Robust Adaptive Particle Swarm Optimization

Genki Ueno
Department of Electrical Engineering
Tokyo Metropolitan University
Tokyo, Japan
ueno-genki@c.metro-u.ac.jp

Keiichiro Yasuda
Department of Electrical Engineering
Tokyo Metropolitan University
Tokyo, Japan
yasuda@eei.metro-u.ac.jp

Nobuhiro Iwasaki
Department of Electrical Engineering
Tokyo Metropolitan University
Tokyo, Japan

Abstract - *It is well known that Particle Swarm Optimization (PSO), which was originally proposed by J. Kennedy et al., is a powerful algorithm for solving unconstrained and constrained global optimization problems. Appropriate adjustment of its parameters, however, requires a lot of time and labor when PSO is applied to real optimization problems. In this paper, we point out that giving diversity to the parameter of each particle enables PSO to solve various problems. And we propose the algorithm that has diversity of particles and an adaptive strategy for tuning the parameters. Some numerical simulations were carried out in order to examine the search ability of the proposed approach.*

Keywords: Optimization, Meta-Heuristics, Particle Swarm Optimization.

1 Introduction

Following two steps are necessary to optimize an actual engineering system by use of optimization techniques. The first step is modeling an actual engineering system and describing as optimization problem. The second step is solving a modeled optimization problem by use of optimization techniques. It is desirable that optimization techniques can be applied to many kinds of optimization problems and obtain as a good solution as possible within a restricted calculation time.

Many classic optimization techniques can be applied to restricted kinds of optimization problems, and these techniques realize effective search by use of the restriction of application. For example, a gradient method can be applied to only the problems whose objective function is differentiable, and searches by using the differentiated objective function.

However it is desirable for optimization techniques not to restrict the applicable problems, so as to be applied to various actual engineering systems. Therefore, we study meta-heuristics because the restriction of applying of meta-heuristics is less than that of classic methods. Many meta-heuristics can be applied to problems in which evaluated value for a solution can be calculated. Meta-heuristics can be applied to the problems whose objective function can't be formulated by use of simulator that calculates objective value from a solution. Therefore meta-heuristics can be applied to complex systems.

One of the features of usual meta-heuristics is existence of some parameters to be adjusted. It can solve various optimization problems efficiently by adjusting their parameters properly, which indicates that we can't solve a problem efficiently without adjusting their parameters properly. Proper parameters are different in each problem, and finding proper parameters needs trial and error. Considering application to an engineering optimization problem, it is undesirable to solve a problem many times to adjust parameters. It is desirable to obtain as a good solution as possible in a restricted calculation time without trial and error for adjustment parameters. Therefore we think that meta-heuristics should have following abilities.

(1) robustness: The ability that will guarantee the performance of search for pre-adjusted parameters against predetermined structural variation of problems to be solved.

(2) adaptability: The ability that will guarantee the adaptable adjustment of their parameters against predetermined structural variation of problems to be solved.

We have studied Particle Swarm Optimization (PSO), which is one of the meta-heuristics is a powerful algorithm for solving unconstrained and constrained global optimization problems. And we have proposed an adaptive algorithm of PSO. In this paper, we point out that diversity of particles by setting diverse values to the parameters of each particle realizes robustness of PSO. We examine adding diversity of particles to the adaptive PSO that we proposed. And we propose an algorithm that has diversity of the parameters of particles and an adaptive strategy for tuning the parameters. Some numerical simulations were carried out in order to examine the search ability of the proposed approach.

2 Outline of Particle Swarm Optimization (PSO)

2.1 Background of PSO

In this section, the background, the fundamental concept and the details of PSO are referred. As expressed in the name, the algorithm of PSO has a swarm composed of multi-particles. Each particle has its own position and velocity (transfer vector). The search by multi-particles distinguishes PSO from many optimization methods proposed so far. Each particle shares global information of $gbest$, and interaction among particles makes their search efficient. Although only simple operations compose the PSO algorithm, the PSO method can solve nonlinear and multi-peaked optimization problems efficiently.

