Impact of Communication Topology in Particle Swarm Optimization

Tim Blackwell and James Kennedy

Abstract—**Particle swarm optimization (PSO) has two salient components: 1) a dynamical rule governing particle motion and 2) an interparticle communication topology. Recent practice has focused on the fully connected topology (Gbest) despite earlier indications on the superiority of local particle neighborhoods. This paper seeks to address the controversy with empirical trials with canonical PSO on a large benchmark of functions, categorized into 14 properties. This paper confirms the early lore that Gbest is the overall better algorithm for unimodal and separable problems and that a ring neighborhood of connectivity two (Lbest) is the preferred choice for multimodal, nonseparable and composition functions. Topologies of intermediate particle connectivity were also tested and the difference in global/local performance was found to be even more marked. A measure of significant improvement is introduced in order to distinguish major improvements from refinements. Lbest, according to the experiments on the 84 test functions and a bi-modal problem of adjustable severity, is found to have significant improvements later in the run, and to be more diverse at termination. A mobility study shows that Lbest is better able to jump between optimum basins. Indeed Gbest was unable to switch basins in the bi-modal trial. The implication is that Lbest's larger terminal diversity, its better ability to basin hop and its later significant improvement account for the performance enhancement. In several cases where Lbest was not the better algorithm, the trials show that Lbest was not stuck but would have continued to improve with an extended evaluation budget. Canonical PSO is a baseline algorithm and the ancestor of all contemporary PSO variants. These variants build on the basic structure of baseline PSO and the broad conclusions of this paper are expected to follow through. In particular, research that fails to consider local topologies risks underplaying the success of the promoted algorithm.**

Index Terms—**Particle swarm optimization (PSO).**

I. Introduction

Since its introduction in 1995, the particle swarm optimization (PSO) algorithm has gone through many changes. The dynamics of the particles have been studied, parameter values have been evaluated, and extended and compact versions have been proposed. The communication topology of the swarm has been investigated, and various dynamic and adaptive variations have been put forward. A March 2018 search for the string "particle swarm" in Google Scholar returned 261 000 results; the algorithm has been looked at quite intensively by a curious and active research community.

For all the variations and innovations, though, the version that has ascended as the "standard" in a majority of recent publications is not the version that has usually been judged as superior over the past 20 plus years. In particular, the research community has settled on the global-best (Gbest) topology for the great majority of implementations, even when distributed communication topologies have been uniformly preferred and praised in the literature. This has sometimes led to disappointing research results; a slight modification of the algorithm might have been successful.

The very first particle swarm papers [1], [2], described a primitive algorithm with some surprising properties. Based on the metaphor of social learning, a population of initially random candidate problem solutions, or particles, moved through the search space and informed one another of better positions. Teaching and learning from one another simultaneously, the system—swarm—evolved toward a global optimum in a large number of standard test problems.

Even in 1995, two kinds of interparticle communication network were proposed. The Gbest network keeps track of the best solution found by any member of the population, and shares that information with all particles. The ring network—known here as Lbest—is the most extreme local topology; it permits information sharing between immediate neighbors only. Even in that first year it was noted that the Lbest topology seemed to make the swarm relatively immune to the attraction of local optima, while Gbest ran quickly but was susceptible to getting stuck.

Over recent years the research community has largely assumed Gbest as the default or standard topology, and, in 2013, Engelbrecht [3] reported that, on a 60 function benchmark, there was no real performance difference between G and Lbest. This conclusion went against a large amount of published research.

This paper seeks to clarify the G/Lbest controversy with rigorous testing on a large benchmark that combines the popular CEC test functions with the problems chosen in the 2013 study. In order to match performance with function characteristics, correlations between error and 14 binary problem properties were investigated. Furthermore, an insight into G/Lbest behavior is obtained by monitoring how late into a run a swarm continues to improve, its diversity at the end of a

Manuscript received March 23, 2018; revised July 9, 2018 and September 21, 2018; accepted October 30, 2018. Date of publication November 12, 2018; date of current version July 30, 2019. *(Corresponding author: Tim Blackwell.)*

T. Blackwell is with the Department of Computing, Goldsmiths, University of London, London SE14 6NW, U.K. (e-mail: t.blackwell@gold.ac.uk).

J. Kennedy was with the Office of Analysis and Research Services, U.S. International Trade Commission, Washington, DC 20436 USA (e-mail: kennedy.jim@gmail.com).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Digital Object Identifier 10.1109/TEVC.2018.2880894

run, and its ability to jump between promising modal basins. This last quality is investigated with a purpose-built bi-modal function that allows for tight control of problem difficulty.

The main finding of this paper is that the topology affects many aspects of the swarm's performance, and a distributed topology offers advantages, especially in the case of more difficult problems.

This paper continues with a review of PSO particle communication; the aim is to spotlight the G/Lbest controversy and argue for a definitive response. Section III tests G/Lbest on the combined benchmark and links performance with the presence of one or more of the fourteen binary function properties. A study of Gbest/Lbest stagnation—the tendency of an algorithm to get stuck—and mobility—the ability to jump between optimum basins—follows. Tests on real-world problems are reported in Section VI. A study of other local topologies completes the empirical investigations; all findings are gathered and presented in the concluding section.

II. Particle Communication

This section gives an overview of the particle swarm algorithm and a review of communication topology and PSO performance. The section ends with comments on the uptake of Gbest as the default topology and summarizes Engelbrecht's [3] 2013 study.

A. Particle Swarm

The particle swarm described herewith is the "canonical" PSO [4]–[8] which differs only from the original 1995 PSO by the inclusion of a convergence controlling inertia weight (equivalent to Clerc and Kennedy's [6] constriction parameter). This PSO contains the two essential components—dynamics and topology—at a conceptually fundamental level. It is widely accepted as the baseline PSO and is invariably used as a reference model in contemporary research.

Each individual, or particle, i in the swarm of M particles is the triplet (x_i, v_i, p_i) where x_i and p_i are D dimensional vectors in the search space X. x_i is the current position, p_i (pbest) is the previous best position achieved (pbest may be equal to x_i), and the velocity v_i is the difference between the current and the immediately prior position. Particles move by adding an updated velocity to x_i. The new position is evaluated with respect to the objective function and, if that position is better, or equal, to any position it has found so far, the position is stored in p_i.

Specifically, the velocity update rule is

$$v_i(t+1) = wv_i(t) + cu_1(t) \circ (n_i(t) - x_i(t)) + cu_2(t) \circ (p_i(t) - x_i(t)) \quad (1)$$

where $u_{1,2}$ are uniform random variables in $[0, 1]^D$ and $\circ$ is the Hadamard (entry-wise) product. n_i is the pbest of the best neighbor, as determined by the communication topology and the inertial weight, w, and c, are two arbitrary parameters, chosen to encourage search by moderating convergence whilst preventing explosion.

The topology typically consists of bidirectional edges connecting pairs of particles, so that if j is in i's neighborhood, i is also in j's. Each particle communicates with other particles in its neighborhood and is influenced by the pbest of the best neighbor. Population topologies are potentially hugely varied, but in practice certain types have been used more frequently. The present paper compares aspects of the two most widely used topologies.

PSO has two arbitrary components, parameter values and the choice of network, and each has been the subject of considerable investigation over the past 20 years.

1) Parameters: Clerc and Kennedy [6] analyzed the system at stagnation and determined that a constriction coefficient χ, equivalent to the inertial weight of (1) with value 0.7968 along with acceleration constants c fixed at 1.4962 provided optimal performance. Shi and Eberhart [4] came to a similar conclusion with slightly different values and a slightly different algebraic arrangement. Subsequent research has mostly adopted the Clerc and Kennedy [6] values and the Shi and Eberhart [4] arrangement of inertia weight, rather than constriction coefficient. The inertia weight encourages convergence by gearing the velocity down.

2) Topology: The first PSO paper, published in 1995, proposed two methods for organizing the particles in order to manipulate the flow of information through the population [2]. One method, called "Gbest," allowed every member of the population to be influenced by the member that had achieved the best performance so far.

In the second topology, "Lbest," however, particles were connected by a sparse network of low connectivity. In a ring neighborhood, for instance, particle i compares its best position with particles $i-1$ and $i+1$ (with appropriate wraparound). Lbest was found to successfully optimize a set of weights of an XOR neural network and, despite hundreds of trials, convergence to a local optimum was never seen. The authors suggested that the good behavior of Lbest could be attributed to the spontaneous formation of groups of exploring particles. However, they noted that, to meet a given error criterion, Lbest requires more iterations than Gbest.

B. Central Versus Distributed Population Communication

As discussed above, the present study compares two traditional particle swarm topologies, Lbest and Gbest.

It is important to bear in mind that while Lbest and Gbest are two venerable and well-known population topologies, they are by no means the only ones, or even the best ones [9]. They are extremes in terms of fluency of propagation through the population and as such represent opposite ideals; in one case new breakthroughs are shared with the entire population and adopted immediately and in the other case new problem solutions are shared locally and compete for adoption against solutions propagated from other parts of the population. Very many topological structures have been proposed, blending these two extremes or innovating in other ways.

The general question has to do with distributed versus centralized communication. Is it better to give all particles the very best information known at any time, or is it better to spread out the search and let individuals persuade each other, one by one?

