Social Interaction in Particle Swarm Optimization, the Ranked FIPS, and Adaptive Multi-Swarms

Johannes Jordan
jordan@lanrules.de

Sabine Helwig
helwig@cs.fau.de

Rolf Wanka
rwanka@cs.fau.de

Department of Computer Science, University of Erlangen-Nuremberg, Germany

ABSTRACT

The interaction among particles is a vital aspect of Particle Swarm Optimization. As such, it has a strong influence on the swarm's success. In this study various approaches regarding the particles' communication behavior and their relationship are examined, as well as possibilities to combine the approaches. A new variant of the popular FIPS algorithm, the so-called Ranked FIPS, is introduced, which resolves specific shortcomings of the traditional FIPS. As all tested PSO variants feature distinct strengths and weaknesses, a new adaptive strategy is proposed which operates on dissimiliarly configured subswarms. The exchange between these subswarms is solely based on particle migration. The combination of the Ranked FIPS and other strategies within the so called Particle Swarm Optimizer with Migration achieves a very good, yet remarkably reliable performance over a wide range of recognized benchmark problems.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Algorithms

Keywords

Particle Swarm Optimization, Social Interaction, Adaptive Optimization, Subswarms, Performance Analysis

1. INTRODUCTION

Particle Swarm Optimization (PSO) [1, 2] is inspired by the social interaction of individuals living together in groups. In the resulting iteration-based optimization algorithm, each individual i, hereafter called *particle*, is characterized by a *position* $\vec{x}_{i,t}$ (where t is the iteration counter), a *velocity* $\vec{v}_{i,t}$ (at which it moves through the search space), and a *fitness value* $f(x_{i,t})$ (where f is the objective function of the optimization problem). The search space position with the best fitness value particle i has visited so far is its *private guide* $\vec{p}_{i,t}$. A subset of all particles is assigned to each particle as its neighborhood. In the standard approach, the best private guide of a particle's neighbors serves as the particle's *local guide* $\vec{l}_{i,t}$. In each iteration all particle positions and velocitys are updated according to the following equations:

$$\begin{aligned}\vec{v}_{i,t} &= \omega \cdot \vec{v}_{i,t-1} + \vec{U}[0, c_1] \otimes (\vec{p}_{i,t-1} - \vec{x}_{i,t-1}) \\ &\quad + \vec{U}[0, c_2] \otimes (\vec{l}_{i,t-1} - \vec{x}_{i,t-1}) \\ \vec{x}_{i,t} &= \vec{x}_{i,t-1} + \vec{v}_{i,t}\end{aligned}$$

where ω, c_1, and c_2 are predefined parameters, $\vec{U}[m, n]$ is a vector of random real numbers uniformly distributed over $[m, n]$, and $\otimes$ denotes element-by-element vector multiplication. Afterwards, particle i updates its private guide.

Particle interaction involves two aspects: First, which particles are defined as a particle's neighbors, and second, how to process the received information. In this paper, different approaches such as stereotyping [3], FIPS [4, 5], and dynamic neighborhood graphs [6, 7] are studied experimentally, also in combination with each other. A novel FIPS variant, the so-called *Ranked FIPS*, is introduced, which outperforms previous strategies on both the *CEC 2005 benchmark suite* and a traditional one. However, each neighborhood configuration has its specific strengths and weaknesses. Therefore, a new particle swarm optimizer which adapts its neighborhood structure to the optimization process by using subswarms and migration is proposed. This variant is able to provide very good, yet more reliable results than other particle swarm optimizers tested.

2. RELATED WORK

As this paper is concentrated on particle interaction, we give a brief overview of three distinct enhancements to the main PSO algorithm dealing with how particles are related and how they communicate with each other.

2.1 The Fully Informed Particle Swarm (FIPS)

The *Fully Informed Particle Swarm* (FIPS) was first described in 2003 by Rui Mendes et al. [4, 8]. It is based on the assumption that individuals do not merely follow their direct archetype, but are influenced by their whole neighborhood in a specific way. A *Fully Informed Particle* therefore has knowledge about the best solutions found so far by any neighbor, not just the most successful one.

In its original form, the FIPS velocity equation completely replaces the PSO one, discarding private guide and local guide. For greater flexibility, in our work it is employed as a replacement to the local guide alone though. In order to

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07 ...$5.00.

reflect the behavior from literature, c_1 can be set to 0. On the other hand, it is still possible to have a private guide, which enables a combination of FIPS with the stereotyping algorithm presented in section 2.2.

The velocity calculation is done according to [5], but enhanced with the private guide, while the position is updated in correspondance to the canonical PSO:

$$\begin{aligned}\vec{v}_{i,t} &= \omega \cdot \vec{v}_{i,t-1} + \vec{U}[0, c_1] \otimes (\vec{p}_{i,t-1} - \vec{x}_{i,t-1}) \\ &\quad + \textstyle\sum_{j=1}^{N} \vec{U}[0, c_2] \otimes (\vec{p}_{j,t-1} - \vec{x}_{i,t-1})\end{aligned}$$

where N denotes the size of the particle's neighborhood and $\vec{p}_{j,t-1}$ denotes the private guide of neighbor particle j.

This version is called *unweighted*, as all solutions propagated by the neighbors are considered in equal measure. Hence, particles are guided to popular solutions, not particularly good ones. A reasonable alteration is to weight every private guide from the neighborhood by its fitness value:

$$\begin{aligned}\vec{v}_{i,t} &= \omega \cdot \vec{v}_{i,t-1} + \vec{U}[0, c_1] \otimes (\vec{p}_{i,t-1} - \vec{x}_{i,t-1}) \\ &\quad + \frac{\sum_{j=1}^{N} f(\vec{p}_{j,t-1})^{-1} \vec{U}[0, c_2] \otimes (\vec{p}_{j,t-1} - \vec{x}_{i,t-1})}{\sum_{j=1}^{N} f(\vec{p}_{j,t-1})^{-1}}\end{aligned}$$

In this form, the weighted FIPS is suitable for solving minimization problems with $f(x) \geq 0$. To enable benchmarks with negative objective values, in the above equation $f(\vec{p}_{j,t-1})^{-1}$ is replaced by $-f(\vec{p}_{j,t-1})$ if $f(\vec{p}_{j,t-1}) < 0$ as a simple workaround.

2.2 Particle Swarms with Stereotyping

In 2002 Kennedy introduced an algorithm which establishes a relationship between particles that are located close to each other in the search space [3]. The swarm is partitioned into several equivalence classes, called *clusters*, according to the *private guides*. The knowledge of the whole cluster is then used for the calculation of $\overrightarrow{v}$.

