Behavior Changing Schedules for Heterogeneous Particle Swarms

Filipe V. Nepomuceno
Department of Computer Science
University of Pretoria
Pretoria, South Africa, 0184
Email: filinep@gmail.com

Andries P. Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria, South Africa, 0184
Email: engel@cs.up.ac.za

***Abstract*—Heterogeneous particle swarm optimizers (HPSO) add multiple search behaviors to the swarm. This is done by allowing particles to utilize different update equations to each other. Dynamic and adaptive HPSO algorithms allow the particles to change their behaviors during the search. A number of factors come into play when dealing with the different behaviors, one of which is deciding when a particle should change its behavior. This paper presents a number of behavior changing schedules and strategies for HPSOs. The schedules are compared to each other using existing HPSO algorithms on the CEC 2013 benchmark functions for real-parameter optimization.**

***Keywords*—Heterogeneous, particle swarm optimization**

I. Introduction

Particle swarm optimization is a stochastic optimization method introduced by Kennedy and Eberhart [5], [9]. It works by iteratively moving particles around an n-dimensional search space optimizing a given function. Each position in the search space has an associated fitness. Every particle has a position (candidate solution), a velocity and a personal best, or $pbest$, which is the best position the particle has visited so far. The best position the swarm has visited is called the global best, or $gbest$. If the swarm is separated into localized neighborhoods then each neighborhood has a neighborhood best, or $nbest$. Every iteration the particles update their velocities and position according to

$$\begin{aligned} v_{ij}(t+1) &= wv_{ij}(t) + c_1 r_{1j}(t)(y_{ij}(t) - x_{ij}(t)) \\ &\quad + c_2 r_{2j}(t)(\hat{y}_j(t) - x_{ij}(t)) \qquad (1) \\ x_{ij}(t+1) &= x_{ij}(t) + v_{ij}(t+1) \qquad (2) \end{aligned}$$

where $v_{ij}(t)$ is particle i's velocity at time step t in dimension j, w is an inertia weight to scale the previous velocity, $x_i(t)$ is the particle's position, $y_i(t)$ is the particle's $pbest$, $\hat{y}(t)$ is the $nbest$, c_1 is the cognitive acceleration coefficient used to scale the influence of the $pbest$, c_2 is the social acceleration coefficients used to scale the influence of the $nbest$ and $r_{1j}, r_{2j} \sim U(0,1)$. After the particles have updated their positions, their $pbest$s are updated if they improved and the $gbest$ is recalculated as the best $pbest$ in the swarm.

Particle swarm optimizers where all the particles use the same update equations are considered homogeneous and swarms where the particles can use different update equations are considered heterogeneous [6], [13]. The update equations determine the search behavior that a particle exhibits which in turn affects the swarm's search behavior. The term "behavior" is used to refer to the combination of the velocity and position update equations. Heterogeneous particle swarm optimizers (HPSO) maintain a behavior pool from which the particles are assigned behaviors. Three types of HPSO exist [13]:

Static HPSO Particles are assigned a behavior which is maintained for the duration of the search.

Dynamic HPSO Particles can change their behavior during the search. The next behavior is selected stochastically or deterministically.

Adaptive HPSO Particles select their next behavior based on a scoring method used on the behaviors.

With regards to dynamic and adaptive HPSO a schedule must be set to determine when a particle should change its behavior. A couple of schedules are already in use, e.g. changing a behavior when the $pbest$ of a particle stagnates (it is considered stagnant if it has not improved for a number of iterations). The focus of this paper is to present other behavior changing schedules and strategies for HPSO algorithms. The schedules are used in existing HPSO algorithms and compared using the CEC 2013 benchmark function set [11].

The rest of this paper is organized as follows: Section II describes how HPSO algorithms work and Section III presents the different behavior changing schedules. The experimental setup is given in Section IV and the results are presented and analyzed in Section V. Section VI concludes the paper.

II. Heterogeneous Particle Swarms

This section describes the HPSO model and different HPSO algorithms. Figure 1 shows how a typical HPSO algorithm works. There are a number of differences from a normal PSO. Three extra elements must be taken into account when dealing with HPSO (shown in Figure 1 with their relevant section):

1) Which behavior to select
2) When to change behaviors
3) How to select behaviors

The following three sub-sections address each of these elements.

A. Behavior Pool

The behaviors in the behavior pool determine which search behaviors the swarm and the particles can exhibit. Therefore,

```
initialize swarm
while stopping conditions not met do
    for each particle do
        if change schedule is triggered then {Section II-C}
            select new behavior for particle {Section II-A}
        end if
    end for
    for each particle do
        update particle's velocity and position
    end for
    for each particle do
        update pbest and gbest
    end for
    for each behavior do
        update behavior score {Section II-B}
    end for
end while
```

Figure 1. Pseudo code for an adaptive heterogeneous PSO model

a wide range of behaviors should be added to the pool even if those behaviors do not perform well on their own. The following are some of the behaviors used in HPSO algorithms.