In fact Lbest and Gbest are end-points of a near-continuum of communication topologies. Besides the large number of possible bidirectional, fully connected structures, particles can communicate through weighted edges, probabilistic edges, fuzzy edges, adaptive links based on an unlimited number of possible rules, and dynamic links of various types, there can be disconnected subpopulations, unidirectional connections—the list is vast. Furthermore, particles can communicate in a large number of ways through the links; in the standard particle swarm the best neighbor is selected, but the particle could use an average of all neighbors [9], or some subset of them, the individuals could employ any of the recombination methods used in evolution strategies [10], and so on.

C. Topology—Early Experiments

As has been noted, the first particle swarms used either Gbest or Lbest topologies. As early researchers were learning how to control the tendency for trajectories to explode by damping the velocity and applying some coefficients on the velocity terms, the question of communication topology did not receive much of a focus until the publication of Watts and Strogatz [11] landmark paper on the "small world" effect in network topologies. Kennedy [5] modified the social network topology of a particle swarm systematically, and concluded that small-world effects were not significant in a population of 20 particles, but needed a larger network. That paper noted that the effect of the topology seemed to vary with the function, and subsequently a number of researchers began looking at aspects of the topology.

Suganthan [12] developed the case that since the Lbest topology seemed better for exploring the search space while Gbest converged faster, it would be reasonable to begin the search with an Lbest lattice and slowly increase the size of the neighborhood until the population was fully connected—Gbest—by the end of the run. That same paper also reported on another kind of topology, where neighbors were defined by proximity in the search space and the number of neighbors was dynamically increased through the course of the run. The authors anecdotally reported some improvement using the neighborhood operator.

Besides Gbest and Lbest, Kennedy and Mendes [13] tested a "pyramid" topology, based on a 3-D wire-frame triangle; the star, with one central node exchanging influence with all others; a graph created with cliques and isolates, as an example of heterogeneity; von Neumann neighborhoods, with neighbors above, below, and on each side on a wrapped 2-D lattice and random graphs having various characteristics. They found that Gbest was ranked second in all topologies as ranked by median necessary iterations. However, they concluded that, in trials where an error criterion was met, Lbest took longer than Gbest to achieve the specified error, but it met the criterion more often.

Peram *et al.* [14] selected an interaction partner for a particle using a weighted Euclidean distance. They identified the particle with the highest fitness distance ratio (FDR) for each vector element, i.e., the ratio of the difference between the target particle's fitness and the neighbor's fitness to the distance between them in the search space on that dimension. FDR and Gbest were used to influence the particle. FDR ensured that the selected neighbor was a good performer and increased the probability that a particle interacted with a neighbor nearby in the search space. The new algorithm outperformed the standard PSO on a set of test functions.

Liang and Suganthan [15] created random subpopulations and occasionally randomized all the connections. Those researchers obtained good results, especially on multimodal problems, with subpopulation size $n = 3$, restructuring every five iterations.

Janson and Middendorf [16] arranged the particles in a dynamic hierarchy, with each particle's own previous success and that of the particle directly above it influencing it. When particles with better performance were moved up the hierarchy they had more effect on poorer particles. The result was improved performance on most of the test functions considered.

Clerc [17] developed a parameter-free particle swarm system called TRIBES, in which the topology, including the size of the population, evolved over time in response to performance feedback. The population was divided into subpopulations with their own order and structure. "Good" tribes were hypothesized to benefit by removal of their weakest member, as they already possessed good problem solutions and thus could afford to reduce their size; "bad" tribes, on the other hand, were thought to benefit by addition of a new, randomly generated member, increasing the possibility of improvement. In the context of the many modifications to the particle swarm that comprise the unique TRIBES paradigm, Clerc [17] reported good results on a number of test functions.

D. Topology—Niching

Interest in niching in genetic algorithms, for the solving of multimodal problems, brought localized topologies back to the foreground in particle swarm research. The nomenclature of niching comes from evolutionary theory, where species narrow their variation in order to fit to a particular optimum; in the social jargon of particle swarms it might be better discussed in terms of conformity and norm formation, where subpopulations of connected particles communicate and influence one another and eventually converge around optima.

Niching divides the swarm into subpopulations in order to identify and explore more than one optimal region simultaneously. Niching techniques in GA include crowding, fitness sharing, the sequential niche technique, and species conservation [18]. Generally, particle swarm implementations have used crowding and fitness sharing techniques. The Lbest topology naturally encourages parallel search and typically discovers multiple optima; because neighborhoods overlap the result is competition between solutions, with one propagating through the population and another solution being forgotten and lost.

Li [19] found that an Lbest particle swarm can operate as a niching algorithm because the pbest of each particle forms a stable network retaining the best positions found so far, while the particles explore the search space more broadly. Also, Li [19] concluded that the Lbest topology in a reasonably large population was able to locate dominant niches (optima) across the search space. This means that particles locate optima that are approximately equally good. However, if the aim of the algorithm is to locate optima that are less dominant as well, a nonoverlapping topology might be recommended.

Liu and Ma [18], in a survey of multimodal particle swarm techniques, maintained that an Lbest topology can provide comparable or better performance, and with more consistency, than niching PSO's with a fixed niche radius, and without the introduction of an undetermined parameter.

Crowding is an approach where less-fit population members are replaced by fitter offspring whilst preserving population size. Populations implementing multiniche crowding are able to converge simultaneously to multiple solutions by encouraging competition between individuals within the same locally optimal group. Parrott and Li [20] called their technique "speciation," where particles that are similar to one another were linked topologically, improving on a technique by Kennedy [21] called "stereotyping" that substituted cluster centroids as sources of influence. Li [19] found that the Lbest topology was able to induce more stable niching behavior.

Fitness sharing is intended to model environmental niches with limited resources by making individuals in the same niche share their fitness; it modifies the search landscape by reducing the payoff in densely populated regions. In practice, the sharing effect is implemented by reducing the fitness of each individual as a function of the number of similar individuals in the population. It is a penalty for conformity.

One of the earliest "fitness sharing" approaches was developed by Parsopoulos and Vrahatis [22]. The method stretches the fitness landscape to remove local optima. Considering a minimization problem, when the swarm converges on an optimum, the fitness of that position is stretched so that it becomes nonoptimal. Thus, the swarm will focus on other areas of the search space, leading to the identification of other solutions.

Parsopoulos and Vrahatis [23] modified the function-stretching approach through the application of two other techniques: 1) deflection and 2) repulsion. The former is another form of modification of the objective function which removes optima which have been already located. The repulsion technique modifies the PSO algorithm by introducing a repulsion force in an area surrounding the optima which have already been found. The combination of these techniques seems to improve the ability of the method to find multiple optima.

Brits *et al.* [24] used a technique of generating subswarms when a local optimum was discovered, called NichePSO. The main swarm explores widely without communication among particles, while the subswarms perform local search of a region.

Researchers investigating techniques for finding multiple optima have experimented with particle swarm topologies, extending the Lbest configuration to give even more independence to subpopulations and largely rejecting Gbest because of its inability to search beyond a single optimal region.

E. Dynamic Topologies

Several investigators have suggested adjusting the topology over time in order to capitalize on the strengths of the two approaches.

Clerc continued to experiment with dynamic topologies in the years following the intriguing TRIBES algorithm; these explorations culminated in a putative baseline PSO known as SPSO-2011 [25]. The algorithm contained a novel form of particle dynamics, and more relevant to this review, an adaptive and random particle communication strategy of great interest. The topology, identical to the 2006 version of the algorithm [26], is modified at the end of each unproductive iteration by demanding that each particle "informs" K particles at random, as well as itself. The result is a heterogeneous communication graph of degree varying between 1 and the swarm size (although, on average, a particle is informed by K others) which harmonizes Lbest and Gbest qualities. However, fully connected particles are rare. Although not intended as a competitive PSO, and in fact not performing well on the challenging CEC 2013 benchmark composition functions, SPSO-2011 showed evidence of scalability to higher dimension [27].

Suganthan's early experiments [12] have already been noted. Bonyadi *et al.* [28] developed a time-adaptive topology, based on the observation that Lbest neighborhoods explore the search space more widely for optimal regions, while Gbest tends to find a good point within an optimal region. Thus, they proposed to change the topology over time from several small subswarms to the fully connected Gbest topology. They argued that subswarms with as few as two members explore the search space very thoroughly, while Gbest more effectively exploits the better quality regions. They started with small subswarms and increased their size linearly through the run until the entire population was fully connected.

Marinakis and Marinaki [29] implemented an expanding neighborhood topology and in a similar vein, Lim and Isa [30] decided to balance the particle swarms' preference for exploration or exploitation by varying the particles' connectivity with time in a version they called PSO with increasing topology connectivity (PSO-ITC). They found that high connectivity topologies favor simple problems whereas Lbest topologies perform better in complex problems.

Each particle in PSO-ITC is initially connected with one neighbor that is randomly selected from the population. As the optimization process evolves, the ITC module gradually increases the particle's topology connectivity by randomly selecting new neighbors until all particles are fully connected. Interestingly, connections in PSO-ITC are unidirectional; A may influence B without B influencing A. If a particle fails to improve for a criterion number of evaluations, then new neighbors are randomly selected. The algorithm incorporates many changes, as well, that are outside the scope of this discussion.

All the cited papers in this review are in agreement that the communication topology of the particle swarm affects its performance. In comparing the two "classical" topologies, most note that Lbest has a tendency for exploration, for finding good regions of the search space and searching in parallel to discover multiple regions, while Gbest excels at exploitation, at finding the highest quality point in a good region.

