Adaptive Threshold Parameter Estimation with Recursive Differential Grouping for Problem Decomposition: Supplementary Material

Yuan Sun, Mohammad Nabi Omidvar, Michael Kirley, and Xiaodong Li

1 Grouping Procedure of Recursive Differential Grouping

The RDG method begins by identifying the interaction between the first decision variable x_1 and the remaining decision variables. If no interaction is detected, x_1 will be placed in the separable decision variable group, and the algorithm will move on to the next decision variable x_2. If any interaction is detected, the remaining decision variables will be divided into two (nearly) equally-sized groups G_1 and G_2. Then the interaction between x_1 and G_1, x_1 and G_2 will be identified respectively. This process is recursively conducted until all the individual decision variables that interact with x_1 are identified and placed in the decision variable subset X_1 with x_1.

Then, the RDG method examines the interaction between X_1 and the remaining decision variables (excluding the decision variables in X_1) to identify the individual decision variables that conditionally interact with x_1 (linked by other decision variables). If any interaction is identified, the interacting decision variables will be placed into X_1. This process is repeated until no interaction can be further detected between X_1 and the remaining decision variables (exclusive X_1). The decision variables in X_1 will be placed in a non-separable group.

The RDG method moves on to the next decision variable that has not been grouped (x_i). The interaction between x_i and the remaining decision variables will be examined, and both the interacting and conditionally interacting (linked) decision variables will be placed into one group with x_i. This process is repeated until all of the decision variables are grouped. It returns the separable (*seps*) and the non-separable (*nonseps*) decision variable groups as the outputs.

2 Decomposition Comparison on the CEC'2010 Benchmark Problems

The detailed results of the RDG2, RDG (with $\alpha = 10^{-12}$), and DG2 methods when used to decompose the CEC'2010 and benchmark problems were presented in Table 1. All of the RDG2, RDG and DG2 methods achieved 100% decomposition accuracy on the CEC'2010 partially separable benchmark problems. The number of function evaluations (FEs) used by RDG2 was close to that used RDG on all of the benchmark problems except for f_6. The RDG2 method used much less FEs than RDG when used to decompose f_6. The reason for this is that the threshold value estimated by the RDG method ($\epsilon = 2.13 \times 10^{-5}$) was too small, resulting in identifying some separable decision variables as non-separable. The DG2 method used a fixed number of FEs to decompose an n-dimensional problem: $(n^2 + n + 2)/2$.

3 Optimization Comparison on the CEC'2010 Benchmark Problems

The detailed results of the RDG2, RDG, and DG2 methods when embedded into the CC framework (with CMA-ES as the component optimizer) to solve the CEC'2010 benchmark problems were presented in Table 2. The CC-RDG2 algorithm generated statistically equally well solution quality with CC-RDG, as the decomposition results generated by the RDG2 and DG2 methods were similar across the CEC'2010 benchmark problems (as shown in Table 1). The CC-RDG2 algorithm generated statistically significantly better/equally well solution quality than the CC-DG2 algorithm on 9/11 out of 20 benchmark problems. The main reason for this was that RDG2 used much less FEs than DG2 in the decomposition stage, saving more FEs for the optimization stage.

Table 1: The experimental results of the RDG2, RDG (with $\alpha = 10^{-12}$) and DG2 methods when used to decompose the CEC'2010 benchmark problems. "a" is the decomposition accuracy; "FEs" is the function evaluations used.

Func	RDG2		RDG ($\alpha = 10^{-12}$)		DG2	
Num	a	FEs	a	FEs	a	FEs
f_1	–	2.99e+03	–	3.00e+03	–	5.00e+05
f_2	–	3.01e+03	–	3.00e+03	–	5.00e+05
f_3	–	5.99e+03	–	6.00e+03	–	5.00e+05
f_4	100%	4.19e+03	100%	4.20e+03	100%	5.00e+05
f_5	100%	4.14e+03	100%	4.15e+03	100%	5.00e+05
f_6	100%	7.19e+03	100%	5.00e+04	100%	5.00e+05
f_7	100%	4.22e+03	100%	4.23e+03	100%	5.00e+05
f_8	100%	5.59e+03	100%	5.60e+03	100%	5.00e+05
f_9	100%	1.40e+04	100%	1.40e+04	100%	5.00e+05
f_{10}	100%	1.40e+04	100%	1.40e+04	100%	5.00e+05
f_{11}	100%	1.36e+04	100%	1.36e+04	100%	5.00e+05
f_{12}	100%	1.43e+04	100%	1.43e+04	100%	5.00e+05
f_{13}	100%	2.92e+04	100%	2.92e+04	100%	5.00e+05
f_{14}	100%	2.05e+04	100%	2.05e+04	100%	5.00e+05
f_{15}	100%	2.05e+04	100%	2.05e+04	100%	5.00e+05
f_{16}	100%	2.09e+04	100%	2.09e+04	100%	5.00e+05
f_{17}	100%	2.07e+04	100%	2.07e+04	100%	5.00e+05
f_{18}	100%	4.98e+04	100%	4.98e+04	100%	5.00e+05
f_{19}	100%	5.99e+03	100%	6.00e+03	100%	5.00e+05
f_{20}	100%	5.08e+04	100%	5.08e+04	100%	5.00e+05

