ANALYSINBG THE MIGRATION PERIOD PARAMETER IN PARALLEL MULTI-SWARM PARTICLE SWARM OPTIMIZATION

Şaban Gülcü[1] and Halife Kodaz[2]

[1]Department of Computer Engineering, Necmettin Erbakan University, Konya, Turkey

[2]Department of Computer Engineering, Selcuk University, Konya, Turkey

ABSTRACT

In recent years, there has been an increasing interest in parallel computing. In parallel computing, multiple computing resources are used simultaneously in solving a problem. There are multiple processors that will work concurrently and the program is divided into different tasks to be simultaneously solved. Recently, a considerable literature has grown up around the theme of metaheuristic algorithms. Particle swarm optimization (PSO) algorithm is a popular metaheuristic algorithm. The parallel comprehensive learning particle swarm optimization (PCLPSO) algorithm based on PSO has multiple swarms based on the master-slave paradigm and works cooperatively and concurrently. The migration period is an important parameter in PCLPSO and affects the efficiency of the algorithm. We used the well-known benchmark functions in the experiments and analysed the performance of PCLPSO using different migration periods.

KEYWORDS

Particle Swarm Optimization, Migration Period, Parallel Algorithm, Global Optimization

1. INTRODUCTION

In recent years, there has been an increasing interest in parallel computing. Software applications developed by using conventional methods run on a computer with limited resources as serial computing. Software executed by a processor on a computer consists of a collection of instructions. Each instruction is processed after another. An instruction is only processed at a time. But in parallel computing, multiple computing resources are used simultaneously in solving a problem. There are multiple processors that will work concurrently and the program is divided into different tasks to be simultaneously solved. Each task is divided into different instructions. The instructions are processed on different processors at the same time. Thus, performance increases and computer programs run in a shorter time. Parallel computing has been used in many different fields such as cloud computing [1], physics [2] and nanotechnology [3].

Recently, a considerable literature has grown up around the theme of metaheuristic algorithms. Particle swarm optimization (PSO) algorithm is developed by Kennedy and Eberhart in 1995 [4] is a popular metaheuristic algorithm. It is a population-based and stochastic optimization technique. It inspired from the social behaviours of bird flocks. Each individual in the population, called particle, represents a potential solution. PSO has been applied to many various fields such as automotive industry [5], energy [6], synchronous motor design [7], bioinformatics [8]. In recent years, many algorithms based on PSO have been developed such as the comprehensive learning PSO (CLPSO) algorithm [9] and the parallel comprehensive learning particle swarm optimization (PCLPSO) algorithm [10]. In recent years, devising parallel models of algorithms has been a healthy field for developing more efficient optimization procedures [11-14]. Parallelism is an approach not only to reduce the resolution time but also to improve the quality of the provided solutions. In CLPSO, instead of using a particle's best information in the original PSO, all other particles' historical best information is used to update the particle's velocity. Further, the global best position of population in PSO is never used in CLPSO. With this strategy, CLPSO searches a larger area and the probability of finding global optimum is increased. The PCLPSO algorithm based on CLPSO has multiple swarms based on the master-slave paradigm and works cooperatively and concurrently. Through PCLPSO, the solution quality and the global search ability are improved. This article studies the effect of the different migration periods on PCLPSO algorithm.

This article has been organized in the following way: Section 2 is concerned with the methodologies used for this study. Section 3 presents the experimental results and the findings of the research. Finally, the article is concluded in Section 4.

2. MATERIALS & METHODS

2.1. PSO

Each particle in PSO represents a bird and offers a solution. Each particle has a fitness value calculated by fitness function. Particles have velocity information and position information updated during the optimization process. Each particle searches the food in the search area using the velocity and position information. PSO aims to find the global optimum or a solution close to the global optimum and therefore is launched with a random population. The particles update their velocity and position information by using Equations (1) and (2) respectively. To update the position of a particle, *pbest* of the particle and *gbest* of the whole population are used. *pbest* and *gbest* are repeatedly updated during the optimization process. Thus, the global optimum or a solution close to the global optimum is found at the end of the algorithm.

$$V_i^d = w * V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (gbest^d - X_i^d) \quad (1)$$

$$X_i^d = X_i^d + V_i^d \quad (2)$$

where V_i^d and X_i^d represent the velocity and the position of the *d*th dimension of the particle *i*. The constant *w* is called inertia weight plays the role to balance between the global search ability and local search ability [15]. c_1 and c_2 are the acceleration coefficients. *rand*1 and *rand*2 are the two random numbers between 0 and 1. They affect the stochastic nature of the algorithm [16]. $pbest_i$ is the best position of the particle *i*. *gbest* is the best position in the entire swarm. The inertia weight *w* is updated according to Equation (3) during the optimization process.

$$w(t) = w_{\max} - t * (w_{\max} - w_{\min}) / T \quad (3)$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum inertia weights and usually set to 0.9 and 0.2 respectively [15]. *t* is the actual iteration number and *T* is the maximum number of iteration cycles.

2.2. CLPSO

CLPSO based on PSO was proposed by Liang, Qin, Suganthan and Baskar [9]. PSO has some deficiencies. For instance, if the *gbest* falls into a local minimum, the population can easily fall into this local minimum. For this reason, CLPSO doesn't use *gbest*. Another property of CLPSO is that a particle uses also the *pbest*s of all other particles. This method is called as the comprehensive learning approach. The velocity of a particle in CLPSO is updated using Equation (4).

$$V_i^d = w * V_i^d + c * rand_i^d * (pbest_{fi(d)}^d - X_i^d) \quad (4)$$

where $\boldsymbol{f}_i = [f_i(1), f_i(2), \ldots, f_i(D)]$ is a list of the random selected particles which can be any particles in the swarm including the particle *i*. They are determined by the *Pc* value, called as learning probability, in Equation (5). $pbest_{fi(d)}^d$ indicates the *pbest* value of the particle which is stored in the list $\boldsymbol{f}_i$ of the particle *i* for the *d*th dimension. How a particle selects the *pbest*s for each dimension is explained in [9].

$$V_i^d = w * V_i^d + c * rand_i^d * (pbest_{fi(d)}^d - X_i^d) \quad (5)$$

CLPSO uses a parameter *m*, called the refreshing gap. It is used to learn from good exemplars and to escape from local optima. The flowchart of the CLPSO algorithm is given in [9].