F. Rise of Gbest

Despite the early advice, a great majority of published PSO research uses Gbest swarms. It is difficult to point to a time when the research paradigm began tipping in favor of Gbest, because the phenomenon is marked more by the absence of language than any presence. Over time, through the middle 2000s, many writers stopped referring to population topology at all, and only described the algorithm in terms of a "personal best" and a Gbest or "population best" term in the velocity formula.

It was simply assumed that the algorithm used the population's best solution to influence all members. For example, the 2015 Proceedings of the Congress on Evolutionary Computation contained 36 papers with particle swarm or "PSO" in the title. Three of the papers described hierarchical or "subswarm" structures that are not relevant to the current discussion. Of the remaining 33 papers, 29, or 91% used the Gbest topology exclusively. This measure is representative of the trend in the research.

The continued exclusive use of the Gbest communication topology in trials of new PSO variants is perplexing given the considerable evidence that Lbest has greater search capacity. Three reasons can be seen immediately. The Gbest PSO, compared to, say, a gradient descent algorithm, is easy and quick to code. The Lbest ring topology is hardly any more difficult to implement, but other Lbest networks are intricate and require careful coding. Gbest performs "well enough," and since it is the simplest algorithm, it seems a natural choice. And Gbest runs faster due to fewer look-ups. The speed difference is not great but might become significant in some applications. Gbest, therefore, is simple to understand and code, has acceptable performance and executes efficiently.

We note, also, that very many papers that assume Gbest as orthodoxy also go on to mention the particle swarm's "well known tendency to converge prematurely." For example, this view point is clearly stated in Pant *et al.*'s [31] study in 2008, Van den Bergh and Engelbrecht's [32] 2010 paper on PSO convergence, and Han and Wang's [33] 2013 paper.

At present, because of the preponderance of publications that literally define the particle swarm in terms of the Gbest topology, and because much of the easily available software implements Gbest with no alternative, we presume that many new researchers are unaware that there is any other way to implement the algorithm.

G. Engelbrecht's [3] Study

Engelbrecht [3] found, from a survey of fifteen papers published between 2000 and 2013, a general recommendation to use Gbest on unimodal problems and to otherwise employ Lbest. However, he concluded, on the basis of an empirical investigation on 60 hand-picked test functions.

1) Neither algorithm is preferred for unimodal, multimodal, separable, or nonseparable problems.
2) In terms of accuracy (mean error), G and Lbest are approximately equivalent.
3) Gbest is slightly more consistent (smaller standard deviation in the error).

These findings allow him to speculate the following.

1) The general recommendation is in error: both Gbest and Lbest must be tested in order to find the optimal PSO for an arbitrary problem.
2) Either Gbest or Lbest could be chosen in order to assess the effect of a specific change to PSO dynamics.

Speculation 1) is a generalization of 1 to all functions and speculation 2) is strictly unsupported since the interaction between particle dynamics and particle communication is not investigated.

Engelbrecht's [3] results are intriguing but his methods are somewhat opaque. Conclusions 1 and 2 are particularly surprising since they question the orthodox view that has emerged over the past 20 years of PSO research. Therefore, it seems worthwhile to replicate his results.

III. Empirical Study—Combined Benchmark

This section forms the central part of this paper: the relative performance of Gbest and Lbest on a comprehensive benchmark.

A. Combined Benchmark

A set of 25 single-objective optimization test functions was assembled for a competition at the 2005 Congress on Computational Intelligence. This benchmark consists of five unimodal functions, seven basic multimodal functions, two multimodal complex functions, and 11 hybrid functions, which were weighted sums of ten basic test problems [34]. This test suite became the standard for comparing algorithms, and in subsequent years possible improvements were identified, resulting in a new benchmark suite being developed for CEC 2013 [35]. This collection of 28 functions expanded on the 2005 composite functions and added new test problems. It includes five unimodal functions, 15 basic multimodal functions, and eight composition functions, and has become the more recent default standard for testing and comparing optimization algorithms.

Engelbrecht's [3] 2013 study used a benchmark of 37 base functions, plus rotated, rotated and shifted, and noisy versions of some of these base functions, totaling 60 problems in all.

The combined benchmark (CEC05 + CEC13 + ENG13), Table S1 (given in the supplementary material), provides 84 unique problems. Four CEC05 functions, F01, F03, F09, and F10 have equivalent versions in the CEC13 set. These functions were not included. CEC05 F06–F08, however, are nonrotated, unbounded, and optimum-on-bounds versions of CEC13 F06–F08, respectively: these functions were retained on account of their important differences. The ENG13 benchmark has several overlaps with CEC 2005 and includes a duplicate. The overlaps and duplicate were removed from our combined test set, and two functions were added—shifted and rotated versions of a couple of base functions. The functions are numbered f_{02-25} (CEC05), f_{31-58} (CEC13), and f_{61-95} (ENG13).

Table S1 in the supplementary material lists the presence of 12 independent binary function properties, as reported in the original documentation [3], [34], [35]. Fourteen properties are annotated, but two are dependent: a function is either unimodal or multimodal, and is either separable or nonseparable. The total number of functions with a given property is given in the second column of Table S1 in the supplementary material. The absence of a property in Table S1 in the supplementary material does not indicate that a function does not have that property, it merely indicates that the documentation is silent on the matter. The published references omit some property definitions and hence we cannot confidently make any inference—for example, the CEC05 composition functions appear, judged by the CEC13 composition functions, to be asymmetric.

B. Methodology

The canonical PSO algorithm as described in Section II-A was used for all experiments. A single fully connected topology (Gbest) and the ring topology (Lbest) were trialed for each function of the benchmark. Particles were initialized uniformly at random in each function's search space X for each run. Particles were allowed to move outside X but were not evaluated. The pbest update was applied immediately after a particle moves and not at the end of an iteration (the particle with the highest index is chosen if there were ties within the neighborhood).

Each algorithm (Gbest and Lbest) was run 50 times on each test function. Runs were terminated at 150 000 function evaluations (evals), where a function evaluation is the number of times $x_i, i = 1, \dots M$, is evaluated, even if $v_i(t) = 0$.

In order to be commensurate with Engelbrecht's [3] BRICS paper, the swarm has $M = 30$ particles and $w = 0.729844$ and $c = 1.49618$. The initial condition $v = 0$ is also taken from the BRICS paper. The spatial dimension, constant across runs, was $D = 30$.

Three measurements were made: 1) best function value, known here as the error; 2) the swarm diversity; and 3) the eval of the last improvement in the run. The actual error (best value minus f^*) was recorded for CEC05 and CEC13. The optimum value f^* is not known for some ENG13 test functions: in these cases, the best function value was recorded.

The diversity, div, defined here and in [3] as the mean separation between any two particle pbests is given by

$$\text{div} = \frac{M(M-1)}{2} \sum_{i=0}^{M-2} \sum_{j=i+1}^{M-1} |p_i - p_j|. \quad (2)$$

Each run was considered as a single trial. Analysis of the 84 000 trials (84 functions * 50 runs * 2 algorithms) was grouped by function and by function property.

For tests of aggregated function data, the three measurements were ranked per function to make data commensurate across functions. Wilcoxon nonparametric tests of significance ($p < 0.05$) were considered more appropriate that t-tests due to non-normality of the raw data, as well as of transformed ranks.

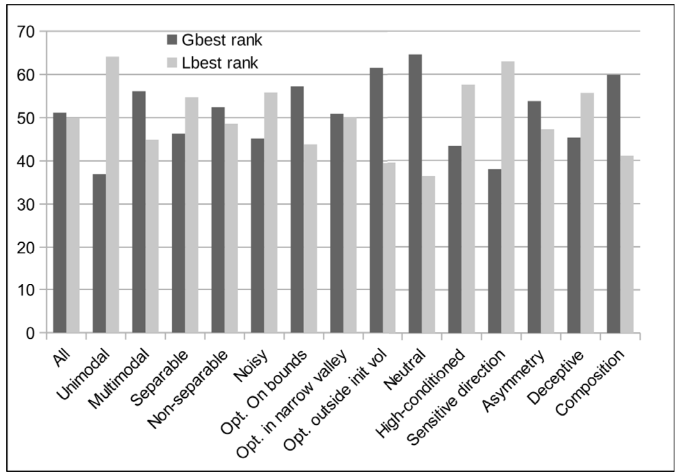

Fig. 1. Gbest and Lbest mean error rank for each function property. All ranks are significant at $p < 0.05$ except for *narrow-valley*. A lower rank indicates better performance.

C. Results

The result of Wilcoxon analysis on the trials on each function is displayed in Tables S2 and S3 in the supplementary material. This method of analysis obviates data normalization and imposition of an arbitrary criterion of success. Bold type indicates significance, $p < 0.05$. The number of "wins" and mean rank is reported in Table I where a win is: a significantly lower function error, a significantly higher last improvement or a significantly higher terminal diversity. The mean error ranks are charted in Fig. 1.

Gbest has better error performance than Lbest on 23 functions, Lbest is superior on 37 functions, and the algorithms are not significantly different on the remaining 24 functions. Lbest therefore performs no worse than Gbest on 73% of the functions in the benchmark. Lbest ranks higher than Gbest, although the difference is small. Lbest improves later in the run in 39 cases (Gbest improves later in 15 functions), and has a higher terminal diversity in 71 functions (Gbest is more diverse in 8). The rankings are again significant, and by a large difference.