2.2 Algorithm of PSO

Although PSO was originally invented in the process of the research on simulating the movement of the swarm in the 2-dimensional space, PSO as an optimization method can work in the optional n -dimensional space.

Each particle has its own position x and transfer vector v. And each particle also has its own $pbest$, or the best position visited so far, and all particles share $gbest$, or the best position visited by all the particles so far. A particle forms a new transfer vector by linearly combining three vectors.

In the $(k+1)$ -th transfer, the j -th coordinate component of transfer vector of the i -th particle is manipulated according to the following equation:

$$\begin{aligned} v_{ij}^{k+1} = w \cdot v_{ij}^{k} &+ c_1 \cdot rand_1() \cdot (pbest_{ij} - x_{ij}^{k}) \\ &+ c_2 \cdot rand_2() \cdot (gbest_{ij} - x_{ij}^{k}) \end{aligned} \quad (1)$$

where $i = 1, \cdots, m$ and m is the size of the swarm; $j = 1, \cdots, n$ is the size of space of a given problem; w, c_1 and c_2 are positive constants; rand() is random number which is uniformly distributed in [0,1]; and k determines the iteration number.

Customarily, the transfer vector v_{ij} is also called velocity in PSO; therefore this paper follows this custom. Now each particle moves according to the following equation:

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1} \quad (2)$$

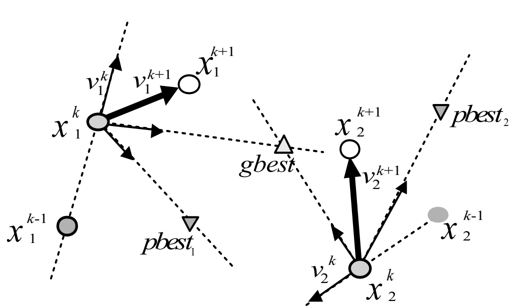

Figure 1. Movement of particles

3 Robustness and Adaptability of Meta-heuristics

We think that meta-heuristics should have the following abilities, i.e., robustness and adaptability.

Robustness is the ability that will guarantee the performance of search for pre-adjusted parameters against predetermined structural variation of problems to be solved. In the algorithm with low robustness, a certain parameter is efficient for small set of problems; it can solve efficiently the problem for which parameters are adjusted, while it can't solve efficiently the problems that have different structure. On the other hand, in the algorithm with high robustness, a certain parameter is efficient for large set of problems; it can efficiently solve the problems for which parameters are adjusted and problems which have different structure.

The adaptability of an optimization method means the ability that will guarantee the adaptable adjustment of their parameters against predetermined structural variation of problems to be solved. We can expect that the algorithm that has the adaptability will obtain a good solution for many problems without strict pre-adjustment of parameters because the algorithm can adjust their parameters by itself.

Figure 2 shows the image of an algorithm that has the robustness and an algorithm that has the adaptability.

