Review

All Currently Known Publications On Approaches Which Solve the Moving Peaks Problem

18/07/07

compiled by Irene Moser

Swinburne University of Technology

Melbourne, Australia

1 Evolutionary Algorithms

Evolutionary algorithms (EA) have been by far the most common approach to the Moving Peaks problem – 9 of 19 reviewed papers are EA-based -, applied first by by the creator of the benchmark in [7].

This seminal paper solves the new benchmark problem using a memory-based approach, which divides the real-value encoded population into two parts. One part is randomised after a change and stores individuals in the memory while the other part is responsible for the search, for which it retrieves individuals from the repository. In a proposed alternative, the population is split into three islands with all populations contributing to the memory while only one of the populations retrieves individuals from this memory. The individuals in the memory are counted as part of the population, which has a size of 100. Only one best individual is added to memory from time to time (in this case, every 10 generations), and several replacement strategies are explored. The two best-performing strategies are replacing the most similar individual and replacing the most similar individual if it is of worse fitness. Individuals are retrieved from memory after each change, which is detected through recalculation of the fitness of the solutions in memory. The experiments on the Moving Peaks problem, using five peaks, compare the performance of the implementations with one, two and three populations combined with memory as well as a single-population GA without memory. The results show a superior performance of the implementation with two populations when the peaks change their heights but do not move, whereas the three-population variation performs best among the candidates when the peaks change their locations as well as their heights.

The author concedes that memory-based approaches have limited application since they are only truly useful in the case of an exact recurrence of a previous situation but also concludes that increased diversity contributes more to the performance of a memory-based GA than to a standard implementation.

The numerical results given use the offline performance as a measure, which is not universally comparable when the maximum height of the landscape is unknown.

In further work, Branke et al. [9] propose the self-organising scouts approach, which does not explicitly detect a change and react to it. The method is based on a forking mechanism which starts with a single population, subsequently dividing off subpopulations with a designated search area and size. The area size is allocated according to a predefined distance from the best individual which defines the initial group at the time of the separation, then remains constant for the subpopulation life cycle. The subpopulation size varies according to a quality factor derived from the group's best individual's fitness and the level of group dynamics, a measure of the groups improvement potential looking at its fitness increase during the last generation. While the parent population is kept out of the child population areas, the child populations may search each other's areas as long as the group's centre (fittest individual) does not enter another child population's space. Whenever this happens, the subpopulation is eradicated.

The algorithm's results on solving the Moving Peaks problem are compared to those produced by a standard GA. The change severity of the peak movement is set to 1, whereas the largest value used for the memory-based approach in [7] is 0.6. The landscape changes every 5000 evaluations. A crucial definable parameter of the self-organising scouts approach is the range of admissible subpopulation sizes. All respective settings chosen for the experiments outperform the standard GA, however the best result is achieved with a larger subpopulation size. Varying the overall population size is reported to have little effect on the outcome. The authors also conclude that increasing the change frequency of the landscape enhances the advantage of the proposed approach over the standard GA.

Again, the results are given as the offline performance with no indication of the maximum height, which does not facilitate a comparison across the reported approaches.

Kramer and Gallagher [17] are using the Moving Peaks problem to compare their different variations of a GA approach to be used as a hardware implementation of a controller for a device. Within the controller, the GA's task is to evolve a neural network with a suitable strategy to optimise a controlled device's performance based on sensor input. The GA receives scalar feedback from a consumer on the controlled device's performance. Due to the specific needs of the controller project, the Moving Peaks parameter settings are somewhat original and do not compare with commonly used instances.

The authors [17] present research based on a GA adaptation to dynamic problems called mrGA, which uses elitism, mutation and resampling. The approach evolves a real-value encoded probability vector which gets feedback from the binary encoded solutions. The re-sampling strategy re-evaluates the fitness of the current best solution to detect a change. This base implemenation is then enhanced with hypermutation and random immigrants to form two different variations, which are tested against the performance of the base version.

As benchmark instances, a Moving Peaks scenario with 25 narrow peaks is used as well as a single peak moving among a grid of smaller, equally sized and shaped peaks. The algorithm variations using enhanced diversification outperform the basic mrCGA as expected on both metrics used: the online error and the pre-shift error.

While the online error is one of the optional metrics provided with the Moving Peaks benchmark, the pre-shift error is a measure introduced by Kramer and Gallagher. It records the deviation from the current maximum immediately before the landscape is changed again.