Lbest, as tested on this comprehensive benchmark, is the better algorithm (lower error). Furthermore, Lbest achieves a higher diversity and is able to improve at later stages of the optimization than Gbest. A reasonable inference from these results is that Lbest searches longer and more widely and, in consequence, finds better solutions.

We find significant differences in error between the algorithms for all properties except narrow-valley; all properties except noise are pertinent with regards to last improvement and, apart from unbounded and sensitive-direction, we find that the property groups yield significant run results.

The algorithms have better ranked errors in six (Gbest) and seven (Lbest) properties. Lbest improves later in all function groups except narrow valley and high conditioned, and has a higher terminal diversity in all groups except opt. outside bounds.

Unimodality and separability indicate simpler problems in the sense that separable problems can be optimized one dimension at a time and unimodal functions can be optimized by any gradient following algorithm. On the other hand, multimodality, asymmetry, and composition represent more difficult functions due to the possibility of entrapment in local optima, the nonuniformity between dimensions, and, in the case of what is arguably the hardest of all properties, composition, a mixture of different characteristics leading to very high modality, optima in funnels, and very marked anisotropy.

Gbest is the preferred algorithm for the simpler problems: unimodal (15 Gbest wins and three Lbest wins) and separable (7-5).

Lbest is the better algorithm for problems featuring each of these more challenging properties. Lbest is a far better optimizer of composition functions (15-1) and better at multimodal (34-8), nonseparable (32-16), and asymmetric (11-4) functions. In each case, Lbest improves later in the run, and maintains a higher diversity.

Neutrality is present in eight functions and Lbest has better error in seven of these, Gbest in none. This result indicates that Gbest has problems transversing flat regions, a finding that is compatible with small diversity.

The correlation between larger diversity, larger improvement, and lower error does not hold for the simpler properties of unimodality and separability. The correlation also does not hold for the deception functions although there are only two deceptive functions in the set and results might be misleading due to interactions with other properties.

However, the proposition that larger diversity and later improvement implies lower error, is true for nonseparable functions, those functions with the global optimum on the bounds of the search space, functions with plateaus (neutral areas), asymmetric, and composition functions. These properties are commonly thought to indicate optimization severity.

Conversely, whenever Lbest is the better algorithm by error, except for opt. outside bounds, it also has later improvement and higher diversity. Gbest might or might not have later improvement if it is the better. It is never the more diverse.

D. Conclusion

The results show that Lbest is the better algorithm for harder functions and indicates that Lbest achieves its success by being more diverse and improving for longer. Easier functions, those that are separable and unimodal, are optimized more efficiently by the Gbest topology.

Regarding modality, we note that our results are consistent with Engelbrecht's [3] 2013 study. He reported that Gbest performed better on 11 unimodal functions compared to Lbest performing better on 3, and that Gbest performed better on nine multimodal functions to Lbest's 19, and concluded, "the empirical analysis of this paper provides convincing support in favor of Gbest PSO even for multimodal and nonseparable problems." We find the same pattern but conclude the opposite. In sum, Lbest performs better than Gbest, especially on harder problems, because it retains its population diversity and continues to search longer.

IV. Stagnation Analysis

A comparison of last improvements indicates algorithm activity late in a run. Improvements, however, might be minute adjustments in the vicinity of a discovered optimum. Late improvements, if tiny, are inconsequential. In other words, what matters is the lateness of the last *significant* improvement. This section considers how significant improvement and the related concept of stagnation can be quantified and reports on G/Lbest measurement of these quantities.

A. Defining Improvement and Stagnation

Any attempt to quantify the last significant improvement (LSI) will involve arbitrary decisions on the nature of f. For example, a significant improvement for a flat function might be trivial for a very hilly function. Despite this arbitrariness, differences in logarithmic, rather than absolute, values seem cogent since is it is known that PSO converges exponentially on symmetric unimodal problems such as the sphere function [36], and it is desirable for PSO convergence on such minima to be significant at any stage in a run. Furthermore, a logarithmic sensitivity would ensure that exploration of fine detail at lower function value will count as significant.

A simple single-parameter measure of LSI-based solely on the observation of the history of best values $f(t)$ at each evaluation t can therefore be defined as follows: an improvement is regarded as significant at level s if

$$\left|\log\left(\frac{f(t+1)}{f(t)}\right)\right| \geq s \times \log\left(\frac{\mathbb{E}[f(0)]}{f_{\min}}\right) \tag{3}$$

where $\mathbb{E}[f(0)]$ is the expected best initial value of the population and $f_{\min}$ is the least value of f over a batch of trials. An improvement is therefore significant if $\log(f)$ changes by more than a fraction s of the expected overall logarithmic change.

The applicability of this definition of LSI can be clarified by considering PSO progress inside a symmetric basin with, say, the optimum at 0: PSO converges to 0 as $x(t) = x(0)e^{-\alpha t}$ [36]. From (3), an improvement on the threshold of significance satisfies

$$\frac{f(x(t+1))}{f(x(t))} = \frac{f(e^{-\alpha}x(t))}{f(x(t))} = \left(\frac{\mathbb{E}[f(0)]}{f_{\min}}\right)^{s} = \text{const.}$$

The underlying assumption on the nature of f is therefore that f is homogenous close to each optimum: $f(e^{-\alpha}x) = \text{const} \times f(x)$, i.e., f, in the vicinity of its optima, is a polynomial of homogeneous degree. This condition will be true for all continuous functions.

An algorithm is not expected to improve at every evaluation of a run. However, a satisfactory algorithm would be expected to improve over an evaluation interval in an appreciable number of runs. Suppose that a run is regarded as *s-stagnant* in an evaluation interval I if there is no s-level improvement in that interval. A stagnation probability can be defined as the proportion of runs that are s-stagnant in I. An algorithm can then be said to be improving or nonstagnant at level s in I with probability at least p if the stagnation probability is less than or equal to $1 - p$. We take, in the following, $p = 0.5$, so that an improving algorithm on a given interval with significance level s does not stagnate on at least half of all runs.

B. Results—Convergence Plots

Convergence plots of the trials reported in the previous section are plotted in Figs. S1–S84 in the supplementary material. The plots show the logarithm of the error as a function of evaluation t for 50 runs of Gbest and Lbest. Only a summary of the vast amount of information contained in these plots can be given here. The main features, split according to modality and separability are as follows.

The unimodal separable functions (U, S), $f = \{31, 35, 61, 67, 68, 72, 77, 78, 89, 90\}$, are the easiest optimization class of the benchmark since they correspond to D 1-D unimodal problems. All plots show a power law f dependence except for f_{78}, a noisy function.

The nonseparable unimodal (U, NS) functions, $f = \{2, 4, 5, 32–34, 73, 76, 85–87, 91\}$, are more challenging than the separable unimodal functions since variables interact and the problem cannot be decomposed into subproblems. The convergence plots show a mix of behaviors: power law fall-offs and stagnant runs, as indicated by horizontal regions.

Multimodal separable functions (M, S), $f = \{41, 52, 65, 74, 79, 92\text{-}94, 95\}$, are arguably the next hardest; decomposition into subproblems is possible, but subproblems may have suboptimal minima which can potentially trap optimizers. Once more the convergence curves show either exponential convergence on an optimum or periods of stagnation with jumps. Some plots show stagnant runs at different function values—parallel horizontal lines—indicating entrapment in local optima. In other cases, the algorithm chosen to optimize a single optimum and exponential convergence is displayed.

The nonseparable multimodal functions (M, NS), $f = \{6, 7, 8, 11–25, 36–40, 42–51, 53–58, 62–66, 69, 70, 71, 75, 80–84, 88\}$, form the largest class. Apart from f_{62-64}, which shows Lbest exponential convergence, the plots display rapid decrease followed by long periods of stagnation. The stagnant periods may be interspersed with jumps.

C. Results—LSI

Histograms of numbers of LSI's per designated interval essentially provide coarse-grained descriptions of the convergence plots.

The LSI at level $s = 0.001$ frequency histograms were computed by counting the LSI's in bins of width 37 500 (Figs. S1–S84 in the supplementary material). The intervals can be conveniently labeled $I_i \equiv 37500 \times [i-1, i]$ for $i = 1$-4. In order for the logarithm to be defined, function values for ENG13 functions with $f_{\min} < 0$ have been shifted up by $|f_{\min}|$ and zero values have been excluded. Logs were then taken to base 10.

The majority (7 out of 10) of the U, S histograms consist of a single bar at I_4, indicating improvement to termination, or a single bar at an earlier interval, corresponding to convergence within the bounds of finite precision arithmetic. 7 of the 12 U, NS histograms are single barred; the lower proportion indicates the increased severity of this class. The center of mass (CM) of the Lbest M, S histograms is predominantly to the right of the CM of the corresponding Gbest histograms (the sole exception if f_{92}). The M, NS histograms show a similar relative Lbest CM bias (the situation for f's $6, 21, 46, 75$, and 85 is not so clear cut). This right bias is indicative of Lbest's ability to significantly improve later in the run.

In summary, the main features of the convergence plots are represented by the histogram distributions, validating the choice of $s = 10^{-3}$.

Tables S5 and S6 in the supplementary material provide the mean LSI, the number of runs with improvements in I_4 (the algorithm is nonstagnant if this number is at least 25) and which algorithm, if either, had significant (Wilcoxon, $p < 0.05$) later LSI's, for $s = 10^{-5}$, a very weak criterion, corresponding to minute improvements, $s = 10^{-3}$, a moderate criterion, and a very tough criterion, 10^{-1}, equivalent to very large jumps. $s = 10^{-7}, 10^{-1}$ were chosen as extremes; 10^{-3} corresponds to the smallest noticeable jumps on the convergence plots.