As PSO was designed with the social interaction and mutual influence of human individuals being in mind, stereotyping takes one step further in the used analogy. Individuals perceive others not solely on the basis of their individual features, but also based on stereotypes found for specific groups they are members of. Additionally, members themselves try to comply with the values and behaviors formulated by social groups they belong to. Such a group does not solely rely on the experiences of individual members; it is possible for it to strike completely new paths. The membership is dependent on the individual itself though. The notion is that individuals with 'similiar' interests tend to form a group. In the domain of PSO this means that particles searching in the same proximity, and therefore most likely near the same local optimum, form a cluster.

Particles belonging to the same cluster need not be neighbors in the swarm's topology. Instead, a distinct relationship is established. Clusters are formed using the search space position of the private guides. A quantization algorithm like k-means (as suggested by Kennedy) can be used to partition them efficiently. The particles are assigned to k clusters such that each cluster will consist of at least one member.

After determining the k clusters and their corresponding epicenters in the search space, it is possible to use them for the velocity calculation in several ways. The epicenter can either replace the private guide, the local guide, or both. Kennedy's research led to the conclusion that only the first method yields a significant performance increase. Replacing the local guide alone resulted in performance degradation. For our experiments we therefore implemented this variant. Since it does not affect the local guide at all, it is possible to use the FIPS algorithm in combination. Kennedy used a cluster count $k = 5$ on a swarm with 20 particles. We investigated different cluster counts and can conclude that 5 is also a good value for the extended swarm size of 50.

2.3 The Completing Graph

For many optimization problems it is advantageous to explore the search space first and only then improve good solutions at the end of the optimization process. Hence, the capability to (smoothly) switch from *exploration* to *exploitation* is a desired algorithm property. The exploration behavior of a particle swarm can, e.g., be controlled via its neighborhood topology: while densely connected swarms show fast convergence and high exploitation potential, sparsely connected swarms thoroughly explore the search space.

Accordingly, Suganthan's [6], and Richards's and Ventura's [7] particle swarms start with a sparse neighborhood topology, and each particle gradually increases the number of its neighbors until the swarm is fully connected. With Richards's and Ventura's approach, each particle p_i initially has only a single neighbor: p_{i+1} (p_0 is the neighbor of p_{N-1}). This is equivalent to a directed *Ring* topology. Then, after a certain number of iterations K, each p_i adds p_{i+2} to its neighborhood, after $2K$ iterations p_{i+3} is added, and so on, until all particles are connected. K is chosen such that the swarm is fully connected after 4/5-ths of the desired function evaluations. If the number of iterations is not known beforehand (as is the case when particles are allowed to enter infeasible space and the evaluation step is skipped), K has to be estimated.

In every time step the swarm is connected via a *Ring*-like topology. In order to combine the completing graph scheme with the successful *von Neumann* topology, the following approach was used in our experiments: Initially, the swarm is connected via an arbitrary desired topology. The goal is to complete the neighborhood graph to a fully connected one until the K_2-th iteration, where K_2 is a user-specified parameter. Therefore, every $K_2/missing\_edges$ iterations, a random edge is inserted between two nodes of the lowest degree. Note that the graph's diameter may rapidly decrease, contrary to Richards's and Ventura's approach.

3. RANKED FIPS

In this section, a novel variant of the FIPS algorithm is presented. First we briefly discuss some shortcomings affiliated with the weighted and unweighted FIPS.

3.1 Motivation

When the FIPS algorithm was introduced by Rui Mendes et al. in 2003 as PSON and PSOWN, respectively, it was tested with six traditional benchmark functions, and the results were very promising, showing major performance improvements over the canonical PSO algorithm [8]. However, particles where initialized uniformly over the whole search space, and the global optima of all used benchmark functions are located in the center of the search space. Further work, e.g., the introduction of FIPS in 2004, revealed that the algorithm is indeed biased towards the center of its initialization in search space. It was found that FIPS was still able to often outperform the regular PSO and often reach fitness criterions in less computation time when using asymmetrical initialization. However, in many cases it also suffered from a very bad performance, the used neighborhood topology being a key factor [4]. Further investigations revealed three major drawbacks of the FIPS algorithm:

- On some problems, e.g. Ackley and Michalewicz, the optimization process is not able to significantly improve the initial particle positions, whereas other PSO variants are more successful.

- The neighborhood topology, especially the average vertex degree and whether a particle is included in its own neighborhood or not, has a strong effect on the velocity of the particles. Using an unsuitable neighborhood configuration results either in premature convergence or in no convergence at all, if particles are allowed to leave the search space. Experimental analyses of Mendes et al. show that the FIPS algorithm is very sensible to its neighborhood settings [4]. They presented guidelines for choosing a neighborhood configuration which restricts FIPS to a very limited subset of appropriate neighborhood topologies.

- While it was found that the weighted FIPS doesn't work significantly better than the unweighted FIPS, the reasons remained unclear. Still, information about the quality of solutions found should intuitively help the optimizer, if utilized right. It seems the weighted FIPS fails to incorporate this information well.
  Examining the optimization runs of several benchmark functions, we compared the resulting weights of the propagated solutions after applying the random operator $\vec{U}$. It was found that after vast improvements were made in the first iterations, in the majority of the optimization time the private guides of particles close to each other in the topology have very similiar fitness values, leading to a strongly diminished effect of the weighting compared to the randomization. The weighting effect renders negligible, explaining the similiar overall behavior in regard to the unweighted FIPS.

3.2 Algorithm

In an attempt to resolve these shortcomings (especially the one mentioned last), in our proposed algorithm, the propagated solutions are not weighted directly by their fitness value. Instead, the fitness values are used to define a ranking, where a lower fitness value corresponds to a lower rank, respectively. The weight of a solution is solely defined by its rank. Therefore, the distance between a pair of fitness values is rendered irrelevant, whereas information about a solution's relative quality is still preserved. There are several methods of defining the static rank weights. In our work, we used the following rule: The weight of a solution with rank r is always twice as high as the weight of the solution with the subsequent rank $r+1$, while the sum over all weights is 1. Depending on the number of neighbors present, this leads to rank weights $r_1 \in [\frac{1}{2}, \frac{2}{3}]$, $r_2 \in [\frac{1}{4}, \frac{1}{3}]$, and so on. Using these rank weights, we formulate the following equation to calculate $v_{i,t}$:

$$\begin{aligned}\vec{v}_{i,t} &= \omega \cdot \vec{v}_{i,t-1} + \vec{U}[0,c_1] \otimes (\vec{p}_{i,t-1} - \vec{x}_{i,t-1}) \\ &\quad + \textstyle\sum_{j=1}^{N} r_j \cdot \vec{U}[0,c_2] \otimes (\vec{p}_{j,t-1} - \vec{x}_{i,t-1})\end{aligned}$$

given

$$\begin{array}{rrcl} \forall r_j, j < N: & r_j &=& 2r_{j+1} \\ & \sum_{j=1}^{N} r_j &=& 1 \\ \forall \vec{p}_{j,t-1}, j < N: & f(\vec{p}_{j,t-1}) &<& f(\vec{p}_{j+1,t-1}) \end{array}$$

This leads to a more stable approach than the weighted FIPS in two respects: Firstly, the rank weights remain the same throughout the whole optimization process and are independent of the objective function. Even if progress is very slow, the ranking gives strong emphasis to better solutions. Secondly, by using the presented formula to calculate the rank weights, the influence of the neighborhood size has only a small effect on the weight of the best solution propagated. This solution has a guaranteed minimum weight factor of 0.5. Both make the ranked FIPS behave much more like the standard PSO, while it still incorporates the knowledge of the whole neighborhood. Optimization runs show reasonable particle velocities independently from the used neighborhood topology. Our experiments have shown that using the ranked FIPS with particles being included in their respective neighborhood does not lead to the erratic behavior known from the original FIPS methods. Nevertheless if particles are not included in their neighborhood, still better results are obtained.