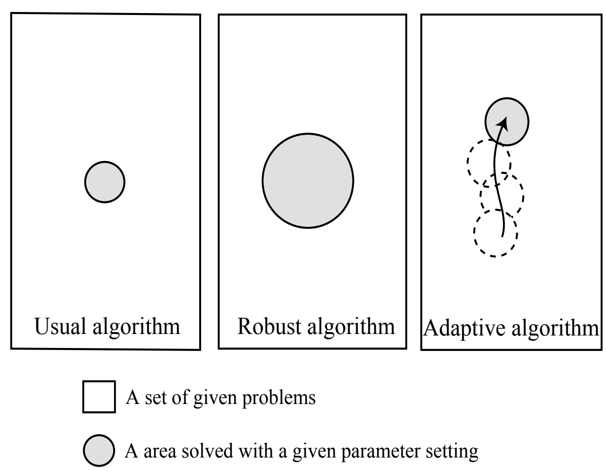

Figure 2. Concept of robustness and adaptability

4 Diversity of Parameters

In the meta-heuristics with multiple search points such as PSO, We can give the parameters of each search point different values or same values. Trajectories of particles of PSO are affected by given parameters w, c_1 and c_2 . When the parameters of each particle are given uniformly, the trajectories of each particle are similar. And it is expected that swarm will search efficiently for favorite problems, but it will not search efficiently for not favorite problems. On the other hand, when parameters of each particle are given unique values, the trajectories of each particle are different. And, it is expected that each particle works efficiently for different problems, and the swarm solves various problems efficiently. If the above-mentioned feature is realized, this is robustness.

Therefore, we verify whether diversity of particles realizes robustness of PSO by numerical simulations. At first, parameters are adjusted for the 50-dimensional problem shown in Equation (3). Next, we compare the performance of PSO having uniform parameters with that having diverse parameters for different dimensional problem. In this simulation, we give diversity of parameters at values of w, values of c_1+c_2, ratios of $c_1:c_2$. Values of w and values of c_1+c_2 ratios of $c_1:c_2$ affect the direction of search of particles. It is thought that these values and ratios affect search ability strongly. Table 1 shows the values of parameters that were adjusted for the 50-dimensional problem of Equation (3).

$$\min\ f=\sum_{i=1}^{n}(x_i^4-16x_i^2+5x_i) \tag{3}$$

$$\text{subj. to } -5\le x_i\le 5$$

Figures 4, 5 and 6 show the simulation results. The performance of PSO having diverse w, c_1+c_2, $c_1:c_2$ for 50-dimensional function are worse than that of PSO having uniform parameters. As the number of dimension of the problem changes, the performances of both algorithms deteriorate. Deterioration of PSO having diverse parameters is less than that of PSO having uniform parameters. It is thought that giving diversity of the parameters of each particle realizes the robustness of PSO from these simulation results.

Tabel. 1 Values of parameters adjusted for 50-dimensional problem

		Uniform parameters	Diverse parameters
Simulation1	w	0.65	$0.4, 0.44, \cdots, 0.8$
	c_1, c_2	1.5	1.5
Simulation2	w	0.6	0.6
	c_1, c_2	1.6	$1.3, 1.33, \cdots, 1.9$
Simulation3	w	0.6	0.6
	c_1	1.6	$0.7, 0.8, \cdots, 2.7$
	c_2	1.6	$2.7, 2.6, \cdots, 0.7$

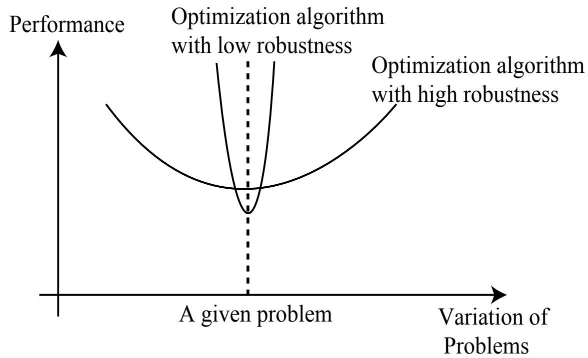

Figure 3. Image of relation between robustness and performance

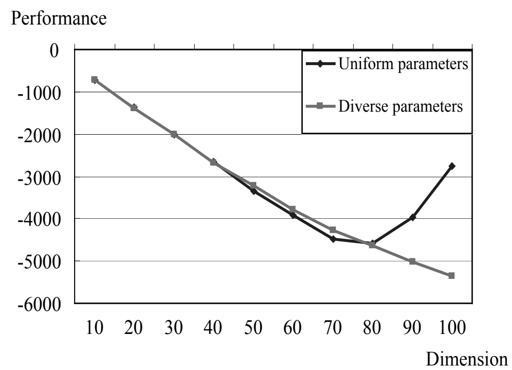

Figure 4. Result of simulation1(w)