These tables are summarized in Table II. As expected, stagnation increases from $s = 10^{-7}$ to $s = 10^{-1}$ for either algorithm. Lbest is evidently the later improver at all three levels. In particular, Lbest improves later in 66 functions at $s = 10^{-3}$ compared to Gbest's 7.

Nonstagnant runs in I_4 will presumably continue to improve beyond termination. For example, although Lbest is the weaker, in terms of error, algorithm for f_{67}, Fig. S56 in the supplementary material suggests that the poorer performance is a matter of run duration. Lbest is simply *slower*.

Focusing on $s = 10^{-3}$, we find that Lbest is nonstagnant in I_4 in 13 of the 22 functions with better Gbest error (59%) and Gbest is nonstagnant on 5 of Lbest's 37 preferred functions (14%). Assuming that a nonstagnant algorithm will continue to improve, Lbest is stuck on only $22-13 = 9$ Gbest-preferred functions whereas Gbest is stuck on $37 - 5 = 33$ Lbest-preferred functions. Lbest is either equivalent (tied error), better (lower error than Gbest), or slower (higher error than Gbest, but significantly improving) on $84 - 9 = 75$ functions.

D. Conclusion

On the combined benchmark of 84 functions: Lbest is the later significant improver, provides a lower error and is more diverse at termination. Choosing Lbest above Gbest would not be detrimental for $84 - 13 = 71$ or 85% of problems with an evaluation budget of 1.5×10^5, and for larger budgets, the stagnation analysis suggests the number could rise to 75 (89%).

V. Mobility Analysis

Although LSI is a guide to an algorithm's rate of convergence on a function, it fails to identify an important property: the ability to discover new optima. Improvements might be significant even if the algorithm is converging on a suboptimal point. Ideally, we wish to measure *mobility*, i.e., the ability to jump between optima.

A. Defining Mobility

Suppose, borrowing the language of dynamical systems, a downhill basin of attraction $\mathcal{B}$ of an attractor x^* (the basin's

TABLE I

COMPARISON OF FUNCTION PROPERTIES. THE TABLE SHOWS THE NUMBER OF WINS FOR EACH MEASUREMENT AND THE MEAN RANK OVER EACH OF THE FOURTEEN FUNCTION PROPERTIES. A WIN ON A TEST FUNCTION IS DEFINED AS LOWER ERROR, AND HIGHER LAST IMPROVEMENT AND TERMINAL DIVERSITY. SIGNIFICANCE WAS DETERMINED BY A WILCOXON RANKED TEST. THE MEAN RANKS WERE DETERMINED BY A WILCOXON TEST ON THE POOLED RANKS OF EACH FUNCTION IN A GIVEN PROPERTY CLASS. BOLD TYPE INDICATES SIGNIFICANCE AT THE 0.05 LEVEL

		Error				Last Improvement				Diversity			
		Gbest		Lbest		Gbest		Lbest		Gbest		Lbest	
Property	**Total**	**#Wins**	**Mean rank**	**#Wins**	**Mean rank**	**#Wins**	**Mean rank**	**#Wins**	**Mean rank**	**#Wins**	**Mean rank**	**#Wins**	**Mean rank**
All	84	23	51.1	37	**49.9**	15	43.3	39	**57.7**	8	33.3	71	**67.7**
Unimodal	22	15	**36.9**	3	64.1	6	48.6	5	**52.4**	4	35.2	18	**65.8**
Multimodal	62	8	56.1	34	**44.9**	9	41.4	34	**59.6**	4	32.6	54	**68.4**
Separable	18	7	**46.3**	5	54.7	3	39.8	11	**61.2**	5	39.4	13	**61.6**
Non-separable	66	16	52.4	32	**48.6**	12	44.3	28	**56.7**	3	31.7	59	**69.4**
Noisy	3	1	**45.2**	1	55.8	1	51.4	0	49.7	0	25.5	3	**75.5**
Opt. on bounds	4	0	57.2	2	**43.8**	0	46.5	1	**54.5**	0	31.4	3	**69.6**
Opt. in narrow valley	3	1	50.9	2	50.1	2	**55.5**	0	45.5	0	36.6	3	**64.4**
Opt. outside init vol	2	0	61.5	1	**39.5**	0	38.7	1	**62.3**	1	49.9	1	51.1
Neutral	8	0	64.6	7	**36.4**	0	43.5	3	**57.5**	0	31.7	8	**69.3**
High conditioned	2	1	**43.4**	1	57.6	2	**62.6**	0	38.4	0	25.7	2	**75.3**
Sensitive direction	2	1	**38.0**	0	63.0	0	38.4	1	**62.6**	1	46.6	1	54.4
Asymmetry	20	4	53.8	11	**47.2**	3	42.8	9	**58.2**	0	26.9	19	**74.1**
Deceptive	2	1	**45.3**	0	55.7	0	34.2	2	**66.8**	0	25.5	2	**75.5**
Composition	19	1	59.9	15	**41.1**	2	45.3	6	**55.7**	0	31.8	16	**69.2**

TABLE II

SUMMARY OF LSI STUDY. THE TABLE SHOWS THE NUMBER OF FUNCTIONS WHERE EITHER ALGORITHM WAS NONSTAGNANT AT LEVEL s IN THE FINAL 25% OF THE AVAILABLE EVALUATIONS, AND THE NUMBER OF FUNCTIONS WITH SIGNIFICANTLY ($p < 0.05$) LATER LSI. THE STATISTICS ARE GROUPED BY ERROR PERFORMANCE

	s = 1E-5				1E-3				1E-1			
	Gbest		Lbest		Gbest		Lbest		Gbest		Lbest	
	non-stagnant	**later**	**non-stagnant**	**later**	**non-stagnant**	**later**	**non-stagnant**	**later**	**non-stagnant**	later	**non-stagnant**	**later**
G better error	14	2	16	7	11	7	13	9	0	9	0	1
L better error	4	0	11	32	0	0	5	35	0	8	0	20
Neither better	10	1	13	20	1	0	8	22	0	7	0	6
Σ	28	3	40	59	12	7	26	66	0	24	0	27

optimum position) is the set of points such that any downhill path starting at a point x in $\mathcal{B}$ will inevitably lead to x^*. This definition fits with the action of non gradient optimizers such as PSO and enables, in simple cases such as 2-CONES (defined below) the tracking of an algorithms's progress between basins. The relative mobility of one algorithm with another can be gauged by the number of jumps between basins.

The 2-CONES function

$$f_{2C}(x) = \min(m_A|x - x_A| + d_A, m_B|x - x_B| + d_B)$$

where cones A and B have depths d_A, d_B and positive gradients m_A and m_B, is a bimodal version of the more general cone landscape [37], [38].

Define $L = |x_B - x_A|$ and $d = d_B - d_A \geq 0$ (so that the global optimum is at x_A). The equal value contours from the two cones meet on the line joining the centers at a distance

$$r_A = \frac{d + m_B L}{m_A + m_B}$$

from the global optimum x_A. A point x is therefore in $\mathcal{B}_A$ if $|x - x_A| < r_A$, and in $\mathcal{B}_B$ if $|x - x_B| < L - r_A \equiv r_B$. The difficulty of an instance of 2-CONES, as perceived by an optimizer, can be controlled with the parameters $m_{A,B}$, d and L. A simple difficulty measure, ρ can be defined as the ratio $(r_A/r_B)^D$ of optimal to suboptimal basin volumes. Smaller relative volumes indicate harder problems.

B. Experiments

A series of experiments was performed with seven 2-CONES instances. The gradients, m_A, of the optimal cone ranged from 0.85 to 1.15. Otherwise, $m_B = 1, d_A = -450$ and $d = 1$. The optima x_A, x_B were placed symmetrically either side of O on the diagonal $\mathbf{1}^D$ at a separation of $L = 500$. The search space $X = [-1000, 1000]^D$; particles were initialized in $X \setminus (\mathcal{B}_A \cup \mathcal{B}_B)$. Error, diversity, the number of jumps between basins, the eval of the last jump, and the basin of the best position on each run were recorded for 1000 runs of each algorithm on each function instance. The number of runs is far higher than is customary in order to improve the significance of the nominal measurements.

C. Results

The results are tabulated in Table S7 (given in the supplementary material) and Table III.