2.3. PCLPSO

Although PSO has many advantages, the main deficiency of PSO is the premature convergence [16]. PCLPSO handles to overcome this deficiency like many PSO variants. The PCLPSO algorithm based on CLPSO was proposed by Gülcü and Kodaz [10]. The solution quality is enhanced through multiswarm and cooperation properties. Also, computational efficiency is improved because PCLPSO runs parallel on a distributed environment.

A population is split into subpopulations. Each subpopulation represents a swarm and each swarm independently runs PCLPSO algorithm. Thus, they seek the search area. There are two types of swarms: master-swarm and slave swarm. The number of the swarms is an important parameter in PCLPSO and we analysed the effects of the number of swarms on the PCLPSO algorithm in our previous work [17]. In the cooperation technique, each swarm periodically shares its own global best position with other swarms. The parallelism property is that each swarm runs the algorithm on a different computer at the same time to achieve computational efficiency. The topology is shown in Figure 1. Each swarm runs cooperatively and synchronously the PCLPSO algorithm to find the global optimum. PCLPSO uses Jade middleware framework [18] to establish the parallelism. The cluster specifications are so: windows XP operating system, pentium i5 3.10 GHz, 2 GB memory, java se 1.7, Jade 4.2 and gigabit ethernet. The flowchart of the PCLPSO algorithm is given in [10].

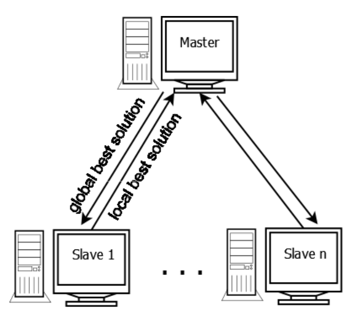

Figure 1. The communication topology [10]

In the communication topology, there isn't any directly communication between slave swarms as shown in Figure 1. Migration process occurs periodically after a certain number of cycles. Each swarm sends the own local best solution to the master in the PCLPSO's migration process. The master collects the local best solutions into a pool, called *ElitePool*. It chooses the best solution from the *ElitePool*. This solution is sent to all slave swarms by the master. Thus, PCLPSO obtains better and more robust solutions.

3. EXPERIMENTAL RESULTS

The experiments performed in this section were designed to study the behaviour of PCLPSO by varying the migration period. The migration period is an important parameter in PCLPSO and affects the efficiency of the algorithm. This article analyses the effect of the migration period on PCLPSO algorithm.

Two unimodal and 12 multimodal benchmark functions which are well known to the global optimization community and commonly used for the test of optimization algorithms are selected. The formulas of the functions are given in next subsection. The properties of these functions are given in Table 1. The number of particles per swarm is 15. According to the dimensions of functions, the experiments are split into three groups. The properties of these groups are given in Table 2. The term FE in the table refers the maximum fitness evaluation.
The experiments are carried out on a cluster whose specifications are windows XP operating system, pentium i5 3.10 GHz, 2 GB memory, java se 1.7, Jade 4.2 and gigabit ethernet. The inertia weight *w* linearly decreases from 0.9 to 0.2 during the iterations, the acceleration coefficient *c* is equal to 1.49445 and the refreshing gap *m* is equal to five. 30 independent tests are carried out for each function. The results are given in next subsections.

Table 1. Type, Global Minimum, Function Value, Search and Initialization Ranges of the Benchmark Functions.

f	Global Minimum x^*	Function Value $f(x^*)$	Search Range	Initialization Range
f_1	[0,0,…,0]	0	$[-100, 100]^D$	$[-100, 50]^D$
f_2	[1,1,…,1]	0	$[-2.048, 2.048]^D$	$[-2.048, 2.048]^D$
f_3	[0,0,…,0]	0	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
f_4	[0,0,…,0]	0	$[-600, 600]^D$	$[-600, 200]^D$
f_5	[0,0,…,0]	0	$[-0.5, 0.5]^D$	$[-0.5, 0.2]^D$
f_6	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_7	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_8	[420.96,420.96,…,420.96]	0	$[-500, 500]^D$	$[-500, 500]^D$
f_9	[0,0,…,0]	0	$[-32.768, 32.768]^D$	$[-32.768, 16]^D$
f_{10}	[0,0,…,0]	0	$[-600, 600]^D$	$[-600, 200]^D$
f_{11}	[0,0,…,0]	0	$[-0.5, 0.5]^D$	$[-0.5, 0.2]^D$
f_{12}	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_{13}	[0,0,…,0]	0	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$
f_{14}	[0,0,…,0]	0	$[-500, 500]^D$	$[-5.12, 5.12]^D$

Table 2. Parameters used in the experiments.

Dimension	FE	Number of swarms	Number of particles
10	$3x10^4$	4	15
30	$2x10^5$	4	15
100	$3x10^5$	4	15

3.1. FUNCTIONS

The functions used in the experiments are the following:

Sphere function:

$$f_1(x) = \sum_{i=1}^{D} x_i^2 \quad (6)$$

Rosenbrock function:

$$f_2(x)=\sum_{i=1}^{D-1}[100(x_i^2-x_{i+1})^2+(x_i-1)^2] \tag{7}$$

Ackley function:

$$f_3(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+20+e \tag{8}$$

Griewank function:

$$f_4(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1 \tag{9}$$

Functions f_1 and f_2 are unimodal. Unimodal functions have only one optimum and no local minima.