- The fully-informed PSO (FIPS) [12] uses a weighted average of all the particles' $pbest$s as the attractor point instead of the $pbest$ and $gbest$.
- The cognitive only PSO (cPSO) [7] removes the social component, i.e. $c_2 = 0$, and acts as a hill climber.
- The social only PSO (sPSO) [7] removes the cognitive component, i.e. $c_1 = 0$, which enhances its exploitation ability.
- The quantum PSO (QSO) [3] was developed for dynamic environments to exploit the $gbest$ with part of the swarm while the rest of the swarm keeps exploring.
- The barebones PSO (BB-PSO) and modified barebones PSO (modBB-PSO) [8] move the particles randomly using a Gaussian distribution centered on the mean of the $gbest$ and $pbest$ with a deviation of $|y_{ij}(t) - \hat{y}_j(t)|$. The modified version exploits the $pbest$ with an exploitation probability of 0.5.
- The time-varying inertia weight PSO (TVIW-PSO) [17] linearly decreases the inertia weight during the search. This allows for fine tuned searching towards the end of the search.
- The time-varying acceleration coefficient PSO (TVAC-PSO) [16] linearly changes the acceleration coefficients. The cognitive acceleration is linearly decreased for more exploration at the beginning of the search and the social acceleration is linearly increased for more exploitation at the end of the search.
- The canonical PSO (PSO) [9] with inertia weight [17] has also been used in HPSO algorithms.

B. Behavior Changing Strategies

A number of HPSO algorithms have been developed to select a particle's next behavior. Engelbrecht proposed the dynamic HPSO (dHPSO) [6] which simply selects a random behavior from the behavior pool.

Montes de Oca *et al* [13] experimented with a static HPSO using two behaviors where the swarm was initialized with different ratios of the two behaviors. This HPSO uses an "identity" behavior changing strategy, i.e. it uses the same behavior as before.

Spanvello and Montes de Oca [18] developed an adaptive HPSO where a particle copies the $gbest$'s behavior based on a probability which is proportional to the difference of a particle's $pbest$ and the $gbest$, i.e.

$$p_{ij} = \frac{1}{1 + \exp\left(-\beta \frac{f(y_i(t)) - f(y_j(t))}{|f(y_j(t))|}\right)} \tag{3}$$

where p_{ij} is the probability that particle i will adopt particle j's behavior and β is a constant that determines how sensitive particle i is to the difference in fitness between particle j and itself. This PSO is called the difference proportional probability PSO (DPP-PSO).

Nepomuceno and Engelbrecht created two versions of the pheromone-based HPSO (pHPSO) [15] inspired by the foraging behavior of ants. Each behavior is given a pheromone concentration. Each iteration the pheromone is updated, using one of two strategies:

- A constant strategy (pHPSO-const) where the pheromone of a behavior is incremented by one if a particle using that behavior improved, incremented by 0.5 if it remained the same and unchanged if it performed worse than the previous iteration.
- A linear strategy (pHPSO-lin) where the pheromone of a behavior is modified based on the difference of the fitness from the previous iteration when using that particular behavior.

Once the pheromone is updated it is evaporated to maintain behavioral diversity. A particle selects its next behavior using roulette wheel selection.

The adaptive-learning PSO II (ALPSO-II) [10] gives each particle its own behavior pool. Each behavior has a reward value, a progress value and selection ratio. The progress value is calculated based on the improvement of a particle using the behavior. The reward value is then calculated as the sum of the normalized progress value, success rate and previous selection ratio. The selection ratio is the normalized reward value scaled down. Each behavior also maintains a monitoring reward value, monitoring progress value and a monitoring selection ratio which is used to decide when to reset the selection ratio. This is to maintain behavioral diversity.

The frequency-based HPSO (f_k-PSO) [14] is a simple self-adaptive HPSO which gives each behavior a success counter. Each time a particle using the behavior improves, the success counter is incremented. The successes are only considered for the previous k iterations. When a behavior change is triggered for a particle, the next behavior is selected using tournament selection based on the success counter.

C. Behavior Changing Schedule

A particle should change its behavior if its current behavior is not contributing to the search anymore. Since this is the focus of this paper, the next section goes into more detail. Some of the strategies already used are to trigger a behavior change every iteration, e.g. ALPSO-II and DPP-PSO, and to trigger a behavior change when the particle's $pbest$ stagnates, i.e. when the $pbest$ does not improve for a number of iterations, e.g. dHPSO, pHPSO and f_k-PSO.

III. BEHAVIOR CHANGING SCHEDULES

This section describes different behavior changing schedules and strategies for HPSO.

A. Schedules

Behavior changing schedules determine when a particle should change its behavior.

Personal Best Stagnation The personal best stagnation schedule triggers a behavior change when the fitness of a particle's $pbest$ does not change for a number of iterations. Using a high value for the stagnation threshold allows particles to move out of bad regions before any improvement is observed. Lower stagnation thresholds allows particles to change behaviors faster which could discard bad behaviors quicker.

Periodic Changing behaviors periodically, e.g. every m iterations, ensures that behaviors will get more chances to be selected which should improve behavioral diversity. Shorter periods would prevent bad behaviors from being used too much while longer periods would benefit the better behaviors. Considering adaptive HPSO algorithms a behavior is considered "better" if it has a higher score, e.g. pheromone concentration for the pHPSO, success score for f_k-PSO, etc.