The tables report on one degree of freedom chi-square tests of success at finding the optimal basin. The chi-square test for independence calculates the expected number of cell observations in a table of nominal-scale data based on row and column sums. The probability of obtaining the observed frequencies, given expectations is then estimated. A significant ($p < 0.05$) outcome suggests that there is dependency between the row

TABLE III

2-CONES. MEAN NUMBER OF BASIN JUMPS AND NUMBER OF RUNS IN THE BATCH OF 1000 THAT ENDED WITH THE BEST FOUND POSITION IN THE OPTIMAL BASIN. THE TABLE ALSO SHOWS THE RELATIVE VOLUME OF THE OPTIMAL BASIN TO THE SUBOPTIMAL BASIN. BOLD TYPE SIGNIFIES SIGNIFICANCE ($p < 0.05$) IN WILCOXON (JUMPS) AND χ-SQUARED (OPTIMAL BASIN) TESTS

		Jumps		**Optimal basin**	
$\mathbf{m_A}$	**rel. vol**	**Gbest**	**Lbest**	**Gbest**	**Lbest**
1.15E+00	1.69E-02	1.00E+00 ± 0.00E+00	**1.06E+00 ± 8.30E-01**	11	17
1.10E+00	6.43E-02	1.00E+00 ± 0.00E+00	**1.44E+00 ± 2.79E+00**	58	65
1.05E+00	2.60E-01	1.00E+00 ± 0.00E+00	**2.16E+00 ± 4.15E+00**	**152**	**273**
1.00E+00	1.13E+00	1.00E+00 ± 0.00E+00	**3.14E+00 ± 5.39E+00**	**527**	**733**
9.50E-01	5.27E+00	1.00E+00 ± 0.00E+00	**2.32E+00 ± 4.54E+00**	**844**	**965**
9.00E-01	2.68E+01	1.00E+00 ± 0.00E+00	**1.38E+00 ± 2.32E+00**	**967**	**996**
8.50E-01	1.49E+02	1.00E+00 ± 0.00E+00	**1.06E+00 ± 7.91E-01**	**992**	**1000**

and column variables. In this case, populations were tabulated for each cones environment, dichotomized by whether they succeeded or failed at finding the optimal basin, by topology and bold type in the tables indicates significance at $p < 0.5$.

The error, for Gbest and Lbest decreases as ρ increases, confirming that a relatively larger optimal basin is easier to optimize. Lbest is significantly better in each case and has a later final basin jump.

The diversity distribution, in each instance, was unimodal for Gbest with near-zero variance but Lbest had a marked bimodal distribution that extended both below and above Gbest's values. The great difference in the distributions means that nonparametric group tests are problematic. Instead, nominal significance tests were employed. The bimodal Lbest distributions were characterized by a tight cluster in the interval $[10^{-13}, 10^{-12}]$ and a second subdistribution with diversities > 1. There were no diversities in $[10^{-12}, 1]$ for any function instance. Nominal categories "low" (div $< 10^{-12}$) and "high" (div > 100) were chosen. Low diversity indicates a very tight swarm, and high diversity means that the swarm is spread over a distance of 20% of the separation between the optima. Table S7 in the supplementary material shows the far higher diversity of the Lbest swarm.

Gbest jumped just once—from the initialization region to either $\mathcal{B}_A$ or $\mathcal{B}_B$—in every run of every cones instance. Lbest, on the other hand, was found to be more mobile, with frequent jumps between $\mathcal{B}_A$ and $\mathcal{B}_B$. The mobility was at a maximum when the basins were near equal volume ($\rho = 1.13$), and fell away for decreasing ρ, presumably due to the increased difficulty of finding the optimum basin. Mobility also decreased for ρ increasing beyond 1.13 since jumps from A are disfavored due to the smaller size of B, and indeed become algorithmically impossible when function value falls below $f(x_B)$. Lbest finds the optimal basin significantly more often in the five easier functions.

D. Conclusion

Lbest performs better on 2-CONES, a simple bimodal function. This performance enhancement is correlated with increased mobility and diversity. Lbest finds the optimal basin more often and jumps later in the run. This experiment pins down a key property of Lbest swarms.

VI. REAL WORLD PROBLEMS

A small number of global optimization problems deriving from problems in the human domain were also tested. These functions have been chosen for their diversity of origin and structure: the FM synth problem (FM), originating in the design of virtual electronic instruments, is low-dimensional and highly multimodal [39]; the design of a gear train (GT), an integer programming problem, is also low-dimensional [40]; the 3-D configuration of molecules as modeled by the funnel structured Lennard Jones (LJ) potential [41]; the control of a continuous stirred tank reactor, arising in the control of industrial chemical mixing, is a system of ordinary differential equations [39]; the spread spectrum radar problem is an NP-hard minimax electrical engineering problem (but can be rewritten as a system with constraints) [39] and the reconstruction of images by tomography (TR), an important inverse problem in medical and industrial imaging, is an underdetermined system, which implies that the global optimum is degenerate [42]. The dimensionality of the problems ranges from 4 to 100 and the global minimum is sought in all cases.

TR is a novel test function in this context. It is constructed from a downsampled standard test image, the Shepp–Logan [43] image phantom. In this case, the phantom was downsampled to a 10 by 10 matrix, y_{ij}. The downsampling is necessary in order to control the high dimensionality of the problem. The aim is to reconstruct y from knowledge of the row and column sums, $r_i(y) = \sum_j y_{ij}, c_i = \sum_i y_{ij}$. Suppose that x_{ij} is a trial image. Then the objective function is $||r(x) - r(y)||^2 + ||c(x) - c(y)||^2$. A 12 atom LJ problem was chosen so that the dimensionality, 30, is commensurate with the combined benchmark.

Table S4 in the supplementary material displays the results of Gbest and Lbest trials using the experiment design of Section III-B. Lbest achieves a significantly lower function value on FM Synth, GT, and TR. The algorithms perform equally well on the remaining three. Lbest has the higher terminal diversity throughout the real world test set and improves later on three functions. Interestingly, Lbest is superior on the lower and higher dimensioned problems.

This test set is too small and the problems are too structurally diverse for any firm conclusion, but the indication is that Gbest is not the first choice PSO algorithm for complex real world optimization.

TABLE IV
RESULTS OF Lk VERSUS GBEST WILCOXON ($p < 0.05$) ERROR TRIALS FOR THE COMBINED BENCHMARK. COLUMNS HEADED L2–L16 SHOW THE SIGNIFICANTLY LOWER ERROR RANKED ALGORITHM ON EACH FUNCTION. THE Lk ALGORITHM WITH THE LME IS TABULATED—OLDEN TEXT INDICATE THAT THE RANKING OF ERRORS DIFFERED IN AT LEAST ONE GROUP IN A KW ($p < 0.05$) FOUR-WAY TEST

f	L2	L4	L8	L16	LME	f	L2	L4	L8	L16	LME	f	L2	L4	L8	L16	LME	f	L2	L4	L8	L16	LME
2	G	G	G	G	**L16**	92	G	G	=	=	**L16**	49	L2	L4	L8	L16	**L16**	11	=	L4	L8	L16	**L8**
4	G	G	L8	L16	**L16**	94	G	G	=	=	**L8**	51	L2	L4	=	=	**L2**	12	=	=	=	=	L2
6	G	G	G	G	L2	15	L2	L4	L8	L16	L4	54	L2	L4	L8	L16	L8	13	=	=	L8	L16	**L16**
32	G	G	=	=	**L8**	16	L2	L4	L8	L16	L4	55	L2	L4	L8	L16	L16	14	=	=	L8	=	**L8**
34	G	G	G	=	**L16**	18	L2	L4	L8	L16	L4	56	L2	L4	L8	L16	**L2**	17	=	L4	L8	L16	**L4**
38	G	G	G	G	L16	19	L2	L4	L8	L16	L2	58	L2	L4	=	=	**L2**	33	=	=	=	L16	**L16**
44	G	G	=	L16	**L16**	20	L2	L4	L8	L16	**L4**	62	L2	L4	L8	L16	**L2**	35	=	=	=	=	L4
52	G	G	=	L16	**L16**	21	L2	L4	L8	L16	**L4**	63	L2	L4	L8	L16	**L2**	39	=	=	=	=	L4
61	G	G	L8	L16	**L16**	22	L2	L4	L8	L16	**L8**	64	L2	L4	L8	L16	L4	40	=	G	=	=	L2
67	G	G	G	G	**L16**	23	L2	L4	L8	L16	**L4**	69	L2	L4	L8	L16	**L2**	45	=	L4	L8	=	L8
72	G	G	G	G	**L16**	24	L2	L4	L8	L16	L2	70	L2	L4	L8	L16	**L2**	50	=	L4	L8	L16	L16
73	G	=	=	=	**L4**	25	L2	L4	L8	L16	**L4**	71	L2	L4	=	=	**L2**	53	=	L4	L8	=	**L4**
76	G	G	G	G	**L16**	31	L2	L4	L8	L16	L8	78	L2	L4	L8	L16	**L8**	57	=	=	L8	L16	**L16**
77	G	G	G	G	**L16**	36	L2	G	=	=	**L4**	80	L2	L4	L8	L16	**L8**	65	=	=	=	=	L8
81	G	G	G	G	L8	37	L2	L4	L8	L16	**L8**	83	L2	L4	L8	L16	**L4**	66	=	=	=	=	L4
84	G	G	=	L16	**L16**	41	L2	L4	L8	L16	L4	88	L2	L4	L8	L16	L4	68	=	=	=	=	L2
85	G	G	G	G	**L16**	42	L2	L4	L8	L16	L8	90	L2	L4	L8	L16	**L2**	74	=	=	=	=	L16
86	G	G	G	G	**L16**	43	L2	L4	L8	L16	**L8**	95	L2	L4	L8	L16	**L2**	75	=	L4	=	=	**L2**
87	G	G	=	=	**L16**	46	L2	L4	=	G	**L2**	5	=	=	=	L16	**L16**	79	=	=	L8	L16	**L8**
89	G	G	G	L16	**L16**	47	L2	L4	L8	L16	**L8**	7	=	=	=	=	L16	82	=	=	=	=	**L2**
91	G	=	=	L16	**L16**	48	L2	L4	L8	L16	L8	8	=	G	G	G	**L2**	93	=	=	=	=	L2

VII. OTHER LOCAL TOPOLOGIES

Any particle communication topology that is not global—one in which particles do not have immediate access to every pbests—is, by definition, local. However, some topologies are more local than others.