Table 2: The optimization results of the RDG2, RDG and DG2 when embedded into the CC framework to solve the CEC'2010 benchmark problems. The entries with the best solution quality are highlighted in bold according to the Wilcoxon rank-sum tests (significance level α=0.05) with Holm p-value correction.

Func	Stats	RDG2	RDG	DG2
f_1	median	**2.72e+05**	2.86e+05	5.18e+05
	mean	2.76e+05	2.84e+05	5.22e+05
	std	3.00e+04	2.28e+04	4.31e+04
f_2	median	4.43e+03	4.43e+03	4.52e+03
	mean	4.42e+03	4.42e+03	4.51e+03
	std	1.76e+02	1.76e+02	1.72e+02
f_3	median	1.14e+00	1.12e+00	1.13e+00
	mean	1.09e+00	1.05e+00	1.04e+00
	std	2.62e-01	3.49e-01	4.05e-01
f_4	median	**1.01e+06**	**9.97e+05**	1.61e+06
	mean	1.01e+06	1.01e+06	1.60e+06
	std	8.20e+04	9.37e+04	1.32e+05
f_5	median	9.75e+07	9.05e+07	9.45e+07
	mean	9.82e+07	9.52e+07	9.12e+07
	std	2.11e+07	2.22e+07	2.07e+07
f_6	median	**1.10e+00**	**1.04e+00**	1.57e+00
	mean	9.95e-01	9.17e-01	1.56e+00
	std	3.89e-01	4.22e-01	9.62e-02
f_7	median	7.31e-19	7.40e-19	7.58e-19
	mean	7.37e-19	7.40e-19	7.58e-19
	std	8.93e-20	8.35e-20	9.22e-20
f_8	median	2.05e-17	2.15e-17	2.11e-17
	mean	3.18e+05	7.97e+05	4.78e+05
	std	1.10e+06	1.62e+06	1.32e+06
f_9	median	**4.82e+06**	**4.74e+06**	6.62e+06
	mean	4.88e+06	4.82e+06	6.62e+06
	std	6.26e+05	5.25e+05	4.32e+05
f_{10}	median	2.89e+03	2.89e+03	2.84e+03
	mean	2.87e+03	2.87e+03	2.84e+03
	std	1.29e+02	1.29e+02	1.37e+02
f_{11}	median	1.52e-12	1.51e-12	1.52e-12
	mean	1.67e-01	3.58e-02	3.51e-02
	std	3.98e-01	1.79e-01	1.75e-01
f_{12}	median	4.28e-22	4.30e-22	4.37e-22
	mean	4.30e-22	4.22e-22	4.26e-22
	std	2.64e-23	8.38e-23	8.96e-23
f_{13}	median	3.98e+00	3.98e+00	3.98e+00
	mean	3.50e+00	5.90e+00	5.90e+00
	std	4.35e+00	4.00e+00	4.32e+00
f_{14}	median	**3.90e-20**	**3.90e-20**	1.89e-19
	mean	3.91e-20	3.91e-20	1.98e-19
	std	2.11e-21	2.11e-21	3.35e-20
f_{15}	median	1.91e+03	1.92e+03	1.91e+03
	mean	1.90e+03	1.94e+03	1.91e+03
	std	9.01e+01	1.10e+02	6.81e+01
f_{16}	median	**8.41e-13**	**8.41e-13**	8.70e-13
	mean	8.45e-13	8.43e-13	8.73e-13
	std	2.06e-14	2.10e-14	2.33e-14
f_{17}	median	**6.89e-24**	**6.89e-24**	7.32e-24
	mean	6.90e-24	6.90e-24	7.36e-24
	std	1.74e-25	2.05e-25	2.48e-25
f_{18}	median	**1.20e+01**	**1.55e+01**	3.21e+01
	mean	1.36e+01	1.50e+01	4.33e+01
	std	7.81e+00	7.19e+00	3.01e+01
f_{19}	median	**5.27e+03**	**5.63e+03**	1.99e+04
	mean	5.27e+03	5.46e+03	2.00e+04
	std	7.12e+02	7.07e+02	2.38e+03
f_{20}	median	8.55e+02	8.55e+02	8.29e+02
	mean	8.26e+02	8.26e+02	8.36e+02
	std	6.35e+01	6.35e+01	5.06e+01