Random Randomly changing behaviors according to a certain probability would provide similar benefits to the periodic schedule but at an irregular interval. Lower probabilities are similar to longer periods and higher probabilities are similar to shorter periods.

Fitness Stagnation Similar to $pbest$ stagnation, fitness stagnation will trigger a behavior change if the fitness of a particle's current position does not improve for m iterations. The implications of the stagnation threshold are the same as the $pbest$ stagnation threshold.

B. Strategies

Behavior changing strategies are used together with the schedules to modify the particles before they change their behaviors.

Velocity Reset The velocity calculated using one behavior might not be ideal for another behavior, e.g. a high velocity is not good for the cPSO's hill climbing behavior. When a behavior change is triggered the velocity is also reinitialized randomly.

Personal Best Reset Another strategy is to move a particle to its $pbest$ when a behavior change is triggered to allow the new behavior to start searching in a good location.

IV. EXPERIMENTAL SETUP

This section describes the experimental setup used to perform the experiments.

A. Setup

Four algorithms were chosen to compare the different schedules and strategies to each other: frequency-based HPSO, dynamic HPSO and the two pheromone-based HPSO. Each schedule ($pbest$ stagnation, fitness stagnation, random and periodic) was combined with the following behavior changing strategies: no resets, resetting the $pbest$, resetting the velocity and resetting both the $pbest$ and the velocity. This gives 16 different behavior changing schedules which will be distinguished using the notation *x/y* where *x* is the behavior changing schedule (pb-stag, fit-stag, random, periodic) and *y* is behavior changing strategy (none, $pbest$, velocity or velocity-$pbest$).

The algorithms were run on the CEC 2013 benchmark set [11] in 10, 30 and 50 dimensions for $10000 \times D$ function evaluations where D is the dimension of the problem. As per the benchmark definition, 51 independent runs were used for each function. All algorithms and problems were implemented using CIlib (http://www.cilib.net).

The Friedman test with the Bonferroni-Dunn post-hoc test was used for statistical analysis as described by Demšar [4]. A 95% confidence interval was used.

B. Parameters

Heterogeneous PSO have two types of parameters: behavior parameters and algorithm parameters. The behavior parameters are the parameters used by the behaviors in the behavior pool. Since the behaviors are chosen for the specific search capabilities they exhibit, changing their parameters would alter those search capabilities. As a result the behavior parameters are not tuned. The values used for the behavior parameters are the same as those used by the authors of the respective algorithms.

The algorithm parameters are those used by the algorithm that controls the behavior pool. The algorithm parameters were tuned using iterated F-Race [1], [2]. F-Race is a racing algorithm used for automatic parameter tuning. F-Race runs an algorithm using a number of different configurations on a set of problems and discards the configurations which perform statistically worse than the best configuration. The statistical test is done using the Friedman test with the Nemenyi post-hoc test.

Iterated F-Race runs the F-Race algorithm a number of times. When the F-Race algorithm completes, new parameter configurations are sampled using the surviving configurations from the previous iteration as models for the new configurations. When tuning parameters it is recommended to use functions from the same problem class as the functions on which the algorithms will be run [2], therefore, the CEC 2005 benchmark functions were selected for the tuning problems since they have similar characteristics to the CEC 2013 benchmark set.

The parameters used by the behavior changing schedules are the $pbest$ stagnation threshold, fitness stagnation threshold, behavior changing probability for the random schedule and the period for the periodic schedule. The values for the schedule parameters obtained via tuning are shown in Table I.

Table I. SCHEDULE PARAMETERS

Reset Strategy	Algorithm	Schedules			
		Stagnation		Periodic	Random
		PBest	Fitness		
None	dHPSO	19	33	29	0.44
	f_k-PSO	32	2	7	0.08
	pHPSO-const	29	21	7	1.0
	pHPSO-lin	2	6	20	0.12
Velocity only	dHPSO	5	8	7	0.16
	f_k-PSO	3	4	41	0.22
	pHPSO-const	31	40	37	0.32
	pHPSO-lin	14	14	19	0.05
PBest only	dHPSO	22	8	9	0.22
	f_k-PSO	45	17	7	0.68
	pHPSO-const	25	25	31	0.07
	pHPSO-lin	10	44	18	0.25
Velocity and PBest	dHPSO	44	5	5	0.04
	f_k-PSO	36	10	29	0.49
	pHPSO-const	30	9	13	0.07
	pHPSO-lin	24	4	23	0.12

The values used for the rest of the parameters are shown in Table II for the f_k-PSO, Table III for the pHPSO-const and Table IV for the pHPSO-lin. The dHPSO uses only the schedule parameters.

