Multi Armed Bandits for Many Task Optimization

Le Tien Thanh - La Van Cuong - Ta Bao Thang
Supervisor: Professor Huynh Thi Thanh Binh

March 26, 2021

Agenda

1. Introduction
2. Related works
3. Proposed method - Many-task Multi-armed Bandit Evolutionary Algorithm (Ma^{2}BEA)
4. Experimental results

Agenda

Many task optimization - MaTSOO benchmark

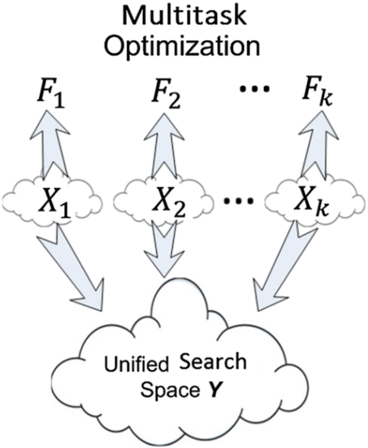
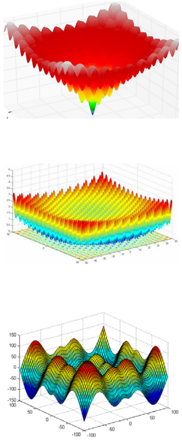


Figure 1: Many task optimization when $k >> 2$ (.i.e 50 tasks) $\rightarrow$ propose Ma^{2}BEA

Single Objective Optimization (SOO)



Agenda

MFEA

MFEA

A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015

Problem

- Do not group ideal assist tasks

MFEA-II

K. K. Bali, Y.-S. Ong, A. Gupta, *et al.*, "Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 69–83, 2019

Problem

- Solving the auxiliary optimization problem of adjusting the knowledge transfer coefficient based on the similarity of populations of the given tasks
- Number of problem increase quadratically when number of tasks are increased

MaTGA

MaTGA

Y. Chen, J. Zhong, L. Feng, *et al.*, "An adaptive archive-based evolutionary framework for many-task optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019

Problem

- Store past solutions
- Measure KLD between past solutions
- Complex, many hyper-parameters, long computation times

EBSGA

EBSGA

R.-T. Liaw and C.-K. Ting, "Evolutionary manytasking optimization based on symbiosis in biocoenosis," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 4295–4303

Problem

- Knowledge exchange by individuals swapping.
- Keeps track of the frequency of success and fail individual swapping todetermine the next pair of tasks to swap the individuals.
- The adaptive transfer strategy is handcrafted.

Agenda

Task selection as a Multi Armed Bandit (MAB) problem

Given

- A K-task MTO problem

Definition (Action)

For a task k, there are $K-1$ actions of choosing k' such that $k' \in \{1, \ldots, K\}$ and $k' \neq k$.

Definition (Reward)

After task k' is selected to be combined with task k, the reward of choosing that action is defined as

$$reward = \begin{cases} 1 \text{ if } f_k(c) < f_k(p), \exists p \in P_k \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

where c is the offspring generated by the reproduction procedure and $f_k(.)$ is the fitness function of the k^{th} task.

UCB function to solve MAB

Property of reward function

- Reward takes two value 0 or 1 $\rightarrow$ reward distribution is generated from an unknown Bernoulli distribution.
- From[a], use KL-UCB to solve.

[a] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

KL-UCB function

$$k' = \underset{j}{\text{argmin}}\ \mu(j) + \frac{1 + t \times log^2(t)}{N(j)} \tag{2}$$

where

- $\mu(j)$ is the mean of reward when task j is chosen
- $N(j)$ is the number of times task j is chosen
- t is the total number of actions chosen.

Ma^{2}BEA, optimize 1 task, 1 generation

Algorithm 1 Ma^{2}BEA on each generation of k^{th} task

Initialize $P_k^{(c)} = \emptyset$;
while number of offspring $< N$ **do**
    Randomly select p_a from P_k;
    if $rand(0,1) < rmp$ **then**
        Choose k' using Formula (2);
        Randomly select p_b from $P_{k'}$;
        $c =$ *Inter-task crossover* between p_a and p_b;
    else
        Sample p_b from P_k;
        $c =$ *Intra-task crossover* between p_a and p_b;
    end if
    $c = mutate(c)$;
    Evaluate offspring c;
    Update estimation $\mu(k')$ and $N(k')$ for Kullback–Leibler UCB (KL-UCB) if c generated by inter-task crossover;
    $P_k^{(c)} = P_c \cup \{c\}$;
end while
$P_k \leftarrow$ Select N best individuals from $P_k \cup P_k^{(c)}$ to form the next-population of task k;

Ma^{2}BEA, proposed structure for manytasking

Algorithm 2 Pseudo code of Ma^{2}BEA

for $k \in \{1, \dots, K\}$ **do**
    Randomly sample N individuals to form subpopulation P_k;
    Evaluate individual in P_k for task k only;
end for
while stopping conditions are not satisfied **do**
    for $k \in random\_permutation(1, \dots, K)$ **do**
        Invoke Algorithm 1 for task k;
    end for
end while

Agenda

Experimental setup

Benchmark

- 10-task benchmarks, easy to debug, known relationship between tasks
- 50-task benchmarks, MaTSOO benchmark from the WCCI2020 Competition