Branke and Schmeck [10] introduce the offline error as a performance measure of the Moving Peaks problem, allowing for cross-publication comparison of results. Furthermore, new empirical results are obtained from the self-organising scouts approach first proposed in [9], this time on a Moving Peaks problem with 10 peaks, as well as on the memory-based “two subpopulations” approach introduced in [7]. For the comparison, a standard GA and a memory-based GA have also been implemented. The experiments record the offline error for different severities of change and conclude that although the performance of the algorithms declines with increasing severity in all cases, the self-organising scouts as well as the memory/search populations approach from [7] perform far better than the standard versions, maintaining a far greater diversity over the landscape. Over a larger number of peaks, both the self-organising scouts and the memory/search population with increased memory size are reportedly able to maintain a suitable diversity even with as many as 50 peaks. However, this diversity is only beneficial to the performance in the case of the self-organising scouts.

The memory/search populations approach proposed by Branke [7] forms the basis of the algorithm developed by Zou et al. [27]. Instead of describing the partitioning as two populations and a memory, as done in the former source, in [27] the authors speak of three populations – memory, mutation and re-initialisation. The memory is initialised as the initial population of all parts are created, and the “re-initialisation” population performs the tasks of Branke’s “search” population. This population only contributes to the memory (as was the case in Branke’s original), whereas the “mutation” population exploits the memory as well as contributes to it.

The re-initialisation population is recreated after a change has been detected (as is the case with the “search” population in [7]). The only apparent difference between the two algorithms is the use of a chaotic sequence for all random number generation in [27].

The authors use two problems for experiments; a parabolic function also solved with an algorithm not discussed here, and the Moving Peaks problem. One of the two problem instances used seems identical with the one used by Branke [7]. The offline performance given in both papers could be expected to be comparable. This would mean that the new algorithm described in [27] competes favourably with the earlier version. The authors, however, maintain that their performance evaluation was different from the method used in the original paper by Branke. The authors also use the offline error as a performance measure, which simplifies comparisons with other approaches.

A speciation approach applying the formation of colonies is described and explored by Ronnewinkel and Martinetz [26]. Named MPGA, it bears some resemblance to the self-organising scouts approach described in [9] and revisited in [10]. In this new approach, colonies are formed around centroids defined as the individuals with the best fitness in the region. The individuals are listed by distances. Slumps in the fitness scores of the individuals indicate borders between colonies. As with the parent population of the self-organising scouts, the “search population” is allowed to roam the space between child populations (called colonies), but not to enter the space of a colony. New centroids are detected and colonies are divided off when two colonies split – it is the emerging fitness slumps between the members which give rise to a split – or when the “search population” finds a new individual whose fitness is competitive with existing centroids. Colonies are merged when a valley between the fitnesses of their members disappears.

To further enhance diversity, the authors keep a list of centroids which is longer than the actual colony count, storing potential candidates at the end. When all centroids’ fitnesses are re-evaluated, centroids may move up in the ranking list and start a new colony. To the same end, the general search population uses a diversity-preserving selection. One obvious advantage of their approach over the self-organising scouts, mentioned also by the authors, is that no minimum or maximum diameter values have to be set beforehand.

For their experiments, the authors use settings almost identical to scenario 1 of the Moving Peaks benchmark and the offline error as a metric. The MPGA is compared with many GAs that maintain diversity by tag sharing, deterministic crowding, multi-niche crowding and restricted tournament selection. The authors find that MPGA outperforms all its competitors, although it is clearly superior only in the case of 5 peaks, less so when 50 peaks are present.

Multi-objective GAs have been proposed for solving single-objective functions. Bui, Branke and Abbass [11] apply this principle to dynamic optimisation and test it on the Moving Peaks problem. Introducing an additional objective serves the purpose of maintaining diversity. To this end, the authors introduce six different additional “artificial” objectives: time-based, random, inversion of the primary objective function, distance to the closest neighbour, average distance to all individuals and the distance to the best individual in the population. These are compared to a standard GA and a random immigrants variation – both with elitism - on a problem instance listed as Scenario 2 on the Moving Peaks web site [8], with a choice of 50 peaks. As a metric, the average generation error is used; each generation's best individual's height is subtracted from the maximum height in the landscape. The multi-objective variations using the distance to all other individuals and the nearest individual respectively are reported to perform significantly better than all other GAs.

