Purposes

The main objectives of this work are two-fold:

- The probability matching (PM) technique is integrated into DE to implement the adaptive strategy selection.
- Four credit assignment techniques based on the relative fitness improvement are compared.

Main Purposes

The main objectives of this work are two-fold:

- The probability matching (PM) technique is integrated into DE to implement the adaptive strategy selection.
- Four credit assignment techniques based on the relative fitness improvement are compared.

Probability Matching Technique (Goldberg, 1990)

Update the quality of a strategy a:

$$Q_a(t+1) = Q_a(t) + \alpha[R_a(t) - Q_a(t)] \tag{1}$$

Update the probability of a strategy a:

$$P_a(t+1) = P_{min} + (1 - K \times P_{min}) \frac{Q_a(t)}{\sum_{i=1}^{K} Q_i(t)} \tag{2}$$

where $R_a(t)$ is the reward of a strategy a at generation t; $Q_a(t)$ is the known quality; $\alpha \in (0, 1]$ is a user-defined adaptation rate; $P_{min} \in (0, 1)$ is a user-defined minimal value of probability, used to ensure that no strategy gets lost (Thierens, 2005).

Relative Fitness Improvement based CA

The relative fitness improvement based CA is adopted in this work (Ong, *et al.*, 2004). For the minimization problem, the relative fitness improvement η_i of the i-th solution is defined as follows:

$$\eta_i = \frac{\delta}{cf_i} \cdot |pf_i - cf_i| \tag{3}$$

where $i = 1, \cdots, NP$. δ is the fitness of the best-so-far solution in the population. pf_i and cf_i are the fitness of the target parent and of its offspring, respectively.

Remark

Note that if no improvement (*i.e.*, the offspring is worse than or equal to its target parent) is achieved, a null credit is assigned (*i.e.*, $\eta_i = 0$).

CA: Average Absolute Reward

Following (Fialho, *et al.*, 2009), four CA methods are implemented to extract the reward for each strategy.
The first CA technique is the "average absolute reward", which is defined as:

- **AverageAbsoluteReward** (AvgAbs):

$$R_a(t) = \frac{\sum_{i=1}^{|S_a|} S_a(i)}{|S_a|} \tag{4}$$

where S_a is the set of all relative fitness improvements achieved by the application of a strategy $a (a = 1, \cdots, K)$ during generation t. $|S_a|$ is the number of elements in S_a. If $|S_a| = 0$, $R_a(t) = 0$.

CA: Average Normalized Reward

The second CA method is the "average normalized reward":

- **AverageNormalizedReward** (AvgNorm):

$$R'_a(t) = \frac{\sum_{i=1}^{|S_a|} S_a(i)}{|S_a|}$$

and

$$R_a(t) = \frac{R'_a(t)}{\max\limits_{b=1,\cdots,K} R'_b(t)} \tag{5}$$

CA: Extreme Absolute Reward

The third CA method is the "extreme absolute reward":

- **ExtremeAbsoluteReward** (ExtAbs):

$$R_a(t) = \max_{i=1,\cdots,|S_a|} S_a(i) \tag{6}$$

CA: Extreme Normalized Reward

The fourth CA method is the "extreme normalized reward":

- **ExtremeNormalizedReward** (ExtNorm):

$$R'_a(t) = \max_{i=1,\cdots,|S_a|} S_a(i)$$

and

$$R_a(t) = \frac{R'_a(t)}{\max_{b=1,\cdots,K} R'_b(t)} \tag{7}$$

How to Choose Strategies

In this work, the following four strategies are selected to form the strategy pool. These strategies are also used in SaDE (Qin, *et al.*, 2009).