4. ADAPTIVE MULTI-SWARM OPTIMIZER

Each particle swarm optimizer has its specific strengths and weaknesses. However, the characteristics of the objective function are rarely known beforehand. Hence, it is desirable to combine the different approaches in such a way that the swarm chooses a configuration which is well-suited for the respective problem. In order to achieve this goal, a new adaptive strategy was developed.

4.1 Presentation of the new Algorithm

The new particle swarm optimizer consists of several subswarms which cannot communicate with each other. Initially, the particles are evenly divided into a predefined number of subswarms. Each subswarm features different parameter settings, such as control parameters, inertia weight, or neighborhood topology. In each iteration, particles can migrate from one subswarm to another by keeping their position, velocity and private guide, but adopting the configuration of the new subswarm. The algorithm is therefore called *Particle Swarm Optimization with Migration (MPSO).*

To migrate a particle, it has to be deleted from its old and added to its new subswarm. As the neighborhood graph is an important algorithmic setting, the subswarms' topologies should be preserved as far as possible. If a subswarm is fully connected or connected via the *Ring* topology, adding and deleting particles is straightforward. For the *von Neumann* topology, the following approach is used: If a particle is deleted, two new communication links are added, from its left to its right neighbor, and from its top to its bottom neighbor. A gap arises which will be filled again if a particle is added to this sub-topology. If a particle is added, but no gaps are available, it is simply inserted at the end of the grid, possibly expanding it in one dimension. For deletion, a random particle is chosen due to our assumption that this decision does not greatly influence the algorithmic behavior. Figure 1 demonstrates the migration procedure for the *von Neumann* topology.

Many different strategies for triggering migration may be integrated into the general concept of MPSO. Two different approaches were investigated: A rather simple one (*MPSO-1*) and a more advanced strategy using penalty points for evaluating the subswarms' performance (*MPSO-2*). MPSO-1 and MPSO-2 will be described in the following subsections and compared with each other in Section 5.2.

4.2 MPSO-1

MPSO-1 is a rather simple realization of the multi-swarm concept. It was implemented for comparison and to determine if it is able to compete with more complicated methods.

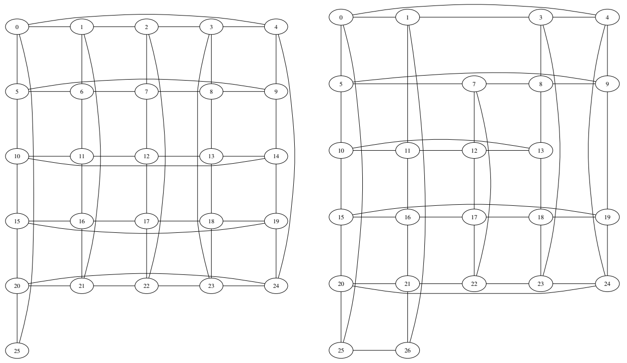

Figure 1: MPSO with 50 particles and two subswarms l, r. Initially each subswarm consisted of 25 particles. Then, l lost particles 12 and 6 which were inserted as 25 and 26 into r. Afterwards, r lost three random particles, which were inserted into the gaps of l's grid, and at the end of the grid, respectively.

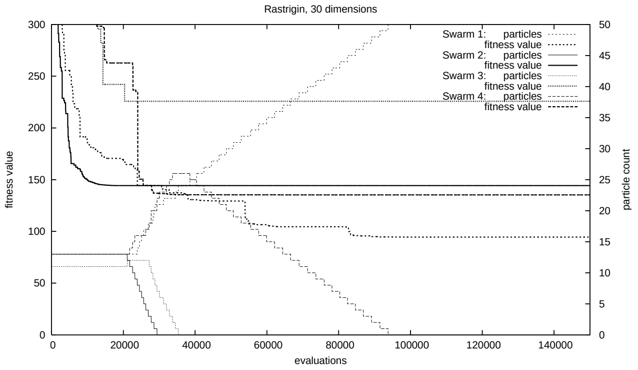

Figure 2: Sample optimization run with 4 subswarms. The current fitness value of every swarm and its particle count are shown over time. Swarm 2 is fast, but suffers from premature convergence and has to spend all particles. Swarm 3 also is not able to improve its result and loses particles. Finally, all particles belong to the steadily improving Swarm 1.

To decide which subswarms are involved in migration, fitness values are assigned to all subswarms. The fitness value of a subswarm is adopted from the best private guide currently present in the subswarm. Then, every K_3 iterations a single particle migrates from the subswarm with worst fitness to the one with best fitness, where K_3 is a user-defined parameter.

4.3 MPSO-2

To have more far reaching trends being the cause of migration, a more sophisticated method is needed than using the data of just a single iteration. We therefore introduce the concept of penalty points: scoring bad on a specified criterion results in penalty points assigned to the subswarm. A possible criterion could be the fitness value as used by MPSO-1. To avoid defining a fixed threshold for the criterion, scoring bad is defined relatively to all other subswarms. To implement this concept, the following steps are performed at each iteration:

- Measure criterion values for all subswarms and compare pairwise. If a ratio higher than P_{ratio} (user-defined) is found, assign a penalty point to the subswarm with lower value.
- Test all subswarms' penalty records against penalty limit P_{limit} (user-defined) to identify ceding subswarms.
- Sort all subswarms with respect to the latest criterion measurement. Migrate a particle from every ceding subswarm to another subswarm in this order. Reset the penalty record of the losing subswarm.

Several possible criteria come to mind: the best solution found (as in MPSO-1), the advancement regarding the best objective value found, or a measurement of convergence (difference between private guides in terms of search space position or fitness value). Here, the advancement over a window of the last 200 iterations is used, i.e. the ratio of the current fitness value and the fitness value 200 iterations ago.