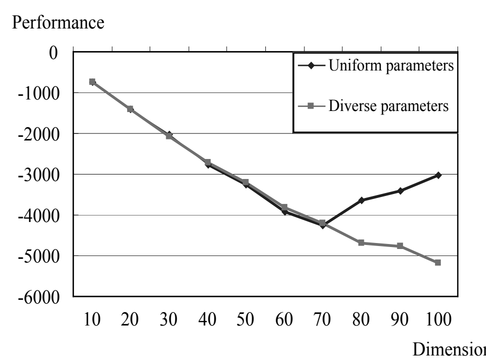

Figure 5. Result of simulation2(c_1+c_2)

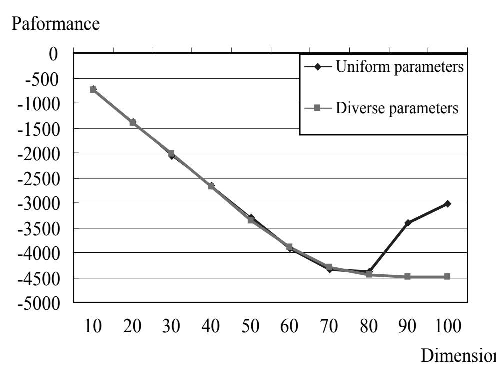

Figure 6. Result of simulation3($c_1:c_2$)

5 Robust-Adaptive PSO

5.1 Adaptive PSO

We have proposed adaptive strategy for tuning the parameters of the PSO method. Two significant preparations will be required in order to add the adaptability to an optimization algorithm. First, the qualitative and quantitative relationship between the parameters and the behavior of the algorithm must be analyzed. Second, the qualitative and quantitative relationship between the behavior of the algorithm and the success, or failure, of the search must be analyzed.

The qualitative and quantitative relationship between parameter w and the degree of convergence of PSO have been analyzed. And the quantitative relationship between the average of absolute value of velocity and the success, or failure of the search has been analyzed. The adaptive algorithm we proposed uses the information about the average of absolute value of velocity as indicator of search. Parameters are tuned adaptively in order to keep the goal velocity. In this algorithm, if the current average velocity of the swarm is larger than the goal velocity, parameter w is shifted to convergent values. Otherwise, parameter w is shifted to divergent values. By applying such a strategy, the average velocity of the swarm will follow the goal velocity.

In Figure7, the proposed ideal velocity reaches 0 before the end of the search, because the search is intended to be intensive in the final stage. And Figure 8 shows the transitions of parameter w, average velocity v_{ave}, and fitness $f(gbest)$ in the adaptive PSO.

$$v_{ave}^{k+1} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left| v_{ij}^{k+1} \right| \quad (4)$$