MFEA/EBSGA/Ma^{2}BEA

- Population size: 100
- rmp: 0.3
- sbxdi: 2
- pmdi: 5

MaTGA

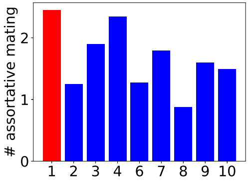
Use authors' default parameters

Result on 10-task benchmark

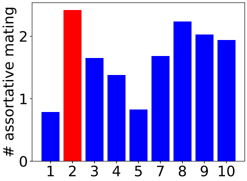
Task	$\mathrm{Ma^2BEA}$	MFEA	MaTGA	EBSGA
T_1	**3.72E-05**	1.31E+00 ($-$)	2.45E-04 ($-$)	5.58E-04 ($-$)
T_2	**6.48E-06**	1.27E+00 ($-$)	4.75E-04 ($-$)	6.10E-05 ($-$)
T_3	**4.35E-07**	1.17E+00 ($-$)	**0.00E+00** ($\approx$)	2.00E-06 ($-$)
T_4	**7.09E-13**	3.37E+00 ($-$)	1.00E-06 ($-$)	1.60E-05 ($-$)
T_5	1.74E+01	8.15E+02 ($-$)	**2.16E-02** ($+$)	3.29E-02 ($+$)
T_6	**7.03E-04**	1.99E+01 ($-$)	3.61E-03 ($-$)	**8.59E-04** ($\approx$)
T_7	1.00E-02	1.05E+01 ($-$)	**5.57E-04** ($+$)	1.60E-03 ($+$)
T_8	1.89E+02	2.34E+03 ($-$)	**3.40E-02** ($-$)	4.73E-02 ($-$)
T_9	**4.00E-07**	4.11E+02 ($-$)	4.52E-03 ($-$)	4.58E-03 ($-$)
T_{10}	2.74E+01	**1.54E+01** ($+$)	1.57E+01 ($+$)	3.41E+01 ($-$)

Table 1: The performance of the GA-based evolution many-tasking algorithms of 10-task benchmark after 30 independent runs. ($-, +,$ *and* $\approx$ denote the corresponding algorithm is significantly worse, better or similar to $\mathrm{Ma^2BEA}$ after Wilcoxon signed-rank test at $\alpha = 0.05$)

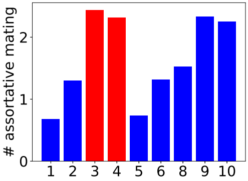
Analyse the KL-UCB



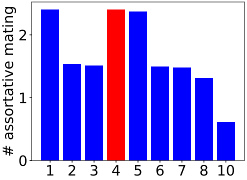
(a) Task 5



(b) Task 6



(c) Task 7



(d) Task 9

Figure 2: The average times of choosing knowledge transfer target for complex tasks by Ma^{2}BEA with the ideal assisted task is highlighted in red

Result on 50-task benchmark

Benchmark	Ma^2BEA	MFEA	MaTGA	EBSGA
B_1	**50 (50)**	0 (0)	0 (0)	0 (0)
B_2	**50 (50)**	0 (0)	0 (0)	0 (0)
B_3	8 (0)	0 (0)	**42 (8)**	0 (0)
B_4	17 (9)	0 (0)	0 (0)	**33 (33)**
B_5	**50 (50)**	0 (0)	0 (0)	0 (0)
B_6	**50 (50)**	0 (0)	0 (0)	0 (0)
B_7	3 (0)	0 (0)	14 (2)	**33 (28)**
B_8	**30 (28)**	0 (0)	0 (0)	20 (20)
B_9	**42 (42)**	0 (0)	0 (0)	8 (8)
B_{10}	8 (6)	2 (0)	11 (10)	**29 (23)**

Table 2: The performance of the GA-based evolution many-tasking algorithms of 10 benchmarks, each contains 50 tasks, after 30 independent runs. Each entry denotes the number of tasks which the corresponding algorithm is better than other algorithms. The figure within the curly brace indicates the number of tasks which the corresponding algorithm is the best, and it is significantly better than the 2^{nd} place algorithm, tested by Wilcoxon signed-rank test at $\alpha = 0.05$)

Execution time analysis

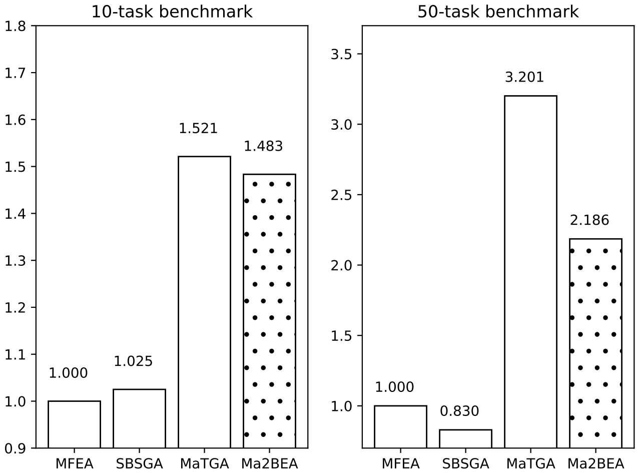


Figure 3: Execution of different algorithms on two many-tasking benchmarks divided by execution time of MFEA on the same runtime environment

Conclusion

Contribution

- KL-UCB as task selection
- New algorithm structure for MaTO
- Submitted a paper in CEC 2021

Future tasks

- Analyze experimental result on Mujoco
- Translate to Thesis

Thank you for your attention!