Table II. ALGORITHM PARAMETERS FOR f_k-PSO

Reset Strategy	Parameter	Schedules			
		Stagnation		Periodic	Random
		PBest	Fitness		
None	Tournament size	2	2	2	2
	k	248	21	19	29
Velocity only	Tournament size	2	3	3	5
	k	19	193	279	441
PBestonly	Tournament size	4	4	4	6
	k	30	66	333	50
Velocity and PBest	Tournament size	2	2	4	5
	k	29	10	289	123

Table III. ALGORITHM PARAMETERS FOR PHPSO-CONST

Reset Strategy	Parameter	Schedules			
		Stagnation		Periodic	Random
		PBest	Fitness		
None	Better score	1.49	1.06	2.09	1.08
	Same score	-0.45	-0.86	-0.36	-0.39
	Worse score	-0.17	-0.06	-0.02	-0.08
	Min pheromone	0.04	0.08	0.07	0.04
Velocity only	Better score	0.76	2.42	0.90	1.17
	Same score	-0.43	-0.21	-0.89	0.51
	Worse score	-0.30	-0.10	1.53	0.18
	Min pheromone	-0.06	-0.04	0.07	0.07
PBestonly	Better score	1.24	0.79	1.53	0.30
	Same score	0.99	0.31	-0.45	-0.05
	Worse score	-0.02	-1.15	-0.25	-0.80
	Min pheromone	0.06	0.05	0.05	0.01
Velocity and PBest	Better score	1.97	1.33	2.78	0.34
	Same score	-0.17	-0.08	0.36	-1.1
	Worse score	-0.16	1.05	-0.33	-0.42
	Min pheromone	0.04	0.02	0.02	0.01

V. RESULTS AND ANALYSIS

This section analyzes the results obtained for the experiments.

Behavior profiles show the average number of particles using the different behaviors during the search process. Figure 2 shows the behavior profile plot for the f_k-PSO on function f_{12} in 30 dimensions. The figure shows that using different schedules does affect the behavior selection process. The variants that either used no reset mechanism or that used a stagnating schedule maintained a better behavioral diversity than the other variants. The variants that used velocity resetting usually had two or three dominant behaviors.

Table IV. ALGORITHM PARAMETERS FOR PHPSO-LIN

Reset Strategy	Parameter	Schedules			
		Stagnation		Periodic	Random
		PBest	Fitness		
None	Gradient	0.07	0.36	1.04	0.19
	Min pheromone	0.003	0.001	0.01	0.01
Velocity only	Gradient	1.19	0.14	0.42	0.18
	Min pheromone	0.05	0.01	0.02	0.001
PBestonly	Gradient	0.44	0.38	0.32	1.65
	Min pheromone	0.01	0.02	0.03	0.03
Velocity and PBest	Gradient	0.51	0.01	1.45	1.02
	Min pheromone	0.01	0.004	0.04	0.06

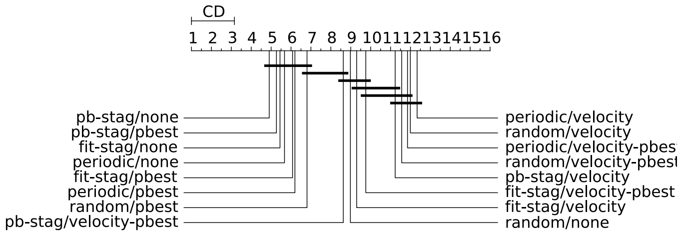

Figure 4. Critical difference diagram for dHPSO

Figure 3 shows the convergence plots for f_{12} in 30 dimensions. The f_k-PSO has been noted to converge later in the search and, therefore, contains a dip near the middle of the search [14]. Figure 3 shows that using different behavior changing schedules changes where the convergence occurs, e.g. velocity resetting variants cause the convergence to occur later in the search and the $pbest$ resetting variants tend to smooth out the dip over the whole search process.

Critical difference diagrams show the average ranks of the variants with the results of the statistical test shown with bold lines. The variants that are linked with a bold line indicate that there is no statistical difference between those algorithms. Figures 4, 5, 6 and 7 show the critical difference diagrams for the dHPSO, f_k-PSO, pHPSO-const and pHPSO-lin respectively. The results show that many of the schedules have very similar performance. Also, different algorithms benefit more from different behavior changing schedule/strategy combinations, e.g. dHPSO obtained the best performance using pb-stag/none, f_k-PSO obtained the best performance with periodic/velocity and the pheromone based algorithms obtained the best performance using pb-stag/$pbest$. Generally, the velocity resetting variants performed the worst for all the algorithms except for the f_k-PSO. This may be because once a particle resets its velocity its current behavior cannot gather enough momentum with successes before the behavior is changed.

