Intelligent Information Management, **2012, 4, 390-395**
http://dx.doi.org/10.4236/iim.2012.46043 Published Online November 2012 (http://www.SciRP.org/journal/iim)

A Quantum Behaved Gravitational Search Algorithm

Mohadeseh Soleimanpour Moghadam, Hossein Nezamabadi-Pour, Malihe M. Farsangi
Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran
Email: soleimanpour@yahoo.com, nezam@mail.uk.ac.ir, mmaghfoori@mail.uk.ac.ir

Received July 31, 2012; revised August 31, 2012; accepted September 15, 2012

ABSTRACT

Gravitational search algorithm (GSA) is a recent introduced global convergence guaranteed algorithm. In this paper, a quantum-behaved gravitational search algorithm, namely called as QGSA, is proposed. In the proposed QGSA each individual mass moves in a Delta potential well in feasible search space with a center which is weighted average of all *k*bests. The QGSA is tested on several benchmark functions and compared with the GSA. It is shown that the quantum-behaved gravitational search algorithm has faster convergence speed with good precision, and thus generating a better performance.

Keywords: GSA; Quantum Mechanics; *k*best; QGSA

1. Introduction

The objective of global optimization is to find the globally best solution of (possibly nonlinear) models, in the search space. Formally, global optimization seeks global solution of a constrained optimization model. Nonlinear models are ubiquitous in many applications, e.g., in advanced engineering design, data analysis, financial planning, scientific modeling, and others. These solutions often require a global search approach.

The field of swarm intelligence is an emerging research area that presents features of self-organization and cooperation principles among group members bio-inspired on social societies. Swarm Intelligence (SI) is the property of a system whereby the collective behaviours of agents interacting locally with their environment cause coherent functional global patterns to emerge.

Gravitational search algorithm (GSA), is a population based swarm optimization technique which was introduced by Rashedi *et al.* in 2009 [1], is a population based evolutionary optimization inspired by law of gravity. Similarly to genetic algorithms [2], GSA is an optimization tool based on a population, where each member is seen as a mass, and each mass is a potential solution to the problem under analysis. It has already been shown that GSA is comparable in performance with other evolutionary algorithms such as particle swarm optimization (PSO) [3] and Genetic Algorithm (GA) [2].

The basic idea of the GSA is to mimic the physical attraction between masses. Agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a global movement of all objects towards the objects with heavier masses.

At the end of the 19th century, classical mechanics encountered major difficulties in describing motions of microscopic particles with extremely light masses and extremely high velocities, and the physical phenomena related to such motions. These aspects forced scientists to rethink the applicability of classical mechanics and lead to fundamental changes in their traditional understanding of the nature of motions of microscopic objects [4]. The studies of Bohr, de Broglie, Schrödinger, Heisenberg and Bohn in 1920s inspire the conception of a new area, the quantum mechanics [5].

Recently, the concepts of quantum mechanics and physics have motivated the generation of optimization methods. Inspired by the GSA and quantum mechanics theories, this work presents a new Quantum-behaved GSA (QGSA) approach with outstanding result for unimodal functions.

The rest of the paper is organized as follows: Section 2 describes the features of classical GSA for continuous optimization, while Sections 3 explains the QGSA. Section 4 presents the results of the optimization. Lastly, some conclusion remarks are given in Section 5.

2. Classical Gravitational Search Algorithm

GSA is a global search strategy that can handle efficiently arbitrary optimization problems. Inspired by law of gravity, Rashedi *et al.* introduced the GSA method which is based on the Newtonian gravity. In the GSA, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a global movement of all objects towards the objects with heavier masses.

The procedure for implementing the classical version of GSA is given by the following steps:

Step 1. Initialization of agent positions: Initialize a population (array) of agents with random positions in the D-dimensional problem space using uniform probability distribution function. To this aim, the GSA consider a system with N agents (masses), the position of the i th agent is defined as follows:

$$X_i = \left(x_i^1, \cdots, x_i^d, \cdots, x_i^