Weierstrass function:

$$f_5(x)=\sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left(a^k\cos\left(2\pi b^k(x_i+0.5)\right)\right)\right)-D\sum_{k=0}^{k\max}\left(a^k\cos\left(2\pi b^k*0.5\right)\right) \tag{10}$$

a=0.5, b=3, $k_{\max}=20$.

Rastrigin function:

$$f_6(x)=10D+\sum_{i=1}^{D}\left[x_i^2-10\cos(2\pi x_i)\right] \tag{11}$$

Noncontinuous Rastrigin function:

$$f_7(x)=10D+\sum_{i=1}^{D}\left[y_i^2-10\cos(2\pi y_i)\right] \tag{12}$$

$$y_i=\begin{cases} x_i & |x_i|<\frac{1}{2} \\ \frac{round(2x_i)}{2} & |x_i|>=\frac{1}{2} \end{cases} \quad for\ i=1,2,...,D.$$

Schwefel function:

$$f_8(x)=418.9829D-\sum_{i=1}^{D}x_i\sin\left(\sqrt{|x_i|}\right) \tag{13}$$

Functions f_3 - f_8 are multimodal. Multimodal functions have only one optimum and many local minima. They are treated as a difficult class of benchmark functions by researchers because the number of local minima of the function grows exponentially as the number of its dimension increases [19-22]. Therefore, obtaining good results on multimodal functions is very important for the optimization algorithms.

Functions f_9 - f_{14} are the rotated version of the f_3 - f_8. The rotation changes the separable functions into the no separable functions which are solved harder. A separable function is rotated by using Equation (14). The matrix **M** in the formula refers to an orthogonal matrix [23] and the variable **y** refers the new input vector of the function.

$$\mathbf{y}=\mathbf{M}*\mathbf{x} \tag{14}$$

Rotated Ackley function:

$$f_9(x)=-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}y_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi y_i)\right)+20+e \tag{15}$$

where $\boldsymbol{y}=\mathbf{M}*\boldsymbol{x}$.

Rotated Griewank function:

$$f_{10}(x) = \sum_{i=1}^{D} \frac{y_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{y_i}{\sqrt{i}}\right) + 1 \tag{16}$$

where $\boldsymbol{y} = \mathbf{M} * \boldsymbol{x}$.

Rotated Weierstrass function:

$$f_{11}(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left(a^k \cos\left(2\pi b^k (y_i + 0.5)\right)\right)\right) - D\sum_{k=0}^{k\max}\left(a^k \cos\left(2\pi b^k * 0.5\right)\right) \tag{17}$$

where a=0.5, b=3, $k_{\max}$ = 20, $\boldsymbol{y} = \mathbf{M} * \boldsymbol{x}$.

Rotated Rastrigin function:

$$f_{12}(x) = 10D + \sum_{i=1}^{D}\left[y_i^2 - 10\cos(2\pi y_i)\right] \tag{18}$$

where $\boldsymbol{y} = \mathbf{M} * \boldsymbol{x}$.

Rotated Noncontinuous Rastrigin function:

$$f_{13}(x) = 10D + \sum_{i=1}^{D}\left[z_i^2 - 10\cos(2\pi * z_i)\right] \tag{19}$$

$$z_i = \begin{cases} y_i & if\,|y_i| < \frac{1}{2} \\ \frac{round(2y_i)}{2} & if\,|y_i| >= \frac{1}{2} \end{cases} \quad for\ i = 1,...,D.$$

where $\boldsymbol{y} = \mathbf{M} * \boldsymbol{x}$.

Rotated Schwefel function:

$$f_{14}(x) = 418.9829 * D - \sum_{i=1}^{D} z_i \tag{20}$$

$$z_i = \begin{cases} y_i \sin\left(\sqrt{|yi|}\right) & if\ |y_i| <= 500 \\ 10^{-3} * \left(|y_i| - 500\right)^2 & if\ |y_i| > 500 \end{cases} \quad for\ i = 1,...,D.$$

where $\boldsymbol{y} = \boldsymbol{y'} + 420.96$, $\boldsymbol{y'} = \mathbf{M} * (\boldsymbol{x} - 420.96)$.

3.2. RESULTS OF THE 10-D PROBLEMS

Table 3 presents the mean of the function values for 10-D problems according to the different migration periods. Table 4 presents the calculation time of the functions for 10-D problems. In [10], the importance of the migration period is emphasized: if the information is very often exchanged, then the solution quality may be better, but the computational efficiency deteriorates. If the migration period is longer, the computational efficiency is better, but the solution quality may be worse. It is apparent from these tables that the computational efficiency is better when the migration period is equal to 100 as expected. But the better values of functions f_1-f_{14} are obtained when the migration period is around 6. The bold text in the tables refers the best results.

Table 3. The mean values for 10-D problems.