Table V summarizes the average ranks taking all the algorithms, functions and dimensions into account. The average ranks is obtained by ranking each algorithm for each function-dimension combination and averaging the resultant ranks. Numbers in bold indicate the highest ranks. Figure 8 shows the critical difference diagram for the results in

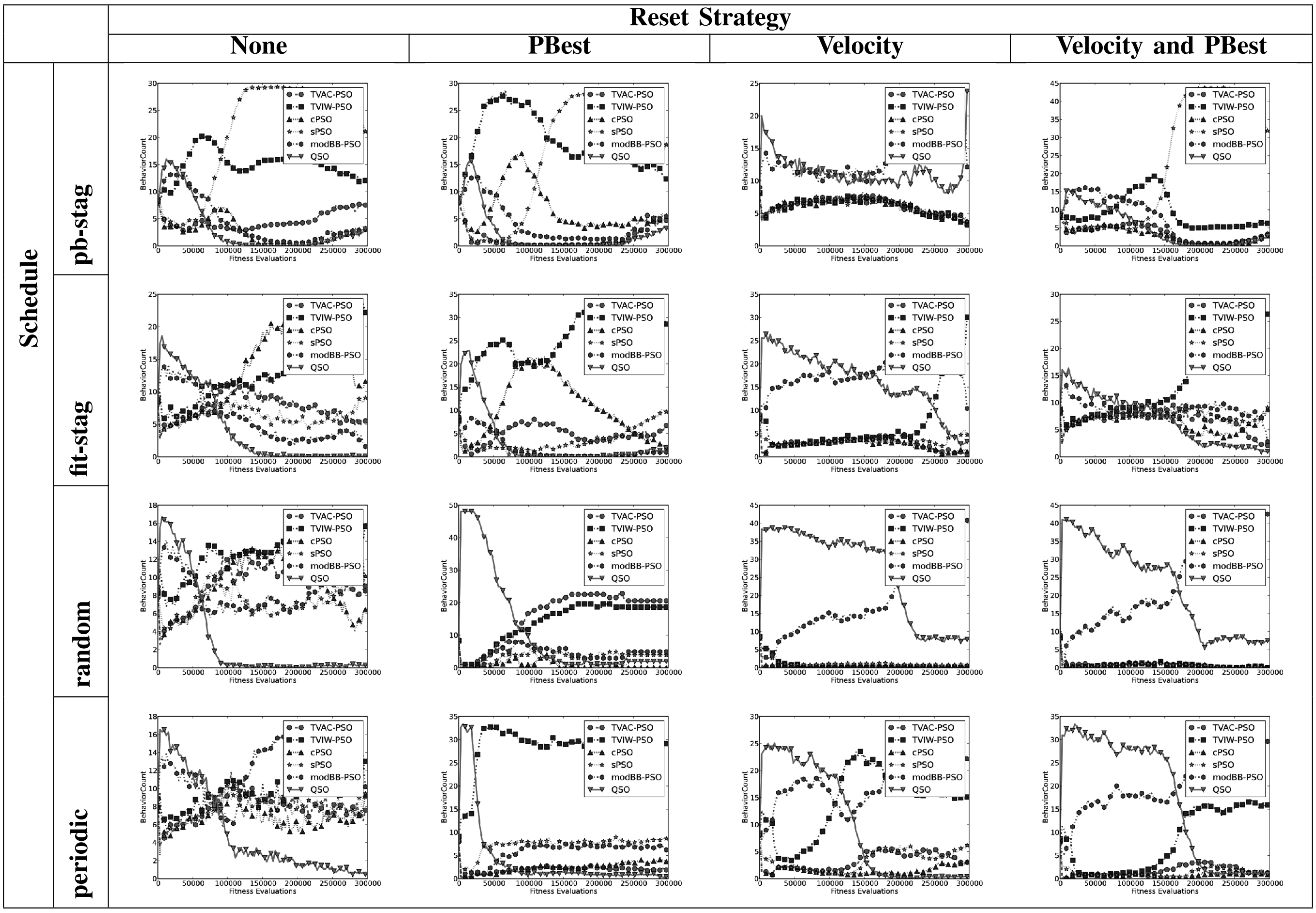

Figure 2. Behavior profile plots for f_k-PSO for f_{12} in 30 dimensions

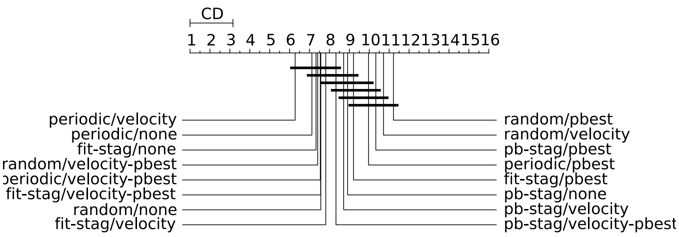

Figure 5. Critical difference diagram for f_k-PSO

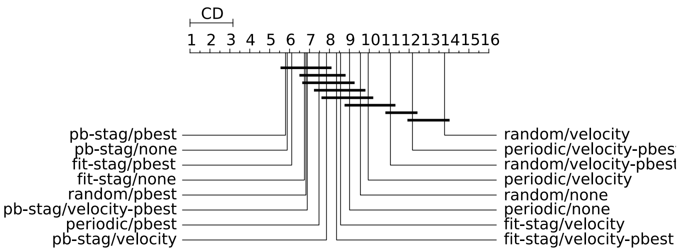

Figure 6. Critical difference diagram for pHPSO-const

Table V. Different variants obtained the highest ranks for different types of functions. For the unimodal functions the pb-stag/none variant performed the best. For the multimodal functions the fit-stag/none variant performed the best and for