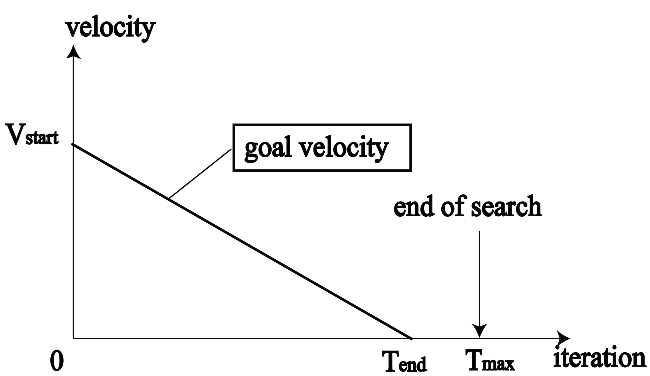

Figure 7. Goal of average velocity

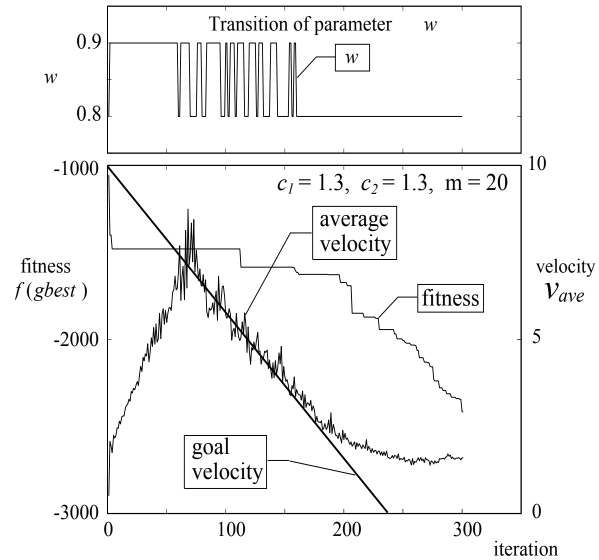

Figure 8. Transitions of parameter w, average velocity v_{ave}, and fitness $f(gbest)$ in the adaptive PSO

5.2 Robust-Adaptive PSO

We have been verified that the above-mentioned adaptive adjustment strategy of parameters obtains good solutions for many problems. In this strategy, parameters of particles are adjusted with uniform values. We examine adding diversity of particles to this adaptive adjustment strategy. It is expected that the algorithm will have the robustness from diversity of the particles and the adaptability from adaptive adjustment strategy.

We verified the performance of algorithm that has diversity of particles and adaptive adjustment strategy. The diversity of w and the diversity of $c_1 + c_2$ didn't improve the search ability of the adaptive PSO. The diversity of ratio of $c_1 : c_2$ improved the search ability of the adaptive PSO for many benchmark problems. Therefore we propose the algorithm that has diversity of $c_1 : c_2$ of particles and an adaptive adjustment strategy.

In this algorithm, the swarm is divided into some groups. And different ratio of $c_1 : c_2$ is given to each group. The average velocities of each group are calculated by Equation (4), w of each group is adjusted separately in order to keep the goal velocity. But the information of $gbest$ is shared with all particles.

[Robust Adaptive PSO Algorithm]

Step1: Generate the initial position x^0 and velocity v^0 of each search point randomly. Set $GoalVelocity^1 := V_{start}$ and $k = 0$, where k is an iteration counter.

Step2: Evaluation: Calculate the evaluated value $f(x^k)$ for each solution x^k.

Step3: Updating of $pbest$ and $gbest$: Compare each evaluated value $f(x^k)$ with the current $pbest$. If $f(x^k) < f(pbest^{k-1})$, set $pbest^k := x^k$. Otherwise, set $pbest^k := pbest^{k-1}$. If $k = 0$, set $pbest^0 := x^0$ without comparison. Compare and update $gbest$, in the same way as $pbest$.

Step4: Transfer of each particle: According to Equations (1) and (2), each particle x^k is transferred to x^{k+1}. Furthermore, ${V_{ave}}^{k+1}$ is calculated by Equation (4).

Step5: Modification of parameters: Compare the current average velocity of each group ${V_{ave\,g}}^{k+1}$ with $GoalVelocity^{k+1}$. If ${V_{ave\,g}}^{k+1} > GoalVelocity^{k+1}$, shift the parameters w, c_1, c_2 of the group to convergent values. Otherwise, shift the parameters of the group to divergent values.

Step6: Modification of $GoalVelocity$: Modify $GoalVelocity^{k+1}$ according to the following equation:

$$GoalVelocity^{k+2} = GoalVelocity^{k+1} - \frac{V_{start}}{T_{end}} \quad (5)$$

If counter k does not reach the predetermined maximum number of iterations $T_{\max}$, set $k = k+1$ and return to Step 2.

5.3 Numerical simulation

We compare the proposed method with the adaptive PSO that has uniform parameters and PSO that has scheduling strategy by the numerical simulation.