P	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1	2.45e-03	7.20e+00	9.02e-02	1.11e-01	1.93e-02	8.93e-01	0.00e+00
2	4.71e-03	**6.70e+00**	8.23e-02	1.38e-01	4.66e-02	9.29e-01	0.00e+00
3	5.66e-03	7.51e+00	2.51e-01	1.49e-01	2.85e-02	1.01e+00	0.00e+00
4	3.28e-03	7.03e+00	1.23e-01	1.43e-01	**1.52e-02**	6.49e-01	0.00e+00
5	3.70e-03	7.68e+00	6.57e-02	1.28e-01	3.18e-02	8.61e-01	0.00e+00
6	4.03e-03	7.94e+00	**6.49e-02**	1.29e-01	3.58e-02	**4.59e-01**	0.00e+00
7	3.24e-03	7.32e+00	8.10e-02	1.36e-01	3.47e-02	1.29e+00	0.00e+00
8	2.15e-03	7.17e+00	9.52e-02	1.38e-01	1.96e+04	1.92e+03	0.00e+00
9	4.23e-03	7.90e+00	9.71e-02	1.40e-01	4.28e-02	6.48e-01	0.00e+00
10	3.87e-03	8.98e+00	7.67e-02	1.25e-01	3.62e-02	7.14e-01	0.00e+00
11	**1.97e-03**	7.17e+00	1.08e-01	1.28e-01	2.90e-02	1.40e+00	0.00e+00
12	3.69e-03	7.78e+00	1.46e-01	1.43e-01	5.82e-02	2.35e+00	0.00e+00
13	3.86e-03	8.26e+00	1.42e-01	**1.03e-01**	3.74e-02	1.60e+00	0.00e+00
14	2.99e-03	7.16e+00	1.09e-01	1.24e-01	3.42e-02	9.23e-01	0.00e+00
15	3.41e-03	8.49e+00	9.15e-02	1.18e-01	6.92e-02	1.26e+00	0.00e+00
16	3.55e-03	8.79e+00	1.96e-01	1.27e-01	4.53e-02	1.13e+00	0.00e+00
17	3.21e-03	7.47e+00	2.64e-01	1.32e-01	4.39e-02	5.52e-01	0.00e+00
18	4.13e-03	8.16e+00	1.64e-01	1.37e-01	4.86e-02	4.86e-01	0.00e+00
19	4.02e-03	7.08e+00	9.34e-02	1.51e-01	4.24e-02	1.07e+00	0.00e+00
20	3.97e-03	6.84e+00	1.31e-01	1.29e-01	3.43e-02	1.25e+00	0.00e+00
50	2.96e-03	7.92e+00	1.26e-01	1.30e-01	5.55e-02	1.21e+00	0.00e+00
100	3.63e-03	6.90e+00	2.55e-01	1.25e-01	3.77e-02	2.14e+00	0.00e+00

Table 3. The mean values for 10-D problems. (cont.)

P	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
1	3.26e-02	4.71e-02	**5.22e-02**	4.35e-02	4.48e-01	0.00e+00	5.01e-01
2	1.06e-01	8.89e-02	1.33e-01	3.13e-02	1.18e+00	0.00e+00	7.91e+01
3	4.54e-02	1.89e-01	1.14e-01	4.61e-02	8.70e-01	0.00e+00	3.95e+01
4	7.99e+01	9.77e-02	9.93e-02	3.86e-02	1.61e+00	0.00e+00	8.64e-01
5	3.95e+01	4.79e-01	1.01e-01	5.60e-02	1.20e+00	0.00e+00	3.93e-01
6	3.95e+01	**2.59e-02**	2.00e-01	5.67e-02	1.22e+00	0.00e+00	**4.13e-02**
7	3.77e-02	8.75e-02	1.39e-01	**2.72e-02**	1.07e+00	0.00e+00	8.31e-02
8	1.17e+03	1.32e+03	1.96e+03	1.97e+04	1.94e+03	0.00e+00	1.16e+03
9	3.95e+01	4.41e-02	1.81e-01	3.46e-02	1.22e+00	0.00e+00	4.93e-02
10	3.95e+01	4.55e-01	9.62e-02	3.88e-02	5.91e-01	0.00e+00	4.00e+01
11	4.53e+01	3.88e-01	8.86e-02	3.95e-02	5.16e-01	0.00e+00	3.97e+01
12	7.91e+01	8.06e-02	1.37e-01	5.13e-02	4.33e-01	0.00e+00	7.90e+01
13	5.68e-02	4.19e-02	1.85e-01	3.68e-02	8.46e-01	0.00e+00	3.95e+01
14	4.87e-02	3.47e-02	1.37e-01	3.85e-02	9.44e-01	0.00e+00	3.95e+01
15	2.41e-01	8.35e-02	9.55e-02	4.11e-02	1.23e+00	0.00e+00	3.98e+01
16	**1.25e-02**	7.43e-02	1.23e-01	3.26e-02	8.33e-01	0.00e+00	3.95e+01
17	3.97e+01	5.03e-01	1.78e-01	3.74e-02	2.15e+00	0.00e+00	1.58e+02
18	1.58e+02	2.96e-02	1.18e-01	5.08e-02	**3.95e-01**	0.00e+00	1.14e-01
19	1.18e+02	5.75e-02	1.31e-01	4.89e-02	8.72e-01	0.00e+00	7.90e+01
20	7.91e+01	3.63e-02	1.42e-01	3.74e-02	5.12e-01	0.00e+00	4.05e+01
50	1.13e-01	7.84e-02	1.37e-01	4.71e-02	1.48e+00	0.00e+00	3.96e+01
100	7.90e+01	3.71e-02	1.18e-01	5.67e-02	1.47e+00	0.00e+00	3.98e+01

Table 4. The calculation time (ms) for 10-D problems.

P	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1	4463	4484	10359	15532	153917	15479	6776
2	2230	2246	5184	7769	77281	7677	3365
3	1484	1497	3450	5163	51307	5089	2245
4	1122	1131	2600	3889	38818	3838	1687
5	901	907	2081	3112	31120	3068	1354
6	750	755	1732	2588	25927	2547	1130
7	643	648	1482	2214	22516	2198	979
8	564	567	1303	1932	19609	1922	854
9	501	507	1158	1723	17349	1687	755
10	458	462	1051	1563	15906	1536	687
11	413	417	946	1406	14292	1380	620
12	377	380	864	1285	13073	1266	568
13	351	353	804	1187	12141	1167	526
14	322	326	736	1094	11229	1078	485
15	305	307	694	1031	10620	1016	458
16	289	290	657	975	10005	953	432
17	271	272	614	911	9437	891	406
18	252	254	573	846	8896	844	385
19	243	244	551	815	8646	828	375
20	234	236	530	784	8281	787	359
50	101	102	220	319	3635	318	156
100	**56**	**57**	**116**	**163**	**2099**	**172**	**94**

Table 4. The calculation time (ms) for 10-D problems. (cont.)