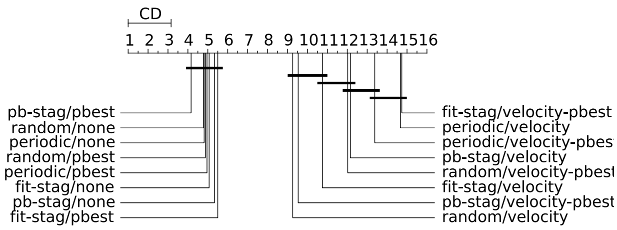

Figure 7. Critical difference diagram for pHPSO-lin

and the composite functions the periodic/none variant obtained the highest rank. Over all the functions the fit-stag/none variant performed the best although there is no statistical difference to the pb-stag/none, pb-stag/$pbest$, periodic/none, fit-stag/$pbest$ and periodic/$pbest$ variants. Again note that the velocity resetting variants performed the worst.

VI. CONCLUSIONS

Dynamic heterogeneous PSOs allow the particles to change their behaviors during the search. One of the factors that affect these HPSO algorithms is deciding when the particles should change behaviors. Four different schedules ($pbest$ stagnation, fitness stagnation, periodically and randomly) and two strategies (resetting the velocity and resetting the $pbest$) were applied to existing HPSO algorithms and run on the CEC 2013

