A New Neighborhood Topology for QUAntum Particle Swarm Optimization (QUAPSO)

Arnaud Flori
Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi, EA 3956)
Vitry-sur-Seine, France
arnaud.flori@u-pec.fr

Hamouche Oulhadj
Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi, EA 3956)
Vitry-sur-Seine, France
oulhadj@u-pec.fr

Patrick Siarry
Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi, EA 3956)
Vitry-sur-Seine, France
siarry@u-pec.fr

ABSTRACT

Swarm Intelligence (SI) is a behavior, used first by Beni and Wang, corresponding to a system working with single and self-organized agents, interacting the ones with each other. This operating concept is implemented in many algorithms. Developed by Kennedy, Eberhart and Shi, Particle Swarm Optimization (PSO) is one of them. Its behavior is based on the movements of birds swarm, and its effectiveness, for looking for the optimal solution of a given problem, is well established. Nevertheless, PSO is known for its weakness in local search. Moreover, the behavior of PSO strongly depends on internal parameters settings. In order to address these problems, we propose a new type of self-adaptive Quantum PSO (QPSO), called QUAntum Particle Swarm Optimization (QUAPSO), based on quantum superposition to set the velocity parameters and hybridized with a Kangaroo Algorithm in order to optimize its efficiency in local search. QUAPSO was compared with five known algorithms from the literature (classical PSO, Kangaroo Algorithm, Simulated Annealing Particle Swarm Optimization, Bat Algorithm and Simulated Annealing Gaussian Bat Algorithm) on a set of 19 test functions. The results show that QUAPSO outperforms the competing algorithms.

CCS CONCEPTS

• **Theory of computation** → **Mathematical optimization**; • **Computing methodologies** → **Modeling and simulation**;

KEYWORDS

Particle Swarm Optimization, Swarm intelligence algorithm, Auto-adaptive algorithm, Self-organization

ACM Reference Format:
Arnaud Flori, Hamouche Oulhadj, and Patrick Siarry. 2019. A New Neighborhood Topology for QUAntum Particle Swarm Optimization (QUAPSO). In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321949

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3321949

1 INTRODUCTION

Particle Swarm Optimization (PSO) is a population-based metaheuristic, belonging to the class of Swarm Intelligence (SI) algorithms, developed the first time by Kennedy and Eberhart [3]. The swarm is composed by N particles whose movement is influenced by three components:

(1) An inertial component, memory of the last movement, that encourages each particle to continue its progression in the current direction, modulated by ω.
(2) A cognitive component, that encourages each particle to follow its best-known position, modulated by $C1$.
(3) A sociologic component, that encourages each particle to follow the best particle in a chosen neighborhood, modulated by $C2$.

The effectiveness of PSO is no longer to demonstrate: it is very efficient for global search and does not require that the optimization problem be differentiable. Nevertheless, PSO is not strong for local search and a local minimum can trap the particles of the swarm when there is stagnation of the best particle. Moreover, a poor choice of its parameters can induce premature convergence of PSO, producing sub-optimal solutions quite far from the best solution. In order to solve problems due to premature convergence and weakness for local search of PSO, we propose a new variant of PSO inspired by quantum superposition (the property of a quantum particle to be in several quantum states) and Kangaroo Algorithm, called QUAntum Particle Swarm Optimization (QUAPSO).

2 ALGORITHM PROPOSED

In QUAPSO, each particle of the swarm can have several velocity parameters, randomly selected among a range of values given in Table 1. The algorithm will set particle's position to the best solution given by the several states tested. The movement of a particle is still guided by the classical PSO equations, but QUAPSO allows the particles to visit multiple solutions at each iteration, and selects the best one. QUAPSO requires to set a new parameter Nc, corresponding to the number of parameter combinations tested but the velocity parameters are no longer to be set, which simplifies the setting of the algorithm.

Table 1: Range of the parameters' values selected

Parameter	Minimum value	Maximum value
ω	0	1
$C1$	0	2
$C2$	0	2

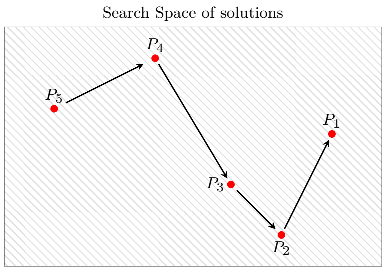

Figure 1: New neighborhood topology - each particle is represented by a red point and a number corresponding to the quality of its solution, the black arrows represent the sociologic component of the movements of the particles

Besides, the swarm is divided in two sub-swarms, the second is focused on looking for a better solution by the Kangaroo method into a hypersphere around the best solution found. When a test fails, hypersphere radius increases by 10%, otherwise, it decreases by 10%. This mechanism helps QUAPSO to refine the best solution found by the first sub-swarm.

Concerning the neighborhood topology, we developed two versions of QUAPSO. The first one (QUAPSO.1) uses a classical star topology, while QUAPSO.2 uses a new neighborhood topology: the first sub-swarm is sorted by the values given by the function fitness of each particle. Then, each particle will follow its own best particle in its sociologic neighborhood. This behavior is illustrated by Figure 1. Moreover, 3 "specialized particles" from the first sub-swarm are focused on the best particle of the swarm.