The same authors extend this work in [12], where they use the NSGA2 algorithm developed elsewhere and based on ranking the non-dominated solutions and maintaining diversity using the crowding distance as a basis for the implementation of the six mentioned "artificial objectives". As in [11], a standard GA and a random immigrants version are used with varying crossover and mutation rates. Additionally, a benchmark algorithm called MutateHigh is used, defined as a GA with a 50% mutation rate. Again, the average generation error is applied as a measure, but this time, the more generally used offline error is also given. The experiments vary in height and width change. The same variations of the multi-objective optimiser as in [11] show superior performance on all four problem instances with different change severities. Each algorithm achieves its best results with different crossover and/or mutation rates.

One of the newer concepts introduced is exaptation, defined as adaptation to a situation that may or may not occur in the future. Fentress [14] *preapts* solutions with a potential to become global maxima using a Genetic Algorithm (GA) with hill-climbing scouts, which is very successful due to the smooth ascent, a consequence of a deliberate simplification of the Moving Peaks problem. So as not to exploit this trait, Fentress further develops his approach into a multi-population GA called mp-eGA with tribes centered around a randomly chosen point. A Gaussian distribution ensures that the centre points of tribes are not located too closely. As the tribes evolve, the search space of the tribes is curbed using a Gaussian distribution to discourage proximity to the centre of another tribe. The population size is adapted dynamically, depending on the quality of the tribe which is a function of the dynamism of the tribe (difference in fitnesses) and the aggregated fitness. This fitness measure has first been introduced in [9].

The performance of this algorithm is compared to the author's implementation of a standard GA and the variations proposed in [7] and [9]. The results are measured as the normalised offline performance and suggest that the mp-eGA outperforms the used benchmarks by 5%-10%. The question arises whether the employed method to enforce diversification in a multi-population GA deserves the term exaptation. This becomes particularly evident in the case of the hill-climbing GA: The populations simply maintain a position on the highest point of the existing peaks;

evidence of the algorithm finding maxima before they emerge could not be found in the description.

Ayvaz et al. [1] present a review of the EA-based approaches and compare their performance on an undefined scenario, exploring the algorithms under different severity of change. The self-organising scouts introduced by Branke [6] produce the best results among all previously known EA-implementations and are reported to perform even better when enhanced with a crossover-based local search introduced in Lozano et al. [21].

2 Particle Swarm Optimisation

This stochastic algorithm was inspired by the flocking behaviour of swarms. Each particle has a slowly diminishing velocity and is attracted both to its own best find of solution and the swarm's best-known location. Similar to simulated annealing, PSO's particles slow down to provide a more fine-grained exploration toward the end of the search. This convergence behaviour is less desirable in dynamic environments. Blackwell [2] introduced charged particles that repel each other and circle around neutral particles of the swarm as a counterweight to this property. Blackwell and Branke [3] apply a multi-population version of the same approach as multi-CPSO to the MP problem. They also introduce multi-Quantum Swarm Optimisation (multi-QSO), a variation whose charged particles move randomly within a cloud of fixed radius centred around the swarm attractor.

Both alternatives use an exclusion radius to stop swarms from exploring the same areas, reinitialising the worse-performing of two swarms within the exclusion zone.

Given a constant number of 100 particles, the authors show experimentally that the optimal number of swarms for Scenario 2 is around 10 (±1), and that the usage of a single swarm leads to convergence to a local optimum for all algorithms (PSO, multi-CPSO, multi-QPO). Multi-QPO shows superior performance in almost all configurations and produces the best result in the test series.

Unlike the previously discussed PSO approaches, Janson and Middendorf [16] propose to respond explicitly to a change in the landscape after detection. Their hierarchical variation of the PSO (H-PSO), first explored in a static context [15], is compared experimentally to variations of the same base algorithm.

Instead of following a global attractor, each particle uses the best location found by the individual immediately above it in the tree structure in addition to its own best find. At each evaluation, the tree structure is adjusted root to leaf by swapping nodes if the child node has performed better. Using more attractors, this structure is likely to maintain diversity in a dynamic environment.

This property is further enhanced by explicit change detection. After a change, a number of particles is re-initialised to a random position. A variation named Partitioned H-PSO (PH-PSO) is also introduced. It divides the tree structure into subtrees whose nodes are reinitialised if they are at a predefined level in the tree. This leaves the root node and a given level of children to preserve the memory of a previous location, while the lower branches have their new root nodes reinitialised. After a given period of time, the branches are rejoined to the tree according to the structure evaluation procedure.