Swarms should not be penalized as long as they still have potential for further improvement, while premature convergence should be penalized, even if a good local optimum was found. At the beginning of the optimization process huge fitness leaps are common. Therefore, if the advancement criterion value is greater than a threshold value of 0.01, it is set to that value. The iteration window size and the threshold value were determined by experiments on the traditional benchmark suite presented in Section 5. Although they seem to also be practical for the CEC problems, further work is needed on this topic. Figure 2 shows a sample optimization run of MPSO-2.

5. EXPERIMENTAL RESULTS

Our experimental analysis consists of two parts: First, the different neighborhood-related algorithms described in Sections 2 and 3 are compared to each other, also in combined form. Next, using the knowledge from the first experiment, adaptive multi-swarm optimizers are set up and investigated.

For the experimental analysis, eight widely-used benchmarks, namely Sphere, Rosenbrock, Rastrigin, Griewank, Ackley, Schwefel 2.6 (function descriptions, search and initialization ranges were taken from [2]), Shaffer (see [8]), Michalewicz, and the *CEC 2005 benchmark suite* [9] were used. Michalewicz is defined as follows:

$$f(\vec{x}) = -\sum_{i=1}^{n} \left(\sin(x_i) \left(\sin\left(\frac{i \cdot x_i^2}{\pi} \right) \right)^{2 \cdot m} \right), \vec{x} \in [0 \ldots 3.14]^n$$

Its initialization space was set to $[2.355 \ldots 3.14]^n$, and Shaffer's one to $[50 \ldots 100]^2$. The initialization ranges of the Schwefel 2.6 function were modified to $[-250 \ldots 250]$, so that neither the best nor the second-best local optimum are located inside the initialization space. When optimizing a *CEC 2005 benchmark*, particles were initialized with uniform distribution over search space. From the *CEC 2005 benchmarks* all noisy problems (f4 and f17), and f8 (which did not yield meaningful results because all swarms were equally unsuccessful) were skipped. A 30-dimensional parameter space was used for all benchmarks except Shaffer, which is a two-dimensional problem. Griewank was additionally tested in 10 dimensions.

All parameters which are not related to the neighborhood strategies themselves are fixed. For the other parameters, we use the following settings: As neighborhood topology, we apply the so-called *von Neumann* graph, a two-dimensional grid with wrap-around edges, as proposed by Kennedy and Mendes [10]. A particle is not included in its own neighborhood. In each iteration, the particles' *private guides* are updated simultaneously after the movement of all particles. Velocities are initialized with uniform distribution over the search space. All other parameters were chosen according to the *Standard PSO* [2]: $c_1 = c_2 = 1.496172, \omega = 0.72984$, and the swarmsize is 50. When a particle leaves the search space, it is not evaluated, and neither its position nor its velocity are altered. The PSO terminates on a limit of $300,000$ objective function evaluations. Each PSO configuration was repeated 100 times. In Table 3 (see Appendix) mean values and standard errors are presented.

5.1 Experiment I

This experiment compares different PSO variants on both a traditional and the CEC05 benchmark suites, with a total of 31 objective functions. A second focus is on the combination of these algorithms, where applicable.

5.1.1 Candidates

Sections 2 and 3 showed several different approaches to alter the PSO's behavior regarding particle communication. Stereotyping creates a second connectivity layer between the particles by means of the private guide, whereas the FIPS replaces the local guide. Finally, we survey the completing graph as a simple dynamic transformation of the neighborhood topology. Adding possible combinations, we end up with the following selection of candidates:

- `regular`: Standard PSO with Von-Neumann topology
- `st`: Stereotyping on private guide
- `fu, fw, fr`: unweighted, weighted and ranked FIPS
- `stfr`: ranked FIPS with stereotyping
- `cg, stcg, frcg`: completing graph with generalized edge insertion on `regular`, `st` and `fr`, respectively
- `dircg`: directed completing graph based on ring topology, with circular edge insertion

The FIPS variants `fu, fw, fr, frcg`, the following settings are applied: $c_1 = 0, c_2 = 2.99234$. The completing graphs are configured to be finished at iteration $K_2 = 6000$, which is the minimal count of iterations passed.

5.1.2 Results

From the results in Table 3 and the ranks presented in Table 1, we can conclude that each presented modification is able to provide significant performance improvements over the Standard PSO, but not all of the tested combinations are able to do so. Although some of the candidates seem to be more suitable for one of the used benchmark suites than for the other, the correlation between the rank sums $\sum_1$ and $\sum_2$ is very high (0.786).

Stereotyping is a strategy that was mostly overlooked ever since its introduction. Our results give reason to reconsider: `st` achieves better results than `regular` on all problems except f15, where only `dircg` beats `regular`. Also, the best solutions are often found by `st`. Stereotyping can be recommended even for completely unknown problems, as the solution quality is always good in this experiment. The low standard deviation of its ranks (the lowest of all candidates) is another indicator for reliability.

The `fr` strategy manages to produce impressive results over the whole test set, on par with `st`. Compared to `st`, `fr` finds the best solutions more often, but has more serious problems, too. While the other FIPS variants also show improvements over the Standard PSO on most problems, they deliver worse results for 11 (`fu`) and 9 problems (`fw`), respectively. However, the neighborhood configuration chosen is the best suited for FIPS according to [4] and our earlier experiments. The ranked FIPS does not completely dominate `regular`, either. Both `fu` and `fw` have serious problems with Michalewicz and Ackley, while `fr` at least works well on Ackley. Interestingly, `frcg` works very well on Michalewicz, but degrades performance for many other problems.

As both stereotyping and the ranked FIPS manage to improve the optimization process, and they do so in different fields, it is reasonable to search for a good combination of both. However, `stfr` failed and led to the worst results on many problems. The reason is that the FIPS does not work well in combination with a private guide, as own experiments have shown. In subsection 5.2 we try a more sophisticated combination.

The key idea of the completing graph method was the transition from exploration to exploitation. However, our results show that densifying the neighborhood graph is not sufficient to produce the supposed benefits. Only `dircg` achieves a significant performance improvement, while all other combinations fail to do so. Previous experiments presented measurable, but insignificant differences between circular edge insertion and generalized least-grade insertion regarding the optimization process using an undirected ring as base topology. This changes with directed edges in the ring topology, where a solution has to be communicated to the whole swarm before it can become the local guide for the originating particle. The swarm converges very slowly. Due to the completion, exploitation still comes into play near the end of the optimization process. Note that it is important to exclude a particle from its neighborhood to prevent local convergence.

This experiment has two clear winners: Kennedy's stereotyping (`st`) and the ranked FIPS (`fr`). While the completing graph also works very well in a very specific setting, our efforts to generalize it proved fruitless.

5.2 Experiment II

Based on the previous experiment, we investigate promising multi-swarm configurations using MPSO-1 and MPSO-2. The number of subswarms involved needs to be chosen along the subswarm's configurations. It is hard to find a high number of subswarms complementing each other well. Also, a higher number of subswarms result in a lower particle count per swarm. Hence, in this experiment different configurations with only two or three subswarms are chosen.