P	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
1	9276	10438	15604	153557	15406	6740	9208
2	4609	5229	7807	77031	7656	3365	4609
3	3068	3484	5245	52078	5188	2287	3120
4	2312	2620	3906	38771	3833	1693	2312
5	1854	2099	3125	31182	3083	1365	1864
6	1536	1745	2599	25922	2547	1130	1542
7	1328	1505	2234	22271	2182	974	1323
8	1167	1318	1958	19693	1938	854	1162
9	1026	1156	1724	17354	1693	750	1026
10	932	1052	1573	15896	1537	688	932
11	844	948	1412	14302	1385	620	839
12	766	865	1286	13073	1261	568	766
13	714	807	1193	12151	1172	531	714
14	656	745	1099	11234	1073	490	656
15	620	698	1031	10620	1010	458	620
16	583	656	974	10016	953	432	583
17	547	615	912	9396	896	406	547
18	516	578	854	8901	838	386	515
19	505	573	844	8708	823	375	505
20	484	547	797	8281	786	365	479
50	198	224	328	3651	323	156	198
100	**115**	**125**	**172**	**2089**	**167**	**89**	**109**

3.3. Results of the 30-D problems

Table 5 presents the mean of the function values for 30-D problems according to the different migration periods. The better mean values of functions f_1-f_{14} are obtained when the migration periods are around 13 as shown from the results. Table 6 presents the calculation time of the function values for 30-D problems. The bold text in the tables refers the best results.

Table 5. The mean values for 30-D problems.

P	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1	**1.04e-09**	2.50e+01	1.49e-05	3.16e-06	1.69e-04	3.33e-01	0.00e+00
2	1.63e-09	2.38e+01	1.07e-05	6.14e-07	1.55e-04	3.34e-01	0.00e+00
3	3.17e-09	2.42e+01	1.48e-05	1.48e-06	1.75e-04	3.33e-01	0.00e+00
4	2.09e-09	2.25e+01	1.39e-05	1.09e-06	1.43e-04	2.23e-03	0.00e+00
5	1.12e-09	2.37e+01	1.77e-05	2.05e-06	1.67e-04	5.48e-04	0.00e+00
6	2.40e-09	2.40e+01	1.35e-05	1.92e-05	**1.26e-04**	9.96e-01	0.00e+00
7	4.05e-09	2.48e+01	1.08e-05	1.55e-06	1.49e-04	9.96e-01	0.00e+00
8	1.37e-09	2.31e+01	1.51e-05	6.05e-07	1.31e-04	9.86e-04	0.00e+00
9	2.15e-09	2.39e+01	1.31e-05	1.29e-05	1.77e-04	3.32e-01	0.00e+00
10	1.51e-09	**2.17e+01**	1.65e-05	1.42e-06	2.59e-04	6.67e-01	0.00e+00
11	1.52e-09	2.77e+01	**8.89e-06**	2.65e-06	1.60e-04	3.34e-01	0.00e+00
12	1.93e-09	3.14e+01	1.35e-05	**3.12e-07**	2.12e-04	3.33e-01	0.00e+00
13	1.32e-09	2.22e+01	1.20e-05	8.73e-07	2.04e-04	6.67e-01	0.00e+00
14	2.33e-09	2.59e+01	1.01e-05	7.74e-07	1.78e-04	6.43e-03	0.00e+00
15	4.27e-09	2.24e+01	1.07e-05	5.33e-07	1.86e-04	**4.59e-04**	0.00e+00
16	1.85e-09	2.53e+01	1.60e-05	1.99e-06	1.87e-04	5.56e-04	0.00e+00
17	1.78e-09	2.49e+01	1.40e-05	7.22e-07	1.73e-04	8.48e-04	0.00e+00
18	2.12e-09	2.53e+01	1.54e-05	4.80e-06	1.73e-04	1.41e-03	0.00e+00
19	3.17e-09	2.29e+01	1.37e-05	7.04e-07	1.51e-04	9.97e-01	0.00e+00
20	1.95e-09	2.52e+01	1.72e-05	1.71e-06	1.85e-04	3.32e-01	0.00e+00
50	2.63e-09	2.49e+01	1.73e-05	9.63e-06	1.37e-04	3.33e-01	0.00e+00
100	3.21e-09	2.29e+01	1.24e-05	2.37e-06	1.44e-04	3.33e-01	0.00e+00

Table 5. The mean values for 30-D problems. (cont.)

P	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
1	1.58e+02	1.35e-05	6.50e-07	1.48e-04	3.33e-01	0.00e+00	1.18e+02
2	1.51e+02	1.33e-05	3.03e-06	2.13e-04	9.96e-01	0.00e+00	1.58e+02
3	1.18e+02	1.37e-05	1.49e-06	2.48e-04	2.19e-03	0.00e+00	3.95e+01
4	1.18e+02	1.12e-05	4.25e-06	1.56e-04	3.34e-01	0.00e+00	1.51e+02
5	7.90e+01	1.09e-05	9.07e-07	**1.16e-04**	3.33e-01	0.00e+00	1.58e+02
6	7.90e+01	1.24e-05	2.85e-07	2.68e-04	3.33e-01	0.00e+00	1.18e+02
7	1.58e+02	1.42e-05	4.93e-07	1.42e-04	1.31e-03	0.00e+00	2.76e+02
8	1.97e+02	1.02e-05	5.70e-06	1.76e-04	6.65e-01	0.00e+00	7.90e+01
9	**3.82e-04**	1.35e-05	8.86e-07	2.34e-04	6.64e-01	0.00e+00	1.58e+02
10	1.18e+02	1.16e-05	5.73e-07	1.70e-04	1.37e-03	0.00e+00	1.58e+02
11	7.90e+01	1.63e-05	8.51e-07	1.43e-04	3.33e-01	0.00e+00	1.58e+02
12	3.95e+01	1.14e-05	1.86e-06	1.50e-04	**6.80e-04**	0.00e+00	1.18e+02
13	7.90e+01	1.20e-05	**1.31e-07**	2.34e-04	1.08e-03	0.00e+00	**3.82e-04**
14	7.90e+01	**9.08e-06**	7.53e-07	1.63e-04	8.36e-04	0.00e+00	1.58e+02
15	1.58e+02	1.43e-05	2.03e-06	2.41e-04	1.07e-03	0.00e+00	1.18e+02
16	7.90e+01	1.15e-05	1.95e-06	1.89e-04	3.35e-01	0.00e+00	3.95e+01
17	1.97e+02	1.24e-05	4.86e-07	2.03e-04	3.33e-01	0.00e+00	1.97e+02
18	1.18e+02	1.01e-05	5.36e-07	1.71e-04	3.32e-01	0.00e+00	3.82e-04
19	1.18e+02	1.55e-05	1.08e-06	1.53e-04	3.32e-01	0.00e+00	2.37e+02
20	2.76e+02	1.37e-05	2.40e-06	2.35e-04	1.33e+00	0.00e+00	3.16e+02
50	2.37e+02	2.33e-05	1.77e-06	1.47e-04	6.65e-01	0.00e+00	1.58e+02
100	1.97e+02	1.82e-05	1.53e-05	1.62e-04	3.32e-01	0.00e+00	7.90e+01