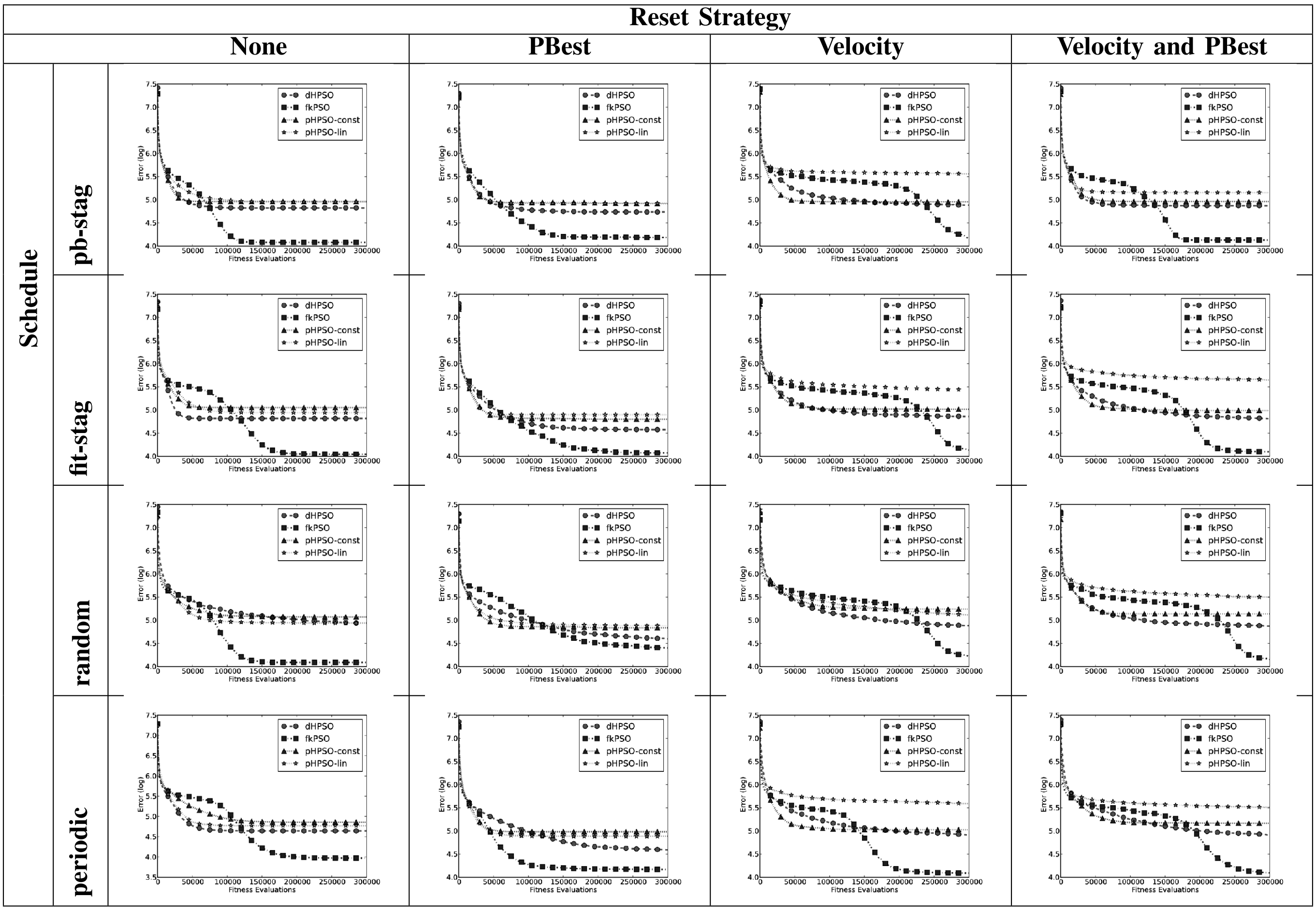

Figure 3. Fitness plots for f_{12} in 30 dimensions

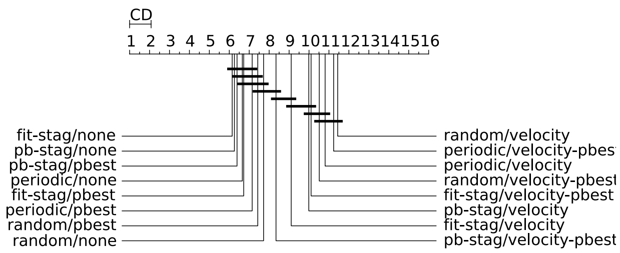

Figure 8. Critical difference diagram for all the algorithms

benchmark function set. The results show that the different schedules do affect the behavior selection abilities of the HPSO algorithms. The results also indicate that resetting the velocity generally results in worse performance and that the algorithms are affected differently by the different schedules.

Future research that can be conducted includes analyzing the behavior pools of the HPSOs and comparing the HPSO algorithms to other metaheuristic algorithms.

REFERENCES

[1] P. Balaprakash, M. Birattari, and T. Stützle, "Improvement strategies for the f-race algorithm: sampling design and iterative refinement," in *Proceedings of the 4th international conference on Hybrid metaheuristics*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 108–122.

[2] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, "A racing algorithm for configuring metaheuristics," in *Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 11–18.

[3] T. Blackwell and J. Branke, "Multi-swarm optimization in dynamic environments," in *EvoWorkshops*, 2004, pp. 489–500.

[4] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[5] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[6] A. Engelbrecht, "Heterogeneous particle swarm optimization," in *Proceedings of the 7th international conference on Swarm intelligence*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 191–202.

[7] J. Kennedy, "The particle swarm: social adaptation of knowledge," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1997, pp. 303–308.

[8] ——, "Bare bones particle swarms," in *Proceedings of the IEEE Swarm Intelligence Symposium*, 2003, pp. 80–87.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[10] C. Li and S. Yang, "Adaptive learning particle swarm optimizer-II for global optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[11] J. Liang, B.-Y. Qu, P. Suganthan, and A. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization," Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and