It is known that the scheduling strategy that reduces w during the search process is effective for various optimization problems. For example, $w = 0.9$ at the beginning of search process), $w = 0.4$ (at the end of search process), $c_1 = c_2 = 2.0$ (changeless), are recommended. In this paper, we call this strategy LDIWA. In this comparison, the recommended values of parameters of each algorithm is used to decide parameters except w, c_1 and c_2. Table2 lists adopted values, which is recommended values. $v_{\max}$ indicates the upper limit of the absolute velocity of each particle. v_{start} indicates goal velocity at the earliest of search. v_{end} indicates the time goal velocity reaches 0. The number of particles is 30 at each algorithm. Swarm of proposed method is divided into 3 groups. different ratio of $c_1 : c_2$ is given to each group.

Tables 3 and 4 list the comparison of the result of applying the LDIWA and the adaptive PSO and proposed method to the benchmark problems. The performance of the proposed method is better than that of LDIWA and that of the adaptive PSO having uniform parameters on many problems.

[Sphere]

$$\min\ f = \sum_{i=1}^{n} x_i^2 \quad (6)$$
$$\text{subj. to } -5 \le x_i \le 5$$

[2^n minima]

$$\min\ f = \sum_{i=1}^{n} (x_i^4 - 16x_i^2 + 5x_i) \quad (7)$$
$$\text{subj. to } -5 \le x_i \le 5$$

[Rastrigin]

$$\min\ f = \sum_{i=1}^{n} (x_i^2 - 10\cos 2\pi x_i + 10) \quad (8)$$
$$\text{subj. to } -5 \le x_i \le 5$$

[Rossenbrocks]

$$\min\ f = \sum_{i=1}^{n/2} \{100(x_i^2 - x_{i+\frac{n}{2}})^2 + (1 - x_i)^2\} \quad (9)$$
$$\text{subj. to } -5 \le x_i \le 5$$

[Six-humpCamelback]

$$\min f = \sum_{i=1}^{n/2} \{(4 + 2.1x_i^2 + \frac{1}{3}x_i^4)x_i^2 + x_i x_{i+\frac{n}{2}} + (-4 + 4x_{i+\frac{n}{2}}^2)x_{i+\frac{n}{2}}^2\} \quad (10)$$
$$\text{subj. to } -5 \le x_i \le 5$$

Tabe2. Values of parameters for comparison

LDIWA	w_{start}	0.9
	w_{start}	0.4
	$v_{\max}$	Maximum width of constraint
	c_1, c_2	(2, 2)
Adaptive PSO	v_{start}	Maximum width of constraint
	v_{end}	80% of maximum iteration
	c_1, c_2	(1.3, 1.3)
Robust Adaptive PSO	v_{start}	Maximum width of constraint
	v_{end}	80% of maximum iteration
	c_1, c_2	(2.0, 0.6), (1.3, 1.3), (0.6, 2.0)

6 Conclusion

In this paper, we point out that diversity of parameters of each particle gives PSO robustness. And we propose an algorithm that has the diversity of group of particles and adaptive strategy for tuning parameters of each group. Some numerical simulations were carried out in order to examine the search ability of the proposed approach.

References

[1] James Kennedy and Russell C. Eberhart : "Swarm Intelligence", Morgan Kaufmann Publishers (2001)

[2] K. Yasuda and N. Iwasaki: "Adaptive Particle Swarm Optimization using Velocity Information of Swarm", 2004 IEEE International Conference on Systems, Man and Cybernetics(2004)

Tabel. 3. Application results (Iteration = 300)