Table 6. The calculation time (ms) for 30-D problems.

P	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1	501775	507009	1172359	1974659	15948339	1945557	798995
2	250866	253531	586153	987428	7980537	972411	399469
3	167366	169169	390778	658059	5319880	647880	267761
4	125544	126884	293141	493516	3991068	487687	199557
5	100431	101484	234481	394644	3195115	388833	160021
6	83734	84672	195575	329072	2661375	325078	133026
7	71819	72603	167788	282494	2283589	277625	115063
8	62819	63466	146556	246678	1996885	243000	99677
9	55866	56447	130387	219428	1779724	217995	88687
10	50291	50831	117366	197528	1602245	195891	80693
11	45772	46262	106825	179744	1451844	177380	72719
12	41866	42291	97666	164303	1328594	161760	66505
13	38700	39109	90266	151909	1228172	149250	61474
14	35984	36359	83941	141228	1144005	139078	57182
15	33578	33928	78306	131734	1066698	129464	53849
16	31519	31891	73522	123447	1001339	121146	50011
17	29675	29981	69169	116419	943761	115297	47604
18	28031	28309	65288	109822	891266	108286	44917
19	26503	26784	61772	103956	842594	102911	42438
20	25150	25500	58628	98703	801474	96807	39802
50	10106	10216	23447	39425	321964	38849	16036
100	**5709**	**5859**	**12519**	**20522**	**166026**	**19396**	**8229**

Table 6. The calculation time (ms) for 30-D problems. (cont.)

P	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
1	1133031	1179693	1989922	15953240	1945870	799182	1132625
2	566000	589636	995141	7977234	973365	399604	566719
3	377073	392870	663896	5323563	648828	266250	377547
4	282766	295719	496531	3991557	485281	200875	282792
5	227630	235781	396563	3194146	388146	160708	226412
6	188677	196448	331984	2661036	323635	133021	188187
7	162130	168682	283714	2283886	277917	114109	162104
8	142198	147823	248208	1996719	243224	100328	141214
9	125990	131151	221162	1781865	216880	88703	125870
10	112917	118557	199417	1601620	195578	79906	112964
11	102786	107448	180286	1452932	176703	72708	102797
12	94031	98177	164938	1328844	161594	66526	94016
13	86943	91120	152703	1228037	149063	61552	87307
14	80755	84635	143021	1143708	138906	57188	80771
15	75724	78495	132213	1067276	129281	53349	75344
16	70739	73745	123818	1000411	121901	50552	70620
17	66531	69307	116651	943453	114552	47135	66542
18	63229	65609	110333	891213	107792	44526	63141
19	59636	61922	104198	843021	101958	42135	59833
20	56427	58750	99307	801073	97214	40130	56760
50	22781	23625	39677	324057	38860	16005	22578
100	**11406**	**11839**	**19865**	**166036**	**19500**	**8229**	**11526**

3.4. RESULTS OF THE 100-D PROBLEMS

Table 7 presents the mean of the function values for 100-D problems according to the different migration periods. The better mean values of functions f_1-f_{14} are obtained when the migration periods are around 12. Table 8 presents the calculation time of the functions for 100-D problems. The bold text in the tables refers the best results.

Table 7. The mean values for 100-D problems.

P	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1	7.04e-03	1.41e+02	3.46e-01	7.03e-03	5.34e-01	2.73e+01	0.00e+00
2	1.95e-02	1.50e+02	1.03e+00	9.48e-03	1.80e-01	3.31e+01	0.00e+00
3	1.69e-02	2.23e+02	2.27e-02	1.22e-02	2.01e-01	2.57e+01	0.00e+00
4	2.05e-02	1.46e+02	1.80e-02	2.76e-02	2.23e-01	2.90e+01	0.00e+00
5	9.95e-03	1.54e+02	**8.78e-03**	4.91e-02	1.81e-01	2.94e+01	0.00e+00
6	1.63e-02	9.65e+01	1.84e-02	1.53e-02	2.30e-01	2.59e+01	0.00e+00
7	8.94e-03	1.81e+02	1.74e-02	6.66e-03	3.34e-01	**2.29e+01**	0.00e+00
8	3.65e-02	9.89e+01	5.83e-01	2.43e-02	6.59e-01	2.80e+01	0.00e+00
9	1.67e-02	9.64e+01	4.70e-01	1.90e-02	2.63e-01	3.04e+01	0.00e+00
10	2.15e-02	1.49e+02	1.03e+00	5.71e-03	**1.33e-01**	2.57e+01	0.00e+00
11	**4.19e-03**	1.30e+02	1.91e-02	1.08e-02	2.35e-01	3.01e+01	0.00e+00
12	1.34e-02	2.04e+02	1.41e-02	7.42e-03	3.28e-01	2.82e+01	0.00e+00
13	1.16e-02	9.75e+01	1.29e-02	7.88e-03	3.47e-01	3.45e+01	0.00e+00
14	6.72e-02	9.82e+01	1.09e+00	1.97e-01	2.16e-01	2.72e+01	0.00e+00
15	5.70e-01	9.34e+01	1.56e-02	**3.93e-03**	4.35e-01	3.13e+01	0.00e+00
16	1.31e-02	1.80e+02	2.03e-02	5.04e-03	3.76e-01	3.80e+01	0.00e+00
17	4.63e-02	**9.10e+01**	2.94e-02	8.61e-03	1.96e-01	2.54e+01	0.00e+00
18	3.27e-02	1.48e+02	1.03e+00	1.99e-02	2.20e-01	2.93e+01	0.00e+00
19	1.32e-02	9.74e+01	2.56e-02	2.58e-02	2.36e-01	2.91e+01	0.00e+00
20	1.68e-02	1.01e+02	1.40e-02	1.83e-02	1.85e-01	2.88e+01	0.00e+00
50	3.02e-02	9.60e+01	1.78e-02	1.46e-02	2.40e-01	2.30e+01	0.00e+00
100	9.58e-03	1.52e+02	7.94e-01	2.89e-02	2.18e-01	2.58e+01	0.00e+00