The number of edges connecting each vertex, or degree, k of a regular graph is perhaps the simplest measure of (inverse) locality. It is simply the number of neighbors in a regular particle topology. Another measure is the minimum flow time (number of iterations) τ for information to become global.

Consider a sequence of topologies constructed from a ring topology by adding links to next-nearest neighbors and to next-next-nearest neighbors and so on. The result is a degree k topology, denoted Lk, where the neighborhood of particle i is $\{i \ominus (k/2), \ldots, i \ominus 1, i \oplus 1, \ldots, i \oplus (k/2)\}$, $\ominus, \oplus$ are arithmetic operators modulo the swarm size, N, and k is an even integer. Lbest $\equiv$ L2 lies at one extreme and Gbest $\equiv$ L$(2\lfloor (N/2) \rfloor)$ lies at the other ($i \ominus (N/2)$ is identified with $i \oplus (N/2)$ when N is even). The flow time in an Lk topology is related to the degree by $\tau = \lceil (N/k) \rceil$. It is an even coarser measure of locality. For example, topologies with k between $2\lceil (N/4) \rceil$ and $N-1$ share a flow time of two iterations.

The Lk topologies can be employed to explore how PSO behavior varies along a spectrum of localities from Lbest to Gbest.

A. Experiments

Local neighborhoods L4, L8, and L16 were tested using the methodology of Section III-B in order to assess the impact of locality on swarm performance over the combined benchmark. The data from the Lbest and Gbest trials reported in Section III was added to provide a span of degrees ($k = 2(\text{Lbest}), 4, 8, 16, 30(\text{Gbest})$) and flow times ($\tau = 15, 8, 4, 2, 1$).

B. Results

Table IV shows the Lk versus Gbest Wilcoxon tests for each function, organized by Gbest preferred, Lbest preferred and tied functions where preference is according to the error results reported in Section III. The table shows the winning topology in each comparison and the Lk topology with the lowest mean error (LME).

1) Gbest Preferred Functions: Gbest loses to L16 (L16 < G) in seven functions, L8 < G in two functions and Gbest ties with L8 or L16 in ten functions. The Kruskal–Wallis (KW) test failed to distinguish the Lk algorithms in three of the 23 functions. L16 returned the LME in 17 cases—however the KW test only indicates that the there is a significant difference in error in at least one, and not necessarily in all four, algorithms.

In summary, Gbest beats any local topology in only ten of the 23 functions in this group.

2) Lbest Preferred Functions: The trials in this group tell a contrasting story. Gbest optimizes better than an Lk topology in just two cases (f36 and f46). L4 < Gbest in 36 out of 37 functions in this group and even L16, the most distant topology from Lbest, is equalled by Gbest just four times and beaten once. The KW test failed to distinguish any Lk in 13 cases; of the remaining 24, L2 recorded the LME 11 times.

In summary, Gbest beat a local topology in two out of 148 comparisons. Lbest is confirmed as the algorithm of choice in this function group.

3) Tied Functions: The functions remain tied for ten of the 24 functions in this group. Otherwise, an Lk topology optimizes better, or is equivalent to, Gbest in all but two (f8 and f40) functions. The KW test fails to distinguish an L*k* algorithm in 50% of the tied functions.

4) All Functions: The number of Wilcoxon winners per degree over the entire benchmark of 84 functions is tabulated as follows.

	L2	**L4**	**L8**	**L16**
G < Lk	23	24	13	12
Lk < G	37	42	43	47
Lk = G	24	18	28	25
LME	21	18	18	27

Although exhaustive tests are required to differentiate the L*k* algorithms, the local versus Gbest tests show that choosing a local topology can reduce the number of functions where Gbest was superior to a mere 12. Of particular interest, the replacement of Gbest with a topology that is only marginally local can improve PSO performance in many cases where Gbest was previously found to be superior to Lbest.

C. Conclusion

It is likely that there is an optimal amount and pattern of connectivity for each function, and perhaps an optimal amount and pattern for general use. Indeed, the trials on several L*k* networks indicate that a judicious choice of local topology can improve Lbest performance in many cases. We find that Gbest beats a local topology in just 14% of functions.

Although the investigation of the aspects of communication topology is an excellent topic for future research, this paper focuses on Gbest's standing as the default topology for PSO—a position that is demonstrably in error.

VIII. Conclusion

A. Message From the Results

The conclusion is easily stated: whenever canonical PSO is used, Lbest and other L*k* local topologies perform better on more difficult problems because particles spread out and keep looking after Gbest has collapsed. Gbest can perform well on some functions, but in the face of multimodality and other difficulty factors, local PSO pulls out ahead.

Lbest performs better or equally to Gbest on 85% of the wide-ranging benchmark of 84 test functions in a budget of 150 000 evaluations. A study of significant improvements in the last quarter of the run suggests that this figure could be extended to 89%. Apart from improved performance, the series of trials and analyses presented here show that Lbest improves later in the run, that the improvements are significant, that it has higher terminal diversity and that it has a higher mobility between optimum basins. Lbest performs equally or better then Gbest on a selection of real world problems.

We find that Gbest beats any local L*k* topology for 14% of the test functions. Even adding a small amount of locality (from global to L16) can significantly improve performance.

In order to cast doubt on the current practice of employing Gbest canonical PSO as the default version, it is sufficient to find a single alternative topology that performs better. This has been accomplished.

The categorization of problem by binary property gives an indication of which type of canonical PSO to utilize on a new problem in the case that some knowledge of that problem type is available. Gbest can certainly be employed if the problem is unimodal or separable. Noisy, high-conditioned, deceptive, and problems with a sensitive/dominant variable could benefit from Gbest (although examples of these categories were not numerous in the combined benchmark). A local topology should be applied to multimodal, nonseparable problems and problems with neutrality (plateaus), asymmetry and with a combination of difficulties (as manifest by the composition functions). There is evidence that local canonical PSO is preferable in situations where the optimum is on or outside the search bounds (but once more, these types were not prevalent in the test set).

Since nonseparable, multimodal and combination functions represent the harder problem type, and unimodality and separability the easier type, local canonical PSO is generally to be picked if the problem is suspected to be complex. In particular, real world problems are invariably complex and an application of canonical PSO would benefit, as witnessed from tests on six wide-ranging real world problems, from a local topology.

These conclusions are drawn from a comprehensive series of trials on the canonical PSO, a version that includes particle dynamics and communication in their purest forms and is historically and, arguably, conceptually, the basis of subsequent PSO development. It is probable that the conclusions transfer to many PSO variants: PSO research that ignores the significant role of interparticle communication risks underselling any claimed advantage; comparative evaluation between a new variant and only Gbest canonical PSO has dubious validity.

B. Measuring Significant Improvement and Mobility

A general method for coarse-graining the information contained in a batch of runs has been advanced: the LSI. LSI depends on a level, s, defined as a proportion average total logarithmic function change in the course of a run. The LSI is placed within a small number of equal evaluation intervals I_i that cover the total evaluation budget. The number of LSI's in a given interval can then be used for algorithm comparison. The probability that an algorithm is not stagnant in the final interval can be used to define an *improving* algorithm. We argued that an improving algorithm in I_4 can be expected to continue to find better function values if the run budget were extended. This enables us to distinguish algorithms that become stuck, from those that are merely optimizing slowly.

The downhill basin of attraction of an optimum has been defined and computed for a bimodal problem. This function (2-CONES) is the superposition of two cone base functions. Different instances of 2-CONES are obtained by tweaking cone parameters. A simple difficulty measure, defined as the ratio of basin volumes, is found to match canonical PSO performance. The ability of an algorithm to jump between basins—the mobility—is then observable.

C. General Conclusion

The Gbest population converges rapidly, which may be beneficial when a problem is simple but, as the present results show, it tends to perform relatively poorly when there are features in the function landscape blocking its way or attracting it toward local optima. Local topologies offer a much richer dynamic, with individuals interacting locally to form subpopulations clustered in the search space, exploring multiple regions before converging, one by one if at all, on a superior solution. Each particle must be persuaded, in a sense, that a solution is better than the one it has been developing, and good problem solutions flow through the population with greater or less speed, depending on the connectivity; one particle persuades its neighbor, and if the solution is good then that neighbor persuades its neighbor on the other side, and so on. This amounts to a distributed marketplace of competing problem solutions. A particle in one part of the ring has no information about what is happening in other parts, and eventually better problem solutions spread through the population.

Let us emphasize that Lbest and Gbest are only two possible topologies for a particle swarm, and the space between them is rich with potential, as demonstrated in Section VII. Besides holding a key position in the particle-swarm literature, they establish a pair of end-points that enable contrast between highly centralized and widely distributed swarm communication patterns. In the centralized Gbest topology all influence flows from the one individual particle that has found the best solution so far. The sense of this is that the entire population acquires the best information available at any time; the problem is that the first decent guess at a solution is unlikely to be the best one, and by betting the whole population's efforts on early candidates, pretty soon the entire population is trapped in an inferior part of the search space.

The Lbest and L*k* topologies can have many parallel searches going on, as pairs and groups of particles entrain and influence one another in one region or another. In practice, better solutions begin to attract adherents, and small subpopulations eventually, but not always, tend to dissolve and merge as superior solutions become the neighborhood best for one after the other. There is not much ambiguity in the present results. There are some functions where Gbest performs better than any local topology. If a researcher is working with a problem that is known to be unimodal or separable, or that has previously been satisfactorily solved with a Gbest approach, or when a speedy and merely adequate solution is required, then it may be sensible to choose the faster and simpler Gbest. And it may perform well on certain more difficult problems, as well, with lesser probability. If a researcher knew in advance what the fitness landscape looked like, then the choice of a solution strategy would be simple and a precise, nonstochastic approach would likely be appropriate.