3 NUMERICAL EXPERIMENTS

We have tested QUAPSO on a set of 19 benchmark functions, selected from the same library [2]. The performance of QUAPSO was also compared with that of five competing algorithms: Kangaroo Algorithm (KA), Particle Swarm Optimization (PSO), Simulated Annealing Particle Swarm Optimization (SAPSO), Bat Algorithm (BA) and Simulated Annealing Gaussian Bat Algorithm (SAGBA). Each algorithm was executed 50 times and each run ends when the stopping criterion, defined for 80000 visited solutions, is reached. Regarding the settings of the algorithms, the swarm's size of each population-based algorithm is 40, and the size of the sub-swarms, for QUAPSO.1 and QUAPSO.2, is 20. The number of attempts for KA, and states tested for QUAPSO.1 and QUAPSO.2 is 5. The neighborhood topology used by PSO and QUAPSO.1 is a star topology, while that of QUAPSO.2 is the new neighborhood topology described earlier in this article.

Table 2: Average performance of the competing algorithms. "-" means the algorithm did not find the optimal solution and best results in bold allow to identify the most efficient algorithm on each function.

Functions	D	PSO	KA	BA	SAPSO	SAGBA	QUAPSO.1	QUAPSO.2
Ackley	2	4.20E-02	3.79E-02	1.80E-03	1.57E-03	6.19E-04	2.62E-04	**1.27E-05**
Beale	2	4.41E-05	7.62E-06	2.28E-04	–	1.41E-08	3.59E-10	**1.67E-11**
Bohachevsky	2	3.34E-02	4.02E-02	1.47E-06	2.27E-05	**1.16E-08**	2.46E-06	5.88E-08
Booth	2	8.84E-05	4.56E-05	5.55E-08	–	9.83E-07	3.50E-09	**8.04E-11**
Branin	2	**3.98E-01**	8.02E+00	7.72E-01	–	**3.98E-01**	**3.98E-01**	**3.98E-01**
Dixon & Price	25	5.58E-05.	1.08E+03	–	–	–	4.06E-09	**1.39E-10**
Goldstein & Price	2	**3.00E+00**	**3.00E+00**	3.04E+00	–	**3.00E+00**	**3.00E+00**	**3.00E+00**
Griewank	2	1.12E-02	9.92E-03	**1.44E-10**	5.78E-08	3.64E-08	4.18E-03	1.78E-03
Hartmann	3	-3.75E+00	-2.37E-01	**-3.86E+00**	-3.77E+00	**-3.86E+00**	**-3.86E+00**	**-3.86E+00**
Hump	2	2.41E-05	2.75E-05	5.06E-04	5.02E-08	**6.15E-10**	4.82E-08	4.65E-08
Levy	2	4.36E-06	8.88E-01	2.53E-09	3.57E-06	1.76E-10	3.18E-10	**4.63E-12**
Matyas	2	2.28E-06	2.05E+00	3.03E-10	2.51E-06	3.79E-09	1.24E-10	**1.47E-11**
Powell	24	3.14E+02	1.36E+01	–	–	–	1.15E-02	**5.36E-03**
Rastrigin	2	1.12E-03	5.12E-02	5.60E-05	1.94E-04	1.58E-07	**5.74E-08**	9.95E-02
Rosenbrock	2	2.54E-04	2.51E-07	7.87E-04	3.54E-03	4.68E-06	6.47E-09	**6.77E-10**
Shubert	2	**-1.87E+02**	-8.11E-03	**-1.87E+02**	–	**-1.87E+02**	**-1.87E+02**	**-1.87E+02**
Sphere	30	1.07E+01	6.37E+01	1.92E-003	4.86E-02	4.61E-05	1.03E-05	**1.65E-06**
Sum Squares	20	3.42E+01	4.51E+02	–	–	3.60E-04	7.06E-05	**1.35E-05**
Zakharov	2	4.75E-05	1.80E-05	9.23E-09	8.78E-03	6.12E-10	9.60E-10	**1.45E-11**

Table 2 provides the average performance for PSO, KA, SAPSO, SAGBA and the two variants of QUAPSO on the 19 test functions. For BA, SAPSO and SAGBA algorithms, the results correspond directly to those published by the authors on this set of 19 test functions [1]. The results show that both QUAPSO.1 and QUAPSO.2 are better than KA, PSO and SAPSO on all test functions, except Griewank and Rastrigin test functions. Moreover, QUAPSO.2 outperforms QUAPSO.1 on 18 test functions. Concerning the other competing algorithms, QUAPSO.1 is defeated against BA and SAGBA on 5 test functions, and QUAPSO.2 still remains defeated on 4 functions. The evaluated algorithms belonging to the class of stochastic algorithms, we also carried out 3 different statistical tests, when the statistical data were provided by the authors. First, we performed a Level test, to determine the heteroscedasticity of data. If heteroscedasticity is confirmed, results have different variabilities and we performed a Kolmogorov-Smirnov test, otherwise, we used a Wilcoxon-Mann-Whitney test. Thereby, we studied the results comparing performances of the two versions of QUAPSO.1, KA and PSO. QUAPSO.1's performances were statistically significant on every comparison, then we compared and studied the results of the two variants of QUAPSO. The p-values computed between QUAPSO.1 and QUAPSO.2 show that all differences measured are statistically significant. That confirms the efficiency of the new neighborhood implemented. However, there is still room for improvement by analysing, for example, the effects of the settings.

REFERENCES

[1] Xing-shi He, Wen-Jing Ding, and Xin-She Yang. 2012. Bat algorithm based on simulated annealing and Gaussian perturbations. *Neural Computing and Applications* 25 (2012), 459–468.
[2] Abdel-Rahman Hedar. 2005. Test functions for unconstrained global optimization [DB/OL]. (2005). http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm
[3] James Kennedy and Russel Eberhart. 1995. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4. Perth, WA, Australia, 1942–1948.