Function	Dim.	LDIWA				Adaptive PSO				Robust Adaptive PSO			
		Mean Value	Best Value	Worst Value	Standard Deviation	Mean Value	Best Value	Worst Value	Standard Deviation	Mean Value	Best Value	Worst Value	Standard Deviation
Sphere	10	0	0	0	0	0	0	0	0	0	0	0	0
	50	164	48	346	68	22	9	42	6	16	7	29	5
	100	679	603	745	32	101	65	140	16	87	56	122	13
	200	1447	1346	1537	47	260	200	333	29	234	173	335	30
2^n minima	10	-775	-783	-727	14	-758	-783	-698	23	-761	-783	-699	24
	50	-1761	-2423	-1008	358	-2913	-3201	-2544	141	-2956	-3220	-2664	140
	100	-2097	-2792	-1567	274	-4895	-5488	-4255	232	-4963	-5418	-4521	193
	200	-3416	-4606	-2695	401	-8296	-8911	-7616	297	-8542	-9344	-7928	314
Rastrigin	10	10	2	24	5	11	3	26	5	10	2	24	5
	50	641	457	820	85	287	220	402	35	244	149	343	39
	100	1620	1448	1727	52	738	602	908	65	676	538	848	64
	200	3373	3199	3521	73	1695	1524	1968	91	1619	1370	1865	106
Rossen brocks	10	7	1	23	5	8	1	24	5	6	0	21	5
	50	85653	10592	209320	52385	1843	517	4765	937	1094	179	3973	756
	100	435260	276210	545000	51350	12226	5865	23407	3445	9315	3282	21289	3284
	200	989700	753070	1209500	84409	39923	25493	61545	8063	34433	18652	59897	8330
Six-hump Camelback	10	-5	-5	-4	0	-5	-5	-4	0	-5	-5	-4	0
	50	5610	648	17506	3711	50	0	220	40	26	-3	85	18
	100	34388	26110	40869	3386	536	151	1138	228	403	103	979	162
	200	77306	59840	87236	4768	1847	728	3129	477	1304	477	2689	398

Tabel. 4. Application results (Iteration = 1000)

Function	Dim.	LDIWA				Adaptive PSO				Robust Adaptive PSO			
		Mean Value	Best Value	Worst Value	Standard Deviation	Mean Value	Best Value	Worst Value	Standard Deviation	Mean Value	Best Value	Worst Value	Standard Deviation
Sphere	10	0	0	0	0	0	0	0	0	0	0	0	0
	50	3	0	56	6	6	2	17	3	2	0	7	1
	100	681	492	745	43	63	42	92	11	46	25	72	10
	200	1456	1292	1556	52	226	180	296	24	189	124	248	20
2^n minima	10	-783	-783	-783	0	-778	-783	-755	11	-781	-783	-755	8
	50	-3014	-3374	-1952	269	-3213	-3445	-2910	103	-3296	-3512	-2910	110
	100	-2062	-2845	-1428	283	-5463	-5900	-5073	181	-5542	-5955	-5049	198
	200	-3434	-4396	-2580	389	-9096	-9792	-8383	302	-9405	-10385	-8722	315
Rastrigin	10	4	0	12	2	7	2	16	3	5	0	15	3
	50	335	170	641	96	239	162	342	34	184	122	273	29
	100	1630	1477	1729	46	704	526	929	73	604	412	730	61
	200	3373	3058	3512	70	1636	1407	1957	103	1588	1378	1821	96
Rossen brocks	10	2	0	16	3	5	0	21	4	3	0	15	3
	50	6178	60	110230	14200	332	81	1604	272	104	38	520	64
	100	422720	234670	541930	56881	6570	1589	13805	2303	4121	1067	7771	1354
	200	990030	778610	1181400	80548	32626	17603	54647	7188	24565	12247	44769	6374
Six-hump Camelback	10	-5	-5	-5	0	-5	-5	-5	0	-5	-5	-5	0
	50	15	-20	184	33	0	-16	34	8	-9	-19	6	5
	100	34197	25123	41230	3341	231	57	636	113	134	35	387	63
	200	76910	66236	86857	4502	1454	621	2933	433	971	434	1918	324