It is hoped that the present results will lead researchers to focus on the dimensions of particle swarm communication topologies and their effects on performance. Particles are unable to solve any problem alone, but through communication they are very powerful. And now, after more than 20 years of research, the effects of the communication topology are still not very well understood. Learned patterns of thought guide our attention to the particles, to the "items" comprising the system, but only a holistic analysis of the population can reveal how the algorithm works and how to improve it. Besides dynamic and adaptive approaches, there is a need to understand the effects of speeding and damping the flow of information through the population during the search process. These effects will interact with other aspects of the algorithm; for instance, though it is customary to have a particle select its best neighbor, this is not necessary; good results can be attained by choosing a neighbor at random Pea *et al.* [44], or averaging across all neighbors Mendes and Kennedy [9], and the effect of the communication topology will depend on the interaction rule. Most of all, it is hoped that the particle swarm research community will show itself to be robust and will re-evaluate the choice of topology used in everyday research. It is painful to read papers that mention the well-known weakness of the particle swarm for being trapped in local optima, and then to realize that the author is unaware that the weakness is an effect of the topology that has been implemented.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, pp. 1942–1948.
[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, 1995, pp. 39–43.
[3] A. P. Engelbrecht, "Particle swarm optimization: Global best or local best?" in *Proc. BRICS Congr. Comput. Intell. 11th Brazil. Cong. Comput. Intell.*, Sep. 2013, pp. 124–135.
[4] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. Congr. Evol. Comput.*, 1998, pp. 69–73.
[5] J. Kennedy, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," in *Proc. Congr. Evol. Comput.*, vol. 3, 1999, pp. 1931–1938.
[6] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
[7] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, 2007, pp. 120–127.
[8] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.
[9] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 204–210, Jun. 2004.
[10] H.-G. Beyer, *The Theory of Evolution Strategies*. Heidelberg, Germany: Springer, 2001.
[11] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
[12] P. N. Suganthan, "Particle swarm optimiser with neighbourhood operator," in *Proc. Congr. Evol. Comput. (CEC)*, 1999, pp. 1958–1962.
[13] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. Congr. Evol. Comput.*, 2002, pp. 1671–1676.
[14] T. Peram, K. Veeramachaneni, and C. K. Mohan, "Fitness-distance-ratio based particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, 2003, pp. 174–181.
[15] J. J. Liang and P. N. Suganthan, "Dynamic multi-swarm particle swarm optimizer," in *Proc. IEEE Swarm Intell. Symp.*, 2005, pp. 124–129.
[16] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1272–1282, Dec. 2005.
[17] M. Clerc, "Standard PSO 2006," Independent Consultant, Rep., 2006. [Online]. Available: http://www.particleswarm.info/Programs.html
[18] Y. Liu and L. Ma, "Stochastic diffusion search algorithm for quadratic knapsack problem," *Control Theory Appl.*, vol. 28, no. 8, pp. 1140–1144, 2011.
[19] X. Li, "Niching without niching parameters: Particle swarm optimization using a ring topology," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 150–169, Feb. 2010. [Online]. Available: https://doi.org/10.1109%2Ftevc.2009.2026270
[20] D. Parrott and X. Li, "A particle swarm model for tracking multiple peaks in a dynamic environment using speciation," in *Proc. Congr. Evol. Comput.*, 2004, pp. 98–103.

[21] J. Kennedy, “Stereotyping: Improving particle swarm performance with cluster analysis,” in *Proc. Congr. Evol. Comput.*, 2000, pp. 1507–1512.
[22] K. E. Parsopoulos and M. N. Vrahatis, “Modification of the particle swarm optimizer for locating all the global minima,” in *Artificial Neural Nets and Genetic Algorithms*. Vienna, Austria: Springer, 2001, pp. 324–327.
[23] K. Parsopoulos and M. Vrahatis, “UPSO: A unified particle swarm optimization scheme,” in *Lecture Series on Computer and Computational Sciences*, vol. 1. Utrecht, The Netherlands: VSP, 2004, pp. 868–873.
[24] R. Brits, A. Engelbrecht, and F. van den Bergh, “A niching particle swarm optimizer,” in *Proc. 4th Asia-Pac. Conf. Simulat. Evol. Learn.*, 2002, pp. 692–696.
[25] M. Clerc, “Standard particle swarm optimisation,” Independent Consultant, Rep. hal-00764996, 2012.
[26] M. Clerc, *Particle Swarm Optimization*. London, U.K.: ISTE, 2006.
[27] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, “Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2013, pp. 2337–2344.
[28] M. R. Bonyadi, X. Li, and Z. Michalewicz, “A hybrid particle swarm with a time-adaptive topology for constrained optimization,” *Swarm Evol. Comput.*, vol. 18, pp. 22–37, Oct. 2014.
[29] Y. Marinakis and M. Marinaki, “A hybridized particle swarm optimization with expanding neighborhood topology for the feature selection problem,” in *Hybrid Metaheuristics*. Heidelberg, Germany: Springer, 2013, pp. 37–51.
[30] W. H. Lim and N. A. M. Isa, “Particle swarm optimization with increasing topology connectivity,” *Eng. Appl. Artif. Intell.*, vol. 27, pp. 80–102, Jan. 2014. [Online]. Available: https://doi.org/10.1016%2Fj.engappai.2013.09.011
[31] M. Pant, T. Radha, and V. P. Singh, “A new diversity based particle swarm optimization using Gaussian mutation,” *Int. J. Math. Model. Simulat. Appl.*, vol. 1, no. 1, pp. 47–60, 2008.
[32] F. Van den Bergh and A. P. Engelbrecht, “A convergence proof for the particle swarm optimiser,” *Fundamenta Informaticae*, vol. 105, no. 4, pp. 341–374, 2010.
[33] C. Han and L. Wang, “Array pattern synthesis using particle swarm optimization with dynamic inertia weight,” *Int. J. Antennas Propag.*, vol. 2016, pp. 1–7, Mar. 2016.
[34] P. N. Suganthan *et al.*, “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” School Elect. Electron. Eng., Nanyang Tech. Univ., Singapore, Rep. NCl-TR-2005001, 2005. Accessed: Nov. 2009. [Online]. Available: http://www3.ntu.edu.sg/home/EPNSugan
[35] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, and Nanyang Technol. Univ., Singapore, Rep. 201212, 2013.
[36] T. M. Blackwell, “Particle swarms and population diversity,” *Soft Comput.*, vol. 9, no. 11, pp. 793–802, Nov. 2005.
[37] T. Richer and T. M. Blackwell, “When is a swarm necessary?” in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 1469–1476.
[38] R. W. Morrison and K. A. De Jong, “A test problem generator for non-stationary environments,” in *Proc. IEEE Congr. Evol. Comput.*, 1999, pp. 2047–2053.
[39] S. Das and P. N. Suganthan, “Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems,” Nanyang Technol. Univ., Singapore, and Jadavpur Univ., Kolkata, India, Rep., Dec. 2010.
[40] E. Sandgren, “Nonlinear integer and discrete programming in mechanical design optimization,” *J. Mech. Design*, vol. 112, no. 2, pp. 223–229, 1990.
[41] D. J. Wales and J. P. K. Doye, “Global optimization by basin-hopping and the lowest energy structures of Lennard–Jones clusters containing up to 110 atoms,” *J. Phys. Chem. A*, vol. 101, no. 28, pp. 5111–5116, 1997.
[42] P. P. Bruyant, “Analytic and iterative reconstruction algorithms in SPECT,” *J. Nucl. Med.*, vol. 43, no. 10, pp. 1343–1358, 2002.
[43] L. A. Shepp and B. F. Logan, “The Fourier reconstruction of a head section,” *IEEE Trans. Nucl. Sci.*, vol. NS-21, no. 3, pp. 21–43, Jun. 1974.
[44] J. Pena, A. Upegui, and E. Sanchez, “Particle swarm optimization with discrete recombination: An online optimizer for evolvable hardware,” in *Proc. 1st NASA/ESA Conf. Adapt. Hardw. Syst. Soc. (AHS)*, 2006, pp. 163–170.

Tim Blackwell received the B.Sc. degree in physics from the University of Bristol, Bristol, U.K., the M.Sc. degree in computer science from University College London, London, U.K., and the D.Phil. degree in theoretical physics from the University of Sussex, Brighton, U.K.

He is a Senior Lecturer of computing with the Goldsmiths, University of London, London. He is well known for the application of swarms to improvised music. He was a Principal Investigator for the EPSRC funded Live Algorithms for Music research network and Extended Particle Swarms project. His current research interests include quantum field theory, computer music, digital art, computational swarm intelligence, and music complexity.

James Kennedy received the Doctorate degree in social psychology from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 1992.

He is a co-originator of the particle swarm algorithm and has been publishing research on the topic since 1995. He was with the U.S. Federal Government for 23 years, retiring as the Director of the Office of Analysis and Research Services, U.S. International Trade Commission, Washington, DC, USA, in 2017.

Dr. Kennedy was a recipient of the IEEE Computational Intelligence Society’s Evolutionary Computation Pioneer Award for his research on particle swarm optimization.