5.2.1 Candidates

Experiment I showed that `fr` and `st` both provide the best performance among the candidates, each having different strengths and weaknesses. Therefore, both were chosen as subswarm strategies for all MPSO configurations. Candidates `fu` and `fw` seem to be an appropriate supplement, while `fu` performs better than both `st` and `fr` for 4 problems and is the single best solution for 2 problems. Alternatively, `dircg` could be favored, being a completely different approach from the first two subswarms.

As a result, the following candidates were tested:

- `M1-2`, `M2-2`: MPSO-1, MPSO-2 with `fr` and `st`
- `M1-3a`, `M2-3a`: MPSO-1, MPSO-2 with `fr`, `st`, `dircg`

	Sphere	Rosenbrock	Ackley	Griewank10	Griewank	Michalewicz	Rastrigin	Schwefel	Shaffer	$\sum_1$
st	**1**	4	**1**	5	5	3	2	4	**1**	**26**
stcg	**1**	3	**1**	6	6	2	4	5	**1**	29
stfr	**1**	10	8	2	4	7	10	10	8	60
fu	**1**	9	10	**1**	**1**	9	9	9	**1**	50
fw	**1**	8	9	3	**1**	10	6	7	**1**	46
fr	**1**	6	**1**	4	**1**	8	**1**	**1**	8	31
frcg	**1**	**1**	4	7	7	**1**	3	8	8	40
regular	**1**	7	7	9	10	4	8	6	**1**	53
cg	**1**	2	6	10	9	6	7	3	**1**	45
dircg	**1**	5	5	8	8	5	5	2	**1**	40

	f1	f2	f3	f5	f6	f7	f9	f10	f11	f12	f13	f14	f15	f16
st	**1**	**1**	8	4	3	**1**	5	6	4	5	**1**	3	5	5
stcg	**1**	**1**	3	3	2	2	6	5	3	**1**	2	4	6	3
stfr	**1**	**1**	2	9	10	8	8	7	6	10	5	7	7	10
fu	**1**	**1**	5	6	8	9	3	4	7	3	9	5	4	8
fw	**1**	**1**	7	5	7	10	2	2	5	4	10	6	3	7
fr	**1**	**1**	4	**1**	4	3	**1**	**1**	**1**	6	4	2	10	**1**
frcg	**1**	**1**	**1**	2	**1**	4	4	3	2	2	3	**1**	8	2
regular	**1**	10	10	10	9	6	10	9	9	8	8	10	2	6
cg	**1**	**1**	9	8	5	7	9	10	10	9	7	9	9	9
dircg	**1**	**1**	6	7	6	5	7	8	8	7	6	8	**1**	4

	f18	f19	f20	f21	f22	f23	f24	f25	$\sum_2$	$\sum_1 + \sum_2$	std.dev.
st	**1**	2	**1**	6	3	5	**1**	5	76	102	**1.95**
stcg	2	**1**	2	7	2	7	**1**	6	70	99	2.01
stfr	7	7	7	10	10	10	10	10	162	222	3.00
fu	6	6	6	**1**	9	3	8	4	116	166	3.09
fw	5	5	5	4	8	2	**1**	3	103	149	2.91
fr	3	3	3	2	4	**1**	**1**	**1**	**58**	**89**	2.43
frcg	4	4	4	3	**1**	6	**1**	2	60	100	2.31
regular	10	9	9	8	7	8	**1**	9	169	222	2.98
cg	9	10	10	9	6	9	9	8	173	218	3.02
dircg	8	8	8	5	5	4	**1**	7	121	161	2.52

Table 1: Ranks of the tested candidates in Experiment I. Sums of ranks for the traditional and the CEC05 benchmarks, standard deviation of all ranks. Best values are printed bold. Mean values and standard errors are presented in the Appendix.

- `M1-3b`, `M2-3b`: MPSO-1, MPSO-2 with `fr`, `st` and `fu`

For the MPSO-1 variants, the iteration count $K_3 = 60$ was chosen. For the MPSO-2 variants, penalty limit $P_{limit} = 60$ and penalty ratio $P_{ratio} = 2$ are used.

5.2.2 Results

For an MPSO strategy to be considered successful, the overall performance should be at least in between the performance of its parts (i.e., the strategies used in the subswarms). The main advantage has to be a broader applicability due to its adaptive characteristic. Both the `M1-2` and `M2-2` configurations achieve this goal, producing comparable results with significantly lower standard deviations of their respective ranks, as seen in Table 2.

The extra efforts spent on the more complicated design of `M2-2` pay off: `M2-2` reaches both the lowest rank sum and the lowest standard deviation. The mean solutions found are worse compared to both `st` and `fr` on 6 problems. However, they are still of very good quality for all problems tested. Remarkably, on some problems `M2-2` is even the single best candidate. The Standard PSO beats `M2-2` only on Shaffer, where `M2-2` sometimes fails to reach the global optimum. An interesting insight was provided by Rosenbrock: On this function, the advancement criterion doesn't trigger migration well, as the swarms never stop to steadily improve.

It can be observed that the overall results of both `M2-3a`, `M2-3b` and both `M1-3a`, `M1-3b` are very similiar, but worse than their 2-subswarm counterparts. The supplemental subswarms tried here cannot sufficiently support the whole optimization process. The higher number of subswarms leads to smaller subswarm sizes, making the optimization process less efficient to start with.

Other than the variants with 3 subswarms involved, both `M2-2` and `M1-2` already show very promising results, even though they weren't analyzed in detail yet and are merely in a proof-of-concept state.

6. CONCLUSION AND FUTURE WORK

The work presented here deals with interaction among particles. Different methods of communication and altering the particles' relationships were examined. A new variant of the popular FIPS algorithm was introduced, with promising qualities observed. It was tested within a wide context of neighborhood-related modifications to PSO. Different possible combinations of these were explored for the first time. To achieve meaningful and comparable results, tests were run both on a comprehensive set of traditionally used benchmark problems and on the newly established CEC05 benchmark suite. Furthermore, a new adaptive strategy was introduced, combining different PSO variants to produce a reliable performance over a wider spectrum of functions. Apart from the general concept of migration between subswarms, two different realizations of which were devised, with focus on simplicity and sophistication, respectively. Complementing subswarm configurations were successfully derived from the results of the previous experiment.