Table 7. The mean values for 100-D problems. (cont.)

P	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
1	2.39e+03	1.02e+00	9.83e-03	2.14e-01	3.21e+01	0.00e+00	1.78e+03
2	2.19e+03	3.43e-01	2.04e-02	2.14e-01	3.04e+01	0.00e+00	2.97e+03
3	2.65e+03	1.03e+00	9.22e-03	3.49e-01	2.90e+01	0.00e+00	1.92e+03
4	2.34e+03	4.46e-01	1.51e-02	1.76e-01	2.60e+01	0.00e+00	2.50e+03
5	2.47e+03	4.87e-01	1.35e-02	1.93e-01	2.93e+01	0.00e+00	2.07e+03
6	2.19e+03	5.14e-01	7.98e-03	1.53e-01	3.04e+01	0.00e+00	2.36e+03
7	2.19e+03	3.02e-01	1.67e-02	2.09e-01	3.02e+01	0.00e+00	2.00e+03
8	2.38e+03	5.64e-01	1.89e-02	3.90e-01	2.64e+01	0.00e+00	2.01e+03
9	2.15e+03	3.04e-01	1.71e-02	2.02e-01	**2.55e+01**	0.00e+00	2.13e+03
10	2.31e+03	8.08e-02	1.70e-02	2.14e-01	3.27e+01	0.00e+00	**1.51e+03**
11	2.89e+03	6.44e-01	9.51e-03	2.43e-01	2.71e+01	0.00e+00	2.09e+03
12	**2.14e+03**	8.80e-01	**5.29e-03**	1.57e-01	2.60e+01	0.00e+00	1.89e+03
13	2.32e+03	3.29e-01	1.47e-02	3.65e-01	2.75e+01	0.00e+00	2.28e+03
14	2.59e+03	**2.00e-02**	1.47e-02	**1.45e-01**	2.59e+01	0.00e+00	1.92e+03
15	2.52e+03	4.84e-01	1.03e-02	1.84e-01	3.12e+01	0.00e+00	2.38e+03
16	2.60e+03	3.32e-01	1.96e-02	1.93e-01	3.04e+01	0.00e+00	2.35e+03
17	2.15e+03	1.03e+00	1.08e-02	1.64e-01	2.90e+01	0.00e+00	2.58e+03
18	2.22e+03	5.92e-01	1.18e-02	2.03e-01	3.26e+01	0.00e+00	2.27e+03
19	2.94e+03	3.74e-01	3.54e-02	1.60e-01	2.86e+01	0.00e+00	2.14e+03
20	2.59e+03	7.05e-01	5.34e-03	2.52e-01	3.12e+01	0.00e+00	2.35e+03
50	2.69e+03	3.96e-01	7.48e-03	4.79e-01	3.41e+01	0.00e+00	2.82e+03
100	2.74e+03	2.25e-01	1.46e-02	1.77e-01	3.11e+01	0.00e+00	2.32e+03

Table 8. The calculation time (ms) for 100-D problems.

P	f_1	f_2	f_3	f_4	f_5	f_6	f_7
1	3865343	3907344	8432688	14837734	107770463	44506938	39000774
2	1934047	1952688	4217828	7418000	54301842	24255033	20953165
3	1288563	1301906	2809469	4943000	36527211	13836234	11972485
4	966735	976625	2108328	3708093	26972501	11635022	9075740
5	773703	782109	1686797	2966766	21603158	9477046	7022733
6	644532	651125	1405250	2471500	17944932	7905807	5853505
7	552656	558188	1204468	2118610	15431906	6772542	5017885
8	484078	488656	1054594	1855094	13514406	5926745	4393292
9	430563	434047	937250	1648172	12009557	5265620	3900917
10	387219	391531	845172	1484891	10813786	4741740	3514864
11	351922	356875	766891	1348719	9823667	4305224	3192875
12	322094	325265	702344	1235453	8995453	3941984	2921792
13	297469	300390	648907	1141359	8314203	3640151	2699677
14	276625	279359	603359	1060469	7723937	3390297	2513115
15	258031	260594	563062	989703	7205234	3158172	2345063
16	241766	244140	527125	926532	6756469	2960333	2193354
17	227922	230109	496844	873141	6363182	2788927	2067406
18	214859	216891	468234	823079	5995797	2628776	1949339
19	203953	206016	444921	781469	5693172	2495177	1849412
20	193954	195843	422985	742984	5414547	2371193	1758130
50	78093	78797	170172	297797	2191109	949484	704271
100	**39468**	**39844**	**85391**	**149562**	**1110818**	**476146**	**352162**

Table 8. The calculation time (ms) for 100-D problems. (cont.)