Table V. AVERAGE RANKS FOR THE BEHAVIOR CHANGING SCHEDULES OVER ALL FUNCTIONS, DIMENSIONS AND ALGORITHMS

	Function	fit-stag/none	fit-stag/pbest	fit-stag/velocity	fit-stag/velocity-pbest	pb-stag/none	pb-stag/pbest	pb-stag/velocity	pb-stag/velocity-pbest	periodic/none	periodic/pbest	periodic/velocity	periodic/velocity-pbest	random/none	random/pbest	random/velocity	random/velocity-pbest
Unimodal	**f1**	10.67	8.92	8.29	9.38	**4.50**	6.12	7.54	7.50	10.71	7.33	11.12	11.21	4.67	8.17	10.21	9.67
	f2	**4.33**	8.33	8.08	8.67	5.33	6.42	10.25	8.42	6.67	6.92	10.33	12.00	9.42	7.42	14.42	9.00
	f3	**4.08**	5.42	10.75	11.50	4.92	5.58	9.83	7.17	6.83	5.92	12.17	13.83	7.50	6.17	13.00	11.33
	f4	**4.58**	8.17	8.17	8.42	5.67	6.75	10.58	8.00	6.25	7.25	11.08	11.75	9.75	6.33	14.17	9.08
	f5	10.17	8.58	8.00	10.25	**3.12**	4.67	7.38	5.46	10.88	9.00	10.04	10.21	6.75	10.75	11.12	9.62
	Mean	6.77	7.88	8.66	9.64	**4.71**	5.91	9.12	7.31	8.27	7.28	10.95	11.80	7.62	7.77	12.58	9.74
	Stddev	3.34	1.41	1.17	1.26	**0.99**	0.81	1.54	1.14	2.32	1.11	0.83	1.33	2.08	1.86	1.86	0.93
Multimodal	**f6**	5.17	5.33	12.33	12.42	**2.75**	2.92	8.25	3.75	10.25	6.75	13.42	13.08	6.67	9.00	12.08	11.83
	f7	6.92	6.42	8.67	9.17	5.67	5.83	10.58	10.17	**4.50**	5.83	13.17	12.67	6.17	6.25	12.67	11.33
	f8	6.75	**3.50**	11.25	9.17	5.33	5.50	11.75	10.42	6.75	5.42	12.08	14.25	5.00	4.42	13.00	11.42
	f9	**6.00**	8.00	7.17	8.42	7.50	7.33	9.67	8.67	7.08	9.50	10.25	8.92	8.92	8.75	10.33	9.50
	f10	5.17	4.42	13.17	10.75	6.50	6.67	13.00	8.92	7.50	**3.00**	10.50	11.25	5.50	6.17	11.00	12.50
	f11	**3.58**	6.92	9.92	11.50	6.50	5.92	8.17	8.33	5.83	7.42	12.17	11.58	9.83	6.50	11.17	10.67
	f12	6.25	4.33	10.33	9.50	6.50	5.33	11.92	9.08	**3.58**	6.17	11.75	11.75	7.17	6.83	13.17	12.33
	f13	4.67	5.17	10.08	11.75	6.75	**4.50**	12.83	10.17	5.08	5.17	12.00	11.25	5.42	7.00	12.75	11.42
	f14	**5.67**	5.92	8.50	10.33	6.58	5.83	7.83	8.92	7.08	8.42	11.67	10.17	10.33	6.75	11.00	11.00
	f15	9.50	7.75	7.42	7.00	8.75	8.92	9.17	9.50	8.00	**6.83**	9.25	8.58	8.42	8.00	9.83	9.08
	f16	3.83	**3.17**	9.75	10.42	7.08	5.25	11.08	10.58	5.42	7.08	10.08	12.33	7.83	7.92	13.17	11.00
	f17	**5.17**	9.83	9.33	10.50	5.75	7.17	7.92	5.83	7.42	9.75	11.50	11.17	7.58	7.50	11.67	7.92
	f18	**2.50**	4.42	10.67	11.00	5.67	5.00	12.33	8.25	3.08	5.67	11.58	13.75	7.92	7.33	13.58	13.25
	f19	**4.00**	4.33	9.67	10.42	5.92	5.33	11.42	7.42	4.92	5.92	10.33	14.00	9.50	8.58	13.25	11.00
	f20	**4.08**	5.42	11.00	12.17	5.08	5.75	12.67	8.25	4.17	5.42	11.42	11.17	7.75	6.58	13.50	11.58
	Mean	**5.28**	5.66	9.95	10.30	6.16	5.82	10.57	8.55	6.04	6.56	11.41	11.73	7.60	7.17	12.14	11.06
	Stddev	**1.70**	1.84	1.64	1.46	1.32	1.37	1.91	1.82	1.92	1.74	1.15	1.69	1.64	1.19	1.23	1.36
Composite	**f21**	9.08	11.33	6.83	9.00	7.67	7.75	6.92	**6.58**	9.83	7.75	8.50	10.08	9.17	6.83	7.83	10.83
	f22	5.58	7.17	9.08	10.08	6.50	**4.92**	7.33	7.33	9.00	9.33	11.08	9.42	11.33	9.58	9.25	9.00
	f23	7.67	6.17	9.58	10.67	8.83	10.08	11.58	10.50	7.17	**4.58**	9.75	10.00	5.67	6.00	8.83	8.92
	f24	7.92	6.92	6.42	11.58	7.75	7.17	8.42	8.92	**6.33**	9.25	9.75	10.67	8.75	6.33	10.58	9.25
	f25	6.75	8.00	6.83	9.50	6.08	8.92	10.00	8.17	**5.92**	9.00	11.17	8.75	7.92	8.75	11.08	9.17
	f26	7.58	7.67	8.58	10.42	5.67	7.50	10.75	9.75	**4.67**	7.75	9.42	9.50	7.25	8.42	9.33	11.75
	f27	6.50	7.00	6.25	10.17	9.58	6.83	10.33	9.42	**5.25**	8.42	8.83	11.33	8.58	8.50	8.92	10.08
	f28	7.92	9.67	8.83	8.75	7.25	8.92	10.17	8.17	6.17	9.33	8.17	9.92	**5.42**	7.17	9.33	10.83
	Mean	7.38	7.99	7.80	10.02	7.12	7.76	9.44	8.61	**6.79**	8.18	9.58	9.96	8.01	7.70	9.39	9.98
	Stddev	1.07	1.70	1.35	0.92	1.34	1.58	1.68	1.30	**1.79**	1.60	1.11	0.79	1.93	1.29	1.02	1.06
Mean		**6.15**	6.72	9.11	10.10	6.26	6.39	9.99	8.34	6.65	7.15	10.81	11.24	7.72	7.43	11.44	10.51
Stddev		**4.14**	4.33	3.76	4.50	3.94	4.37	4.09	3.80	4.30	4.21	4.49	4.11	4.38	4.68	4.29	3.87

Technical Report, Nanyang Technological University, Singapore, Tech. Rep., 2013.

[12] R. Mendes, J. Kennedy, and J. Neves, "The Fully Informed Particle Swarm: Simpler, Maybe Better," *IEEE Transactions on Evolutionary Computation*, pp. 204–210, 2004.

[13] M. Montes de Oca, J. Pena, T. Stutzle, C. Pinciroli, and M. Dorigo, "Heterogeneous particle swarm optimizers," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2009, pp. 698–705.

[14] F. Nepomuceno and A. Engelbrecht, "A self-adaptive heterogeneous pso for real-parameter optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 2013.

[15] ——, "A self-adaptive heterogeneous pso inspired by ants," in *Proceedings of the 8th international conference on Swarm Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 188–195.

[16] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.

[17] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, 1998, pp. 69–73.

[18] P. Spanevello and M. Montes de Oca, "Experiments on adaptive heterogeneous PSO algorithms," in *Proceedings of Doctoral Symposium on Engineering Stochastic Local Search Algorithms*, 2009, pp. 36–40.