A comparison between PSO, H-PSO and PH-PSO is given using three different functions, the Moving Peaks problem among them. It is not clear which scenario is used, though change severities of 0.1 and 3.0 are given. The offline errors are shown as a graph, not listing numerical values. It seems that H-PSO outperforms the other variations on dynamics with smaller severity, while PH-PSO is most successful on more severe changes if all subtrees are reinitialised after a change.

Blackwell and Branke [4] add anti-convergence to the exclusion and quantum/charged particle features they first conceived by Blackwell and Branke [3] and Blackwell [2] respectively. Anti-convergence maintains diversity by reinitialising the worst-performing swarm as soon as all swarms have converged. As exclusion radii around the existing swarms are enforced, the new swarm is guaranteed to search for a new peak. As in [2] and [3], exclusion is enforced by reinitialising the worse-performing of the conflicting swarms.

Some analysis is given as to the ideal parameter settings for swarm size and quantum cloud radius, which are derived from the peak shift distance. Similarly, the number of swarms is set equal to the number of peaks. This assumes that both the distance and the peak count are known and do not vary greatly.

Implementations based on quantum and charged particles are compared on Scenario 2 of the MP problem. The best-performing configuration is a quantum multi-swarm approach with 5 quantum and 5 neutral particles and as many swarms as there are peaks. The experiments show that anti-convergence is beneficial when there are more swarms than peaks.

The authors point out that the performance deteriorates when there are more swarms than peaks due to the constant reinitialisation of swarms that cannot find a peak – as all peaks are occupied – and are constantly being reinitialised. Consequently, this approach shows little promise for an environment where the number of peaks is unknown, which is likely to be the case in a realistic optimisation problem.

Parrott and Li [25] adapt the speciation technique introduced for GA by Li et al. [19] to PSO. Given an area with a fixed radius around the attractor (the best particle in the species), the members of the species are defined by the area they are located in. In case of overlap, the member belongs to the species with superior attractor. When a maximum number of species members is exceeded, the worst members are reinitialised to random positions.

As a measure, the deviation of the best particle from the global optimum is used. The authors investigate different population sizes and speciation radii. They conclude that large radii and larger population sizes lead to a more even spread over the landscape and therefore provide swifter reactions to change. As a benchmark, the problem defined by De Jong and Morrison [13], is used. The approach is only reviewed here because it forms the basis of another approach to the Moving Peaks.

Li, Branke and Blackwell [20] combine some of the aspects of previous work of Blackwell and Branke [4] with the notion of speciation PSO (SPSO) introduced in an earlier publication of Li [18]. The approach tackles problem dynamics by detection and response. It is designed to optimise problems with primarily unknown numbers of peaks.

After each change, the species are reinitialised by sorting the particles according to current quality and adding them to a current species if the landscape already has particles within the predefined species radius or by making them new species centres if they are not, observing the maximum member count of species devised first by Parrot and Li [25]. Particles that exceed this threshold are initialised to a random location. Various anti-convergence techniques explored by Blackwell and Branke in [3] and [4] are added to a variety of different implementations whose performance is compared. As a new step towards diversity maintenance, a diversity measure is introduced for particles within the same species. Species which have converged past this threshold are reinitialised around the best-performing particle.

The approach is tested on Scenario 2 of the MP problem. The best-performing variation is a PSO with speciation and an initial population of neutral particles, in which the converged swarms are placed randomly within a radius of their previous attractor. A subswarm is considered as having converged when the distance between the central (attractor) particle and the farthest particle is less than 0.1. One half of the particles are reinitialised as neutral, the other half as quantum particles as described by Blackwell and Branke [4]. The neutral particles are designed to perform the hill climbing task.

3 Others

Only three of the available approaches use neither EA nor PSO as a basis for their implementations. Mendes and Mohais [22] experiment with a multi-population approach of differential evolution (DE) and explore seven different schemes in combination with three ways of developing elite individuals, one of them based on the idea of quantum particles introduced by Blackwell and Branke [3].

The experiments use Scenario 2 whose peak count of 10 is assumed to be a known constant and subsequently used as the fixed number of populations and in determining the diameter of the exclusion zones. In case of an overlap, the population with an inferior best solution is re-initialised.