It was observed that both Kennedy's stereotyping and the newly introduced ranked FIPS can be considered as a major improvement over the canonical PSO algorithm. The ranked FIPS allows the FIPS concept to be used independently of the neighborhood topology chosen, and additionally performs significantly better than the original FIPS. Kennedy and Mendes suggested to seek after very best suited neighborhood topologies for both PSO and FIPS [5]. Research

	Sphere	Rosenbrock	Ackley	Griewank10	Griewank	Michalewicz	Rastrigin	Schwefel	Shaffer	$\sum_1$
fr	**1**	2	**1**	**1**	**1**	9	**1**	**1**	8	**25**
st	**1**	**1**	**1**	4	7	**1**	2	7	**1**	**25**
regular	**1**	9	9	9	9	7	9	9	**1**	63
M1-2	**1**	4	4	6	6	4	3	2	**1**	31
M1-3a	**1**	6	6	8	8	5	6	3	**1**	44
M1-3b	**1**	8	7	2	5	8	7	6	**1**	45
M2-2	**1**	5	3	5	4	3	4	4	7	36
M2-3a	**1**	3	5	7	**1**	2	5	5	**1**	30
M2-3b	**1**	7	8	3	**1**	6	8	8	8	50

	f1	f2	f3	f5	f6	f7	f9	f10	f11	f12	f13	f14	f15	f16
fr	**1**	**1**	2	**1**	2	7	**1**	**1**	**1**	7	8	**1**	9	7
st	**1**	**1**	4	4	**1**	2	4	4	8	6	**1**	4	8	8
regular	**1**	9	8	9	9	9	9	9	9	9	9	9	7	9
M1-2	**1**	**1**	7	3	5	8	2	2	3	5	3	3	6	4
M1-3a	**1**	**1**	6	5	6	**1**	5	5	4	4	4	7	3	6
M1-3b	**1**	**1**	9	8	8	5	6	6	5	8	6	8	4	2
M2-2	**1**	**1**	**1**	2	3	3	3	3	2	3	2	2	5	3
M2-3a	**1**	**1**	5	7	4	4	7	7	7	2	5	6	**1**	5
M2-3b	**1**	**1**	3	6	7	6	8	8	6	**1**	7	5	2	1

	f18	f19	f20	f21	f22	f23	f24	f25	$\sum_2$	$\sum_1 + \sum_2$	std.dev.
fr	3	5	5	4	2	**1**	**1**	2	72	97	2.80
st	**1**	2	2	8	**1**	8	**1**	6	85	110	2.69
regular	9	9	9	9	7	9	**1**	8	176	239	2.65
M1-2	5	3	3	5	4	7	**1**	3	84	115	1.90
M1-3a	8	7	7	7	8	2	**1**	7	105	149	2.39
M1-3b	6	6	6	3	9	5	**1**	**1**	114	159	2.66
M2-2	2	**1**	**1**	**1**	3	3	**1**	5	**51**	**87**	**1.51**
M2-3a	7	8	8	6	5	6	**1**	9	112	142	2.49
M2-3b	4	4	4	2	6	4	**1**	4	91	141	2.59

Table 2: Ranks of the tested candidates in Experiment II. Sums of ranks for the traditional and the CEC05 benchmarks, standard deviation of all ranks. Best values are printed bold. Mean values and standard errors are presented in the Appendix.

not presented here revealed that well suited topologies for the regular PSO algorithm seem to be equally-well suited for the ranked FIPS though. The problems of FIPS on Michalewicz are still to be explained, whereas analyzing the combination of the ranked FIPS with the completing graph on Michalewicz could give viable insights.

The investigations into the completing graph as a dynamic modification to the neighborhood topology yielded an interesting observation: In order to produce the desired balance between exploration and exploitation over time, it is possible to model the feedback loop of particles using accordingly designed directed neighborhood graphs.

MPSO as a new approach to an adaptive optimizer already produced very good results, while no major drawbacks were observed. The main goal was an increased reliability compared to non-adaptive algorithms. This was achieved without a distinct performance tradeoff. Still, many aspects need further analysis, particularly of the described MPSO-2 variant. The amount of newly introduced parameters should be minimized. While the measurement was done on advancement, ultimately it worked best to identify convergence. Other methods to achieve similiar characteristics should be considered, especially with simplicity in mind.

7. REFERENCES

[1] J. Kennedy and R. C. Eberhart. *Swarm Intelligence.* Morgan Kaufmann Academic Press, 2001.

[2] D. Bratton and J. Kennedy. Defining a standard for Particle Swarm Optimization. In *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, 2007.

[3] J. Kennedy. Stereotyping: Improving particle swarm performance with cluster analysis. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 1507–1512. IEEE Service Center, 2000.

[4] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, 2004.

[5] J. Kennedy and R. Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man and Cybernetics*, 36(4):515–519, July 2006.

[6] P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1958–1962. IEEE Press, 6-9 July 1999.

[7] M. Richards and D. Ventura. Dynamic sociometry in particle swarm optimization. In *Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing*, pages 1557–1560, North Carolina, September 2003.

[8] R. Mendes, J. Kennedy, and J. Neves. Watch thy neighbor or how the swarm can learn from its environment. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pages 88–94. IEEE, 2003.

[9] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical report, Nanyang Technological University, Singapore, 2005.

[10] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, pages 1671–1676. IEEE Press, 2002.

APPENDIX

Table 3: Mean best fitness value und standard errors of all investigated PSO configurations. The best possible objective value is presented together with the function name.

	Sphere (0)	Rosenbrock (0)	Ackley (0)	Griewank10 (0)	Griewank (0)	Michalewicz
st	0 ± 0	12.6828 ± 0.64128	0 ± 0	0.0097496 ± 0.001084122	0.00088427 ± 0.000477080	−27.048 ± 0.0947
stcg	0 ± 0	11.0347 ± 0.69692	0 ± 0	0.01435008 ± 0.001285017	0.00105974 ± 0.000300287	−27.273 ± 0.0773
stfr	0 ± 0	83.1315 ± 6.84333	19.8251 ± 0.02047	0.00103504 ± 0.000323430	0.00027076 ± 0.000156931	−21.534 ± 0.4549
fu	0 ± 0	28.8177 ± 2.16316	20.6174 ± 0.00826	0.0008716 ± 0.000288123	0 ± 0	−14.354 ± 0.0769
fw	0 ± 0	28.0251 ± 1.92892	20.6085 ± 0.00880	0.00161604 ± 0.000398050	0 ± 0	−14.321 ± 0.0689
fr	0 ± 0	14.5786 ± 0.35119	0 ± 0	0.00598495 ± 0.000781610	0 ± 0	−14.647 ± 0.1013
frcg	0 ± 0	3.158 ± 0.18863	1.3613 ± 0.49620	0.0148625 ± 0.001159888	0.00137997 ± 0.000359280	−27.595 ± 0.0689
regular	0 ± 0	27.1572 ± 2.81880	4.7595 ± 0.82488	0.02636437 ± 0.001900490	0.00524597 ± 0.000734530	−24.454 ± 0.0960
cg	0 ± 0	7.9983 ± 0.59072	3.5714 ± 0.74121	0.02883631 ± 0.002114088	0.00490211 ± 0.000657427	−24.302 ± 0.1149
dircg	0 ± 0	14.4438 ± 1.85019	2.0022 ± 0.56470	0.02099247 ± 0.001734157	0.0018973 ± 0.000413487	−24.352 ± 0.0949
M1-2	0 ± 0	16.8173 ± 1.12045	0.3837 ± 0.26862	0.01132982 ± 0.001063869	0.00058903 ± 0.000209316	−24.912 ± 0.2964
M1-3a	0 ± 0	18.8044 ± 1.60735	3.0507 ± 0.69951	0.01525 ± 0.001284279	0.00108413 ± 0.000327385	−24.671 ± 0.2014
M1-3b	0 ± 0	25.1636 ± 2.26098	4.2748 ± 0.80505	0.00710593 ± 0.000745932	0.00039429 ± 0.000202352	−24.332 ± 0.3031
M2-2	0 ± 0	18.3962 ± 1.27212	0.382 ± 0.26740	0.01057738 ± 0.001058579	0.00034504 ± 0.000174323	−25.306 ± 0.2370
M2-3a	0 ± 0	16.5593 ± 1.16814	2.9681 ± 0.68641	0.01457251 ± 0.001321692	0 ± 0	−25.775 ± 0.1310
M2-3b	0 ± 0	22.5293 ± 1.91066	4.4205 ± 0.81027	0.00849868 ± 0.001027115	0 ± 0	−24.578 ± 0.2169