P	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
1	8416360	8465946	14887180	107772532	49520680	38914469	8426766
2	4212494	4264592	7455683	54305695	25255891	20959078	4219047
3	2811024	2825860	4945657	36529329	14836586	11975688	2811625
4	2111044	2134315	3753972	26980211	11635633	9480899	2111984
5	1694133	1733743	3175391	21495318	9477302	7026943	1693172
6	1411989	1528062	2471500	17938292	7900083	5856531	1408031
7	1211146	1309620	2204610	15424662	6778599	5022761	1209568
8	1059526	1147609	1983295	13512172	5927734	4393734	1058516
9	940396	1018443	1872491	12005406	5267141	3902958	940396
10	847583	918953	1602417	10818677	4742500	3516010	847760
11	770406	833901	1527842	9821073	4306505	3190875	769755
12	704885	763354	1372219	8991464	3942151	2923417	703380
13	651985	706177	1236329	8320052	3640547	2697932	651417
14	605307	656714	1129041	7723333	3387615	2510213	605740
15	568490	613292	1097830	7225792	3161109	2342984	565031
16	530781	572339	1005362	6760963	2959318	2194333	530792
17	499156	539521	923150	6362182	2790052	2068448	499276
18	468990	509729	857309	5997474	2628448	1949177	469271
19	446589	482698	817491	5701068	2494927	1850594	447359
20	424125	459135	779244	5417932	2372333	1759313	423594
50	170109	184245	327097	2188411	949104	704172	170130
100	**85651**	**92641**	**170664**	**1110604**	**476083**	**353761**	**85646**

4. CONCLUSIONS

The purpose of the current study was to analyse the effect of the migration period parameter on PCLPSO algorithm. PCLPSO based on the master-slave paradigm has multiple swarms which work cooperatively and concurrently on distributed computers. Each swarm runs the algorithm independently. In the cooperation, the swarms exchange their own local best particle with each other in every migration process. Thus, the diversity of the solutions increases through the multiple swarms and cooperation. PCLPSO runs on a cluster. We used the well-known benchmark functions in the experiments. In the experiments, the performance of PCLPSO is analysed using different migration periods. This study has shown that the calculation time decreases when the migration period is longer. We obtained better results for low-dimensional problems when the migration period is around 6. We obtained better results for 30-dimensional problems when the migration period is around 13. We obtained better results for high-dimensional problems when the migration period is around 12. The migration period should be tuned for different problems. Namely, it varies with regard to the difficulty of problems. As future work, we plan to investigate the effects of the number of particles to be exchanged between swarms on the performance of the PCLPSO algorithm.

REFERENCES

[1] M. Mezmaz, N. Melab, Y. Kessaci, Y.C. Lee, E.-G. Talbi, A.Y. Zomaya, D. Tuyttens, A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems, Journal of Parallel and Distributed Computing, 71 (2011) 1497-1508.

[2] Z. Guo, J. Mi, P. Grant, An implicit parallel multigrid computing scheme to solve coupled thermal-solute phase-field equations for dendrite evolution, Journal of Computational Physics, 231 (2012) 1781-1796.

[3] J. Pang, A.R. Lebeck, C. Dwyer, Modeling and simulation of a nanoscale optical computing system, Journal of Parallel and Distributed Computing, 74 (2014) 2470-2483.

[4] J. Kennedy, R. Eberhart, Particle swarm optimization, 1995 Ieee International Conference on Neural Networks Proceedings, Vols 1-6, (1995) 1942-1948.

[5] A.R. Yildiz, A new hybrid particle swarm optimization approach for structural design optimization in the automotive industry, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 226 (2012) 1340-1351.

[6] M.S. Kıran, E. Özceylan, M. Gündüz, T. Paksoy, A novel hybrid approach based on particle swarm optimization and ant colony algorithm to forecast energy demand of Turkey, Energy conversion and management, 53 (2012) 75-83.

[7] M. Mutluer, O. Bilgin, Design optimization of PMSM by particle swarm optimization and genetic algorithm, in: Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on, IEEE, 2012, pp. 1-4.

[8] G.E. Güraksın, H. Haklı, H. Uğuz, Support vector machines classification based on particle swarm optimization for bone age determination, Applied Soft Computing, 24 (2014) 597-602.

[9] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, Ieee T Evolut Comput, 10 (2006) 281-295.

[10] Ş. Gülcü, H. Kodaz, A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization, Engineering Applications of Artificial Intelligence, 45 (2015) 33-45.

[11] E. Alba, Parallel metaheuristics: a new class of algorithms, John Wiley & Sons, 2005.

[12] G.-W. Zhang, Z.-H. Zhan, K.-J. Du, Y. Lin, W.-N. Chen, J.-J. Li, J. Zhang, Parallel particle swarm optimization using message passing interface, in: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1, Springer, 2015, pp. 55-64.

[13] M. Pedemonte, S. Nesmachnow, H. Cancela, A survey on parallel ant colony optimization, Applied Soft Computing, 11 (2011) 5181-5197.

[14] B. Li, K. Wada, Communication latency tolerant parallel algorithm for particle swarm optimization, Parallel Computing, 37 (2011) 1-10.

[15] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, IEEE, 1998, pp. 69-73.

[16] F. Van Den Bergh, An analysis of particle swarm optimizers, in, University of Pretoria, 2006.

[17] Ş. Gülcü, H. Kodaz, Effects of the number of swarms on parallel multi-swarm PSO, International Journal of Computing, Communication and Instrumentation Engineering, 3 (2016) 201-204.

[18] F.L. Bellifemine, G. Caire, D. Greenwood, Developing multi-agent systems with JADE, John Wiley & Sons, 2007.

[19] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, Evolutionary Computation, IEEE Transactions on, 3 (1999) 82-102.

[20] B.-Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, Evolutionary Computation, IEEE Transactions on, 17 (2013) 387-402.

[21] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, Evolutionary Computation, IEEE Transactions on, 14 (2010) 150-169.

[22] S.C. Esquivel, C.A. Coello Coello, On the use of particle swarm optimization with multimodal functions, in: Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, IEEE, 2003, pp. 1130-1136.

[23] R. Salomon, Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms, BioSystems, 39 (1996) 263-278.