The authors report receiving their best results using a scheme which involves the best individual and the difference between four random population-best individuals. This best-performing implementation uses 40% elite individuals created from the population-best individual using a Gaussian distribution. The experiments show that these settings equal the performance observed by Blackwell and Branke [3]. Mendes and Mohais [22] also observe that smaller populations achieve better results and attribute this to the frequency of change in Scenario 2, which is once in 5000 iterations.

Stochastic Diffusion Search (SDS) is a metaheuristic inspired by neural networks. Meyer, Nasut and Bishop [23] offer a detailed explanation of the approach, which involves subdividing the objective function and assigning different parts of it to agents who evaluate it for their adopted solution. Initially, solutions are assigned randomly. The success of the partial solution evaluation by the agent decides whether the agent is active. Inactive agents randomly choose another agent and adopt its solution if the chosen agent is active. If not, a random solution is created.

Although enhanced with some factors aimed at maintaining diversity, this approach does not compare favourably to local search and PSO on a pattern matching problem. The authors provide neither a detailed description of the adaptation of SDS to the Moving Peaks problem nor numerical results; the performance is simply described as “following the peak almost perfectly”.

Moser and Hendtlass [24] apply a multi-individual version of Extremal Optimisation (EO) to the Moving Peaks problem. EO was first introduced by Boettcher and Percus [5] and lends itself well to dynamic problems where the time of change is not known. The Multi-Individual Multi-Phase EO algorithm devised especially for the Moving Peaks problem, however, is better suited to the tracking task of following the peaks and scores high on the offline error metric. It consists of separate global and local search phases with deterministic sampling strategies and devices to save function evaluations.

4 Comparison

The results of the approaches described above are listed here if they were stated in a comparable form. The scenarios with the given parameters do not represent instances of the problem, as the resulting instances depend on the random number generator used. This limitation, however, is reduced in proportion to the number of trials used. Therefore, the number of experiments is listed with the results.

Authors	Base algorithm	No of trials for average	No of peaks	No of evaluations**	Result	Unit
Bui et al. [12]	EA	30	50	2500***	9.52 ±0.45	aGE*
Blackwell & Branke [3]	PSO	50	10	5000	2.16 ±0.06	offline error
Li et al. [20]	PSO	50	10	5000	1.93 ±0.06	offline error
Mendes & Mohais [22]	DE	50	10	5000	1.75 ±0.032	offline error
Blackwell & Branke [4]	PSO	50	10	5000	1.72 ±0.06	offline error
Moser & Hendtlass [24]	EO	100	10	5000	0.66 ±0.2	offline error

All approaches used Scenario 2 as a benchmark.
*Average generation error; average of best individual of each generation.
**Time measured as evaluations at the solver's disposal, before next change. For the offline error, the longer the algorithm runs after a good solution has been found, the more the result is skewed toward a good solution.
***Authors mention 25 generations with a population size of 100.
All authors chose a change severity of 1.0 and a λ of 0.0.

Many of the reviewed papers' results are not included in this comparison. Three possible reasons would account for an exclusion:

- No specific scenario was used / the information on the settings used was incomplete;
- No numerical results were given (no attempt was made to interpret graphical representation);
- The numerical results used an incomparable unit.

References

[1] Ayvaz, D., Topcuoğlu, H. and Gurgen, F., “A Comparative Study of Evolutionary Optimisation Techniques in Dynamic Environments”, Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 1397 – 1398, 2006.

[2] Blackwell, T. M., “Swarms in Dynamic Environments”, Proceedings of Genetic and Evolutionary Computation Conference, pp. 1-12, 2003.

[3] Blackwell, T. and Branke, J., ”Multi-swarm optimization in dynamic environments”, in G. R. Raidl, Applications of Evolutionary Computing, LNCS vol. 3005, pp. 489-500, Springer, 2004.

[4] Blackwell, T. and Branke, J., ”Multi-swarms, Exclusion and Anti-Convergence in Dynamic Environments”, IEEE Transactions on Evolutionary Computation, vol. 10/4, pp. 51-58, 2006.

[5] Boettcher, S. and Percus, A. G., “Extremal optimization: Methods derived from Co-Evolution”, Proceedings of the Genetic and Evolutionary Computation Conference, pp. 825-832, 1999.

[6] Branke, J., “Evolutionary Optimization in Dynamic Environments”, Kluwer, 2001.