	Rastrigin (0)	Schwefel (-12569.487)	Shaffer (0)	f1 (-450)	f2 (-450)	f3 (-450)
st	38.495 ± 0.8981	−8197.3 ± 73.34	0 ± 0	−450 ± 0	−450 ± 0	528923 ± 15309.9
stcg	42.704 ± 1.1982	−8170.6 ± 64.52	0 ± 0	−450 ± 0	−450 ± 0	360494 ± 11051.3
stfr	165.693 ± 3.2835	−4830.2 ± 51.43	0.000194318 ± 0.000136023	−450 ± 0	−450 ± 0	313812 ± 8978.58
fu	102.301 ± 2.4993	−5807.9 ± 198.36	0 ± 0	−450 ± 0	−450 ± 0	474109 ± 11844.3
fw	98.361 ± 2.2244	−8086 ± 133.42	0 ± 0	−450 ± 0	−450 ± 0	481954 ± 12596.1
fr	25.037 ± 0.7446	−8920.3 ± 61.02	0.000194318 ± 0.000136023	−450 ± 0	−450 ± 0	463154 ± 12930.1
frcg	39.798 ± 1.0925	−6518.4 ± 74.66	0.000194318 ± 0.000136023	−450 ± 0	−450 ± 0	210984 ± 7209.02
regular	100.938 ± 2.3547	−8100.9 ± 46.70	0 ± 0	−450 ± 0	−449.99983 ± 0.000047009	916289 ± 36341.9
cg	99.346 ± 2.6098	−8439.1 ± 55.10	0 ± 0	−450 ± 0	−450 ± 0	585366 ± 22135.1
dircg	71.985 ± 1.7065	−8832.9 ± 54.05	0 ± 0	−450 ± 0	−450 ± 0	476351 ± 18227.4
M1-2	45.256 ± 1.2900	−8628.3 ± 61.35	0 ± 0	−450 ± 0	−450 ± 0	909012 ± 47909.4
M1-3a	71.837 ± 1.8238	−8453.4 ± 55.99	0 ± 0	−450 ± 0	−450 ± 0	852047 ± 46839.0
M1-3b	71.9 ± 2.1901	−8370.1 ± 52.99	0 ± 0	−450 ± 0	−450 ± 0	1365630 ± 77000.9
M2-2	50.518 ± 1.3737	−8452.7 ± 57.50	7.9904e-05±0.000079504	−450 ± 0	−450 ± 0	456816 ± 13749.5
M2-3a	69.463 ± 1.8485	−8381.2 ± 56.67	0 ± 0	−450 ± 0	−450 ± 0	529336 ± 18709.6
M2-3b	84.204 ± 1.8550	−8148.1 ± 66.34	0.000194318 ± 0.000136023	−450 ± 0	−450 ± 0	494060 ± 15043.0

	f5 (-310)	f6 (390)	f7 (-180)	f9 (-330)	f10 (-330)	f11 (90)
st	3168.1 ± 41.75	405.34 ± 1.199	−179.98943 ± 0.000841	−289.01 ± 0.971	−279.98 ± 0.997	104.87 ± 0.408
stcg	3022 ± 38.44	401.55 ± 0.4790	−179.98851 ± 0.000785	−287.73 ± 0.901	−281.85 ± 1.136	104.18 ± 0.333
stfr	4482.4 ± 71.86	660.52 ± 32.292	−179.98094 ± 0.00266	−271.51 ± 1.142	−269.45 ± 1.049	108.00 ± 0.454
fu	3337.9 ± 36.60	427.89 ± 3.188	−179.97524 ± 0.00448	−300.94 ± 0.560	−282.04 ± 1.948	108.27 ± 0.531
fw	3253.4 ± 28.72	420.47 ± 2.397	−179.97298 ± 0.00503	−303.82 ± 0.528	−295.38 ± 1.792	106.33 ± 0.492
fr	2633.5 ± 28.96	407.51 ± 1.677	−179.98793 ± 0.000770	−304.60 ± 0.661	−301.09 ± 0.565	101.27 ± 0.287
frcg	2655 ± 37.88	400.94 ± 2.711	−179.98782 ± 0.000999	−291.35 ± 1.017	−294.22 ± 0.754	103.87 ± 0.367
regular	4652.7 ± 105.93	444.96 ± 6.363	−179.98262 ± 0.00130	−265.28 ± 1.759	−237.71 ± 2.406	119.02 ± 0.246
cg	4396.7 ± 99.97	412.44 ± 3.243	−179.98161 ± 0.00153	−266.29 ± 2.016	−233.77 ± 3.574	119.03 ± 0.312
dircg	3855.4 ± 88.14	414.64 ± 3.332	−179.98314 ± 0.00126	−280.59 ± 1.363	−255.36 ± 1.799	118.23 ± 0.350
M1-2	2913 ± 41.48	414.64 ± 2.465	−179.98749 ± 0.00101	−296.36 ± 0.716	−290.64 ± 1.371	102.53 ± 0.313
M1-3a	3255.3 ± 66.20	415.46 ± 2.899	−179.99022 ± 0.000674	−288.46 ± 1.087	−278.77 ± 1.417	103.11 ± 0.313
M1-3b	3475 ± 62.54	421.35 ± 3.940	−179.98823 ± 0.00103	−287.83 ± 0.894	−276.73 ± 1.524	104.16 ± 0.290
M2-2	2895.3 ± 47.39	409.6 ± 1.289	−179.98934 ± 0.000944	−294.32 ± 0.789	−286.81 ± 1.266	101.42 ± 0.263
M2-3a	3390.2 ± 65.14	412.72 ± 2.592	−179.98877 ± 0.00108	−287.55 ± 0.975	−273.02 ± 1.269	104.57 ± 0.323
M2-3b	3369.8 ± 54.18	418.82 ± 3.843	−179.98821 ± 0.00122	−284.82 ± 0.947	−270.07 ± 1.656	104.19 ± 0.292