1) "DE/rand/1":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) \tag{8}$$

2) "DE/rand/2":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) + F \cdot \left(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\right) \tag{9}$$

3) "DE/rand-to-best/2":

$$\mathbf{v}_i = \mathbf{x}_{r_1} + F \cdot \left(\mathbf{x}_{best} - \mathbf{x}_{r_1}\right) + F \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) + F \cdot \left(\mathbf{x}_{r_4} - \mathbf{x}_{r_5}\right) \tag{10}$$

4) "DE/current-to-rand/1":

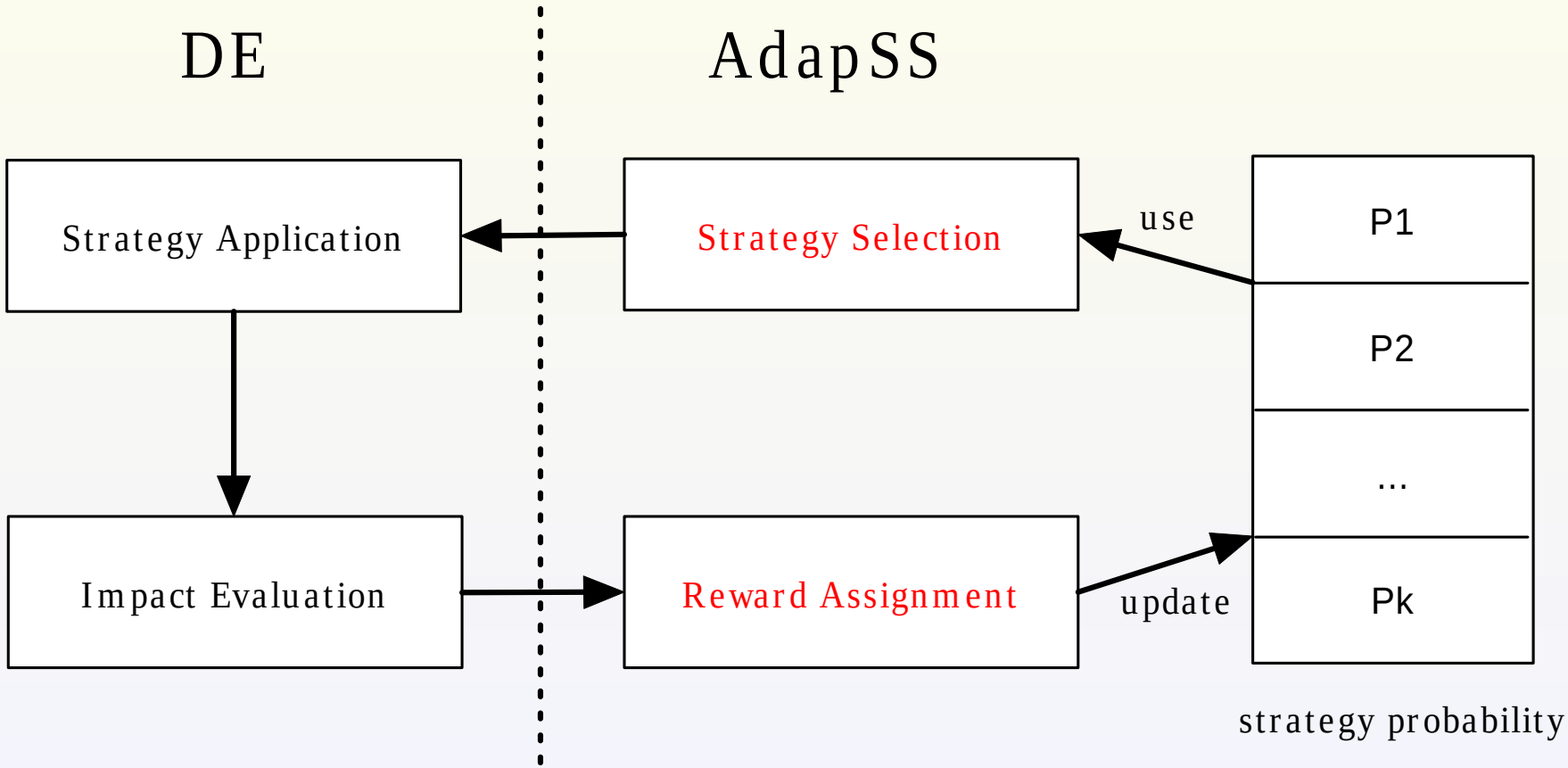
$$\mathbf{v}_i = \mathbf{x}_i + F \cdot \left(\mathbf{x}_{r_1} - \mathbf{x}_i\right) + F \cdot \left(\mathbf{x}_{r_2} - \mathbf{x}_{r_3}\right) \tag{11}$$

How to Choose Strategies

Remark

- All of them are controlled by the DE crossover rate CR.
- Note that other strategies could also be introduced in the pool; these four strategies can be seen as instances used as test-bed for the evaluation of the proposed method.

Framework of AdapSS-DE



Difference between PM-AdapSS-DE and SaDE

Main Difference

Note that in SaDE (Qin, *et al.*, 2009), the strategy adaptation is also implemented. However, our approach is completely different from SaDE in the credit assignment. In SaDE, the success and failure number of runs are considered. While the relative fitness improvement is used in our approach.

Benchmark Functions

In order to verify the performance of our approach, 13 scalable benchmark functions are chosen from the literature (Yao, *et al.*, 1999) as the test suit.

- Unimodal functions: $f_{01} - f_{04}$, f_{06}, and f_{07};
- Multimodal functions: f_{05}, $f_{08} - f_{13}$.

The above-mentioned functions have many variety characteristics, such as multimodal, non-separable, noise, etc.

Parameter Settings

In this study, we only use reasonable parameter settings that have been used in the DE literature. We don't make any effort to find the best parameter settings.

- Dimension of each function: $D = 30$;
- Population size: $NP = 100$;
- $CR = 0.9$ and $F = 0.5$;
- $K = 4$, $P_{min} = 0.05$, and $\alpha = 0.3$;
- Value to reach: For functions $f_{01} - f_{06}$ and $f_{08} - f_{13}$, VTR $= 10^{-8}$; for functions f_{07}, VTR $= 10^{-2}$;
- Max_NFFEs: For $f_{01}, f_{06}, f_{10}, f_{12}$, and f_{13}, Max_NFFEs $= 150,000$; for $f_{03} - f_{05}$, Max_NFFEs $= 500,000$; for f_{02} and f_{11}, Max_NFFEs $= 200,000$; for $f_{07} - f_{09}$, Max_NFFEs $= 300,000$.

Performance Criteria

Four performance criteria are selected from the literature (Suganthan, *et al.*, 2005).

- **Error**: The error of a solution $\mathbf{x}$ is defined as $f(\mathbf{x}) - f(\mathbf{x}^*)$.
- **NFFEs**: The NFFEs is also recorded when the VTR is reached.
- **Successful rate (S_r)**: S_r is calculated as the number of successful runs divided by the total number of runs.
- **Convergence graphs**: The convergence graphs show the median error performance of the best solution over the total runs.

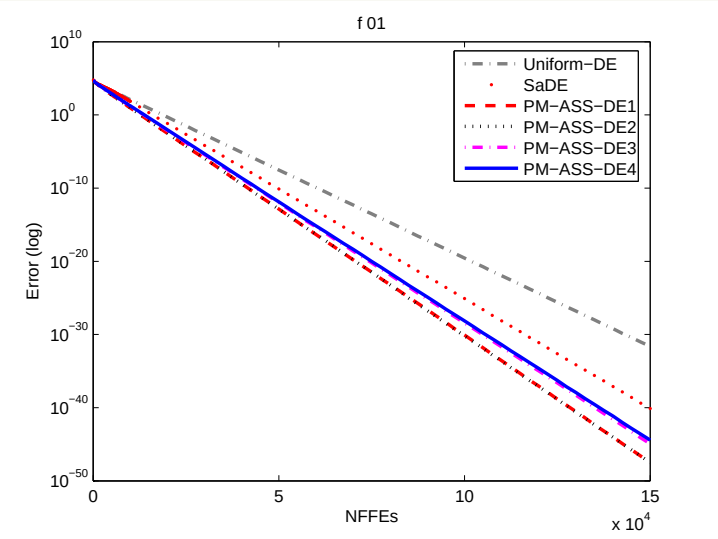
Comparison on Different Credit Assignment Methods

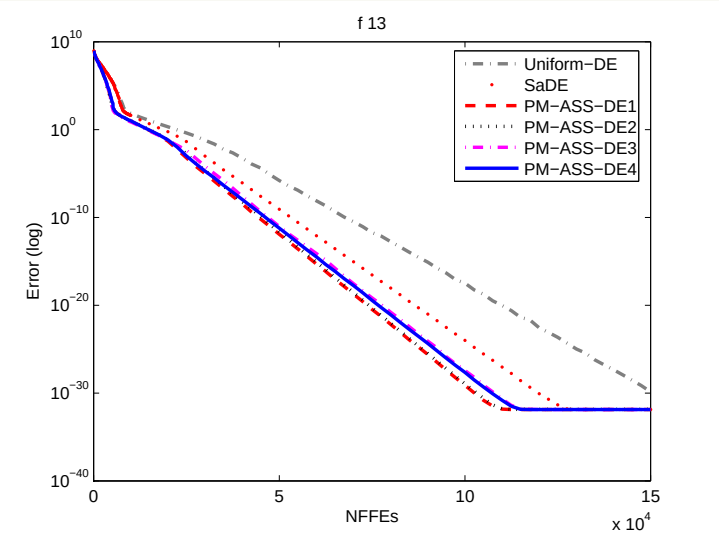
Six DE variants are considered in this experiment.

- **Uniform-DE (U-DE)**: DE with the uniform strategy selection;
- **SaDE**: SaDE method (Qin, *et al.*, 2009) without CR and F adaptation;
- **PM-AdapSS-DE-1 (my-DE-1)**: PM-AdapSS-DE with *AvgAbs* CA method;
- **PM-AdapSS-DE-2 (my-DE-2)**: PM-AdapSS-DE with *AvgNorm* CA method;
- **PM-AdapSS-DE-3 (my-DE-3)**: PM-AdapSS-DE with *ExtAbs* CA method;
- **PM-AdapSS-DE-4 (my-DE-4)**: PM-AdapSS-DE with *ExtNorm* CA method.

Analysis of Strategy Adaptation: General Performance

	U-DE	SaDE	my-DE-1	my-DE-2	my-DE-3	my-DE-4
Error Ranking	57	48	35	29	26	29
Error Dominated	8	7	2	3	2	2
NFFEs Ranking	62	56	22	25	29	37





Analysis of Strategy Adaptation: General Performance

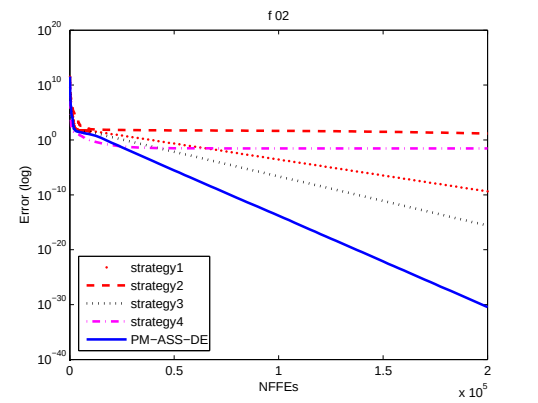
- PM-AdapSS-DE obtains better results than Uniform-DE and SaDE in terms of the convergence rate and the error values;
- The averaged reward based PM-AdapSS-DE (PM-AdapSS-DE-1 and PM-AdapSS-DE-2) is better than the extreme reward based PM-AdapSS-DE methods in most of the functions.
- Hence, PM-AdapSS-DE-1 will be further discussed in the following experiments.

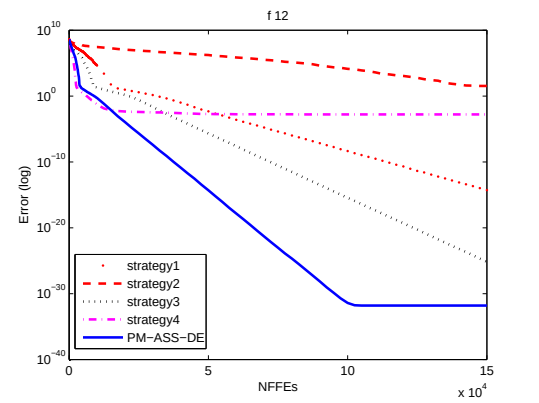
Analysis of Strategy Adaptation: Adaptation

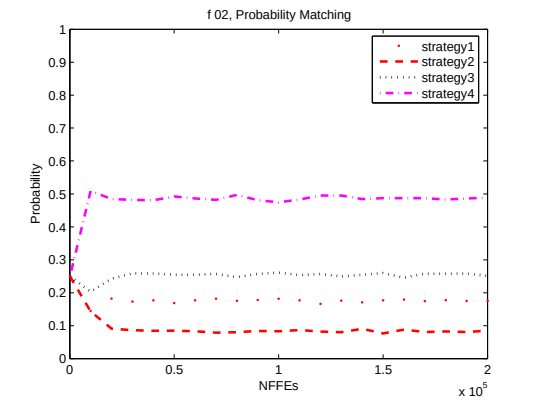
PM-AdapSS-DE is compared with DE using the single strategy in the pool.

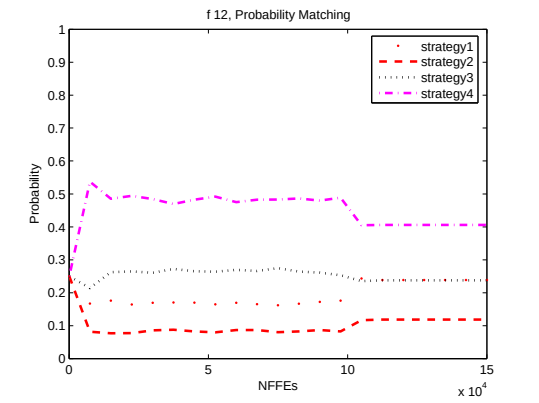
- Strategy1: DE with "DE/rand/1/bin" strategy;
- Strategy2: DE with "DE/rand/2/bin" strategy;
- Strategy3: DE with "DE/rand-to-best/s/bin" strategy;
- Strategy4: DE with "DE/current-to-rand/1/bin" strategy;

Analysis of Strategy Adaptation: Adaptation









Analysis of Strategy Adaptation: Adaptation

The Wilcoxon ranked test with $\alpha = 0.05$ is used to compare the error values.

	S-1	S-2	S-3	S-4	PM-AdapSS-DE
$w/t/l$	$9/2/2$	$12/1/0$	$10/2/1$	$11/1/1$	$--$
$\sum S_r$	9.06	0.00	10.82	1.14	10.82

- PM-AdapSS-DE is significantly better than DE with each single strategy in the pool on most of the functions.
- On the majority of the test functions PM-AdapSS-DE is capable of producing the fastest convergence speed compared with the four DE algorithms.
- In general, our approach is able to efficiently select the suitable strategy for different problems.

Analysis of Strategy Adaptation: Adaptation

The Wilcoxon ranked test with $\alpha = 0.05$ is used to compare the error values.

	S-1	S-2	S-3	S-4	PM-AdapSS-DE
$w/t/l$	$9/2/2$	$12/1/0$	$10/2/1$	$11/1/1$	$--$
$\sum S_r$	9.06	0.00	10.82	1.14	10.82

- PM-AdapSS-DE is significantly better than DE with each single strategy in the pool on most of the functions.
- On the majority of the test functions PM-AdapSS-DE is capable of producing the fastest convergence speed compared with the four DE algorithms.
- In general, our approach is able to efficiently select the suitable strategy for different problems.

Conclusions

- probability matching + relative fitness improvement based CA + DE
- PM-AdapSS-DE implements the adaptive strategy selection. It is capable of choosing the suitable strategy adaptively for a problem at hand.

Future Work

1. How to assign the reward, e.g., average/extreme fitness improvement, fitness + diversity, and so on?
2. Other strategy selection techniques?
3. Combine AOS with other DE model;
4. The influence of the parameter adaptation of DE in AOS-DE;
5. For the constrained optimization problems;
6. ...