[7] Branke, J., "Memory-Enhanced Evolutionary Algorithms for Changing Optimization Problems", Congress on Evolutionary Computation (CEC'99), vol.3 pp. 1875-1882, IEEE, 1999.

[8] Branke, J., ”The Moving Peaks Benchmark”, http://www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/, viewed 22/05/07.

[9] Branke, J., Kaußler, T., Schmidt, C., and Schmeck, H., “A multi-population approach to dynamic optimization problems”, in Adaptive Computing in Design and Manufacturing 2000, 2000.

[10] Branke, J. and Schmeck, H., “Designing evolutionary algorithms for dynamic optimization problems”. Advances in evolutionary computing: theory and applications, pp. 239-262, 2002.

[11] Bui, L.T., Branke, J. and Abbass, H. A., “Diversity As a Selection Pressure in Dynamic Environments”, in Proceedings of the 2005 conference on genetic and evolutionary computation, pp. 1557 – 1558, 2005.

[12] Bui, L.T., Branke, J. and Abbass, H. A., “Multiobjective optimization for dynamic environments”, in Congress on Evolutionary Computation, vol. 3, pp. 2349 – 2356, IEEE 2005.

[13] De Jong, K.A. and Morrison, R.W., “A Test Problem Generator for Non-Stationary Environments”, Proceedings of the Congress on Evolutionary Computation, pp. 2047-2053, 1999.

[14] Fentress, S. W, “Exaptation as a Means of Evolving Complex Solutions”, Master’s Thesis, University of Edinburgh, 2005.

[15] Janson, S. and Middendorf, M., “A Hierachical Particle Swarm Optimizer”, Proceedings of the Congress on Evolutionary Computation, pp. 770-776, 2003.

[16] Janson, S. and Middendorf, M., “A Hierachical Particle Swarm Optimizer for Dynamic Optimization Problems”, in G. R. Raidl, Applications of evolutionary computing, LNCS vol. 3005, pp. 513-524, Springer, 2004.

[17] Kramer, G. R. and Gallagher, J. C., “Improvements to the *CGA Enabling Online Intrinsic”, NASA/DoD Conference on Evolvable Hardware, p. 235, 2003.

[18] Li, X., “Adaptively choosing neighbourhood bests using species in a particle swarm optimzer for multimodal function optimization”, Proceedings of Genetic and Evolutionary Computation Conference 2004, pp. 105-116, 2004.

[19] Li, J., Balazs, M.E., Parks, G.T. and Clarkson, P.J., “A Species concerving Genetic Algorithm for Multimodal Function Optimization”, Evolutionary Computation, vol. 10(3), pp. 207-234, 2002.

[20] Li, X., Branke, J. and Blackwell, T., "Particle Swarm with Speciation and Adaptation in a Dynamic Environment ", in Proceedings of Genetic and Evolutionary Computation Conference, 2006.

[21] Lozano, M., Herrera, F., Krasnogor, N. and Molina, D., “Real-coded memetic algorithms with crossover hill-climbing”, Evolutionary Computation, vol. 12, pp. 273 – 302, 2004.

[22] Mendes, R. and Mohais, A., “DynDE: a differential evolution for dynamic optimization problems”, IEEE Congress on Evolutionary Computation, pp. 2808-2815, 2005.

[23] Meyer, K.D., Nasut, S.J. and Bishop, M., "Stochastic diffusion search: Partial function evaluation in swarm intelligence dynamic optimisation", Studies in Computational Intelligence, vol 31, 2006.

[24] Moser, I. and Hendtlass, T., "A Simple and Efficient Multi-Component Algorithm for Solving Dynamic Function Optimisation Problems", IEEE CEC 2007, to be published.

[25] Parrott, D. and Li, X., "A Particle Swarm Model for Tracking Multiple Peaks in a Dynamic Environment using Speciation", Proceedings of the 2004 Congress on Evolutionary Computation, pp. 105-116, 2004.

[26] Ronnewinkel, C. and Martinetz, T., "Explicit speciation with few a priori parameters for dynamic optimization problems", in GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, pp. 31-34, 2001.

[27] Zou, X., Wang, M., Zhou, A. and Mckay, B., "Evolutionary Optimization based on Chaotic Sequence in Dynamic Environments", in Proceedings of the 2004 IEEE International Conference on Networking, 2004.