	f12 (-460)	f13 (-130)	f14 (-300)	f15 (120)	f16 (120)	f18 (10)	f19 (10)
st	4045.9 ± 502.42	−127.37 ± 0.065	−288.24 ± 0.052	457.36 ± 7.218	324.79 ± 18.651	886.11 ± 5.596	892.44 ± 5.289
stcg	3394 ± 396.38	−127.36 ± 0.061	−288.20 ± 0.050	460.17 ± 7.450	304.29 ± 17.274	888.38 ± 5.514	891.98 ± 5.387
stfr	11545.6 ± 1050.60	−126.79 ± 0.080	−287.74 ± 0.041	464.38 ± 7.338	374.97 ± 18.411	913.76 ± 5.251	916.76 ± 4.889
fu	3576.4 ± 559.00	−124.71 ± 0.209	−288.08 ± 0.031	457.03 ± 6.832	351.03 ± 17.664	907.01 ± 4.556	914.30 ± 3.875
fw	3839.6 ± 421.41	−124.49 ± 0.228	−288.04 ± 0.032	452.33 ± 5.878	336.71 ± 19.148	901.28 ± 5.000	909.86 ± 4.382
fr	4061.9 ± 452.68	−127.02 ± 0.071	−288.43 ± 0.037	487.37 ± 7.825	277.18 ± 16.820	889.35 ± 5.203	900.29 ± 4.524
frcg	3502.2 ± 571.12	−127.13 ± 0.062	−288.60 ± 0.054	477.16 ± 7.763	295.96 ± 17.899	894.39 ± 5.012	904.21 ± 4.270
regular	10448.3 ± 991.29	−125.74 ± 0.122	−287.49 ± 0.033	447.08 ± 9.493	333.99 ± 9.913	936.56 ± 1.979	932.83 ± 2.483
cg	10625.7 ± 1291.83	−125.89 ± 0.110	−287.61 ± 0.050	478.91 ± 11.492	367.50 ± 10.926	933.48 ± 3.275	935.34 ± 2.757
dircg	6945.5 ± 803.79	−126.4 ± 0.087	−287.66 ± 0.045	435.91 ± 8.334	306.64 ± 9.387	915.84 ± 3.966	919.87 ± 3.728
M1-2	3863 ± 510.48	−127.24 ± 0.066	−288.32 ± 0.041	431.44 ± 5.296	260.75 ± 12.949	897.22 ± 4.915	895.12 ± 4.921
M1-3a	3324.4 ± 350.87	−127.17 ± 0.057	−288.17 ± 0.047	407.14 ± 7.603	274.03 ± 10.555	908.25 ± 4.303	907.99 ± 4.295
M1-3b	4604.6 ± 442.10	−127.12 ± 0.068	−288.09 ± 0.039	407.89 ± 7.309	248.67 ± 10.379	900.56 ± 5.112	901.64 ± 5.025
M2-2	2534.4 ± 355.06	−127.29 ± 0.047	−288.37 ± 0.065	426.15 ± 7.705	256.78 ± 12.349	886.64 ± 5.385	880.73 ± 5.545
M2-3a	2476.7 ± 303.31	−127.13 ± 0.060	−288.21 ± 0.057	402.44 ± 7.841	270.54 ± 9.387	907.18 ± 4.577	912.99 ± 4.017
M2-3b	2353.2 ± 300.79	−127.05 ± 0.082	−288.22 ± 0.058	405.79 ± 6.973	245.43 ± 8.901	895.84 ± 5.368	896.97 ± 5.174

	f20 (10)	f21 (360)	f22 (360)	f23 (360)	f24 (260)	f25 (260)
st	892.48 ± 5.286	921.244 ± 18.667	1304.7 ± 2.456	925.74 ± 13.765	460 ± 0	513.16 ± 0.232
stcg	893.57 ± 5.234	947.458 ± 21.937	1296.0 ± 2.367	942.22 ± 16.464	460 ± 0	513.40 ± 0.245
stfr	920.14 ± 4.717	977.019 ± 24.977	1378.3 ± 3.127	1081.67 ± 28.206	519.87 ± 23.722	521.70 ± 9.702
fu	917.48 ± 3.404	860.000 ± 0	1347.4 ± 1.579	919.35 ± 12.336	469.79 ± 9.743	511.54 ± 0.226
fw	906.80 ± 4.545	875.879 ± 9.456	1346.9 ± 1.735	910.88 ± 9.630	460 ± 0	511.43 ± 0.238
fr	900.33 ± 4.526	866.389 ± 6.357	1306.0 ± 0.994	894.16 ± 0.000049	460 ± 0	509.61 ± 0.133
frcg	900.77 ± 4.545	869.411 ± 7.016	1287.1 ± 2.423	931.70 ± 13.965	460 ± 0	510.72 ± 0.140
regular	932.71 ± 2.478	974.268 ± 22.982	1323.6 ± 3.480	949.77 ± 16.356	460 ± 0	515.78 ± 0.289
cg	936.99 ± 2.861	974.670 ± 24.633	1322.8 ± 3.933	987.62 ± 20.583	473.26 ± 10.610	515.27 ± 0.266
dircg	921.84 ± 3.360	898.478 ± 13.535	1313.8 ± 3.250	925.23 ± 12.601	460 ± 0	514.47 ± 0.257
M1-2	893.35 ± 5.077	866.494 ± 6.461	1315.1 ± 1.749	925.60 ± 13.702	460 ± 0	511.03 ± 0.170
M1-3a	909.40 ± 4.197	872.561 ± 7.711	1332.7 ± 2.513	898.29 ± 4.0194	460 ± 0	514.12 ± 0.286
M1-3b	902.41 ± 4.923	865.580 ± 6.526	1339.8 ± 2.517	910.47 ± 7.987	460 ± 0	509.53 ± 1.370
M2-2	885.42 ± 5.422	865.575 ± 6.521	1307.1 ± 2.103	906.77 ± 8.825	460 ± 0	512.68 ± 0.209
M2-3a	912.51 ± 3.994	868.685 ± 7.247	1320.1 ± 2.449	910.79 ± 9.634	460 ± 0	515.87 ± 0.322
M2-3b	898.22 ± 5.112	865.576 ± 6.522	1322.6 ± 2.869	908.76 ± 8.4038	460 ± 0	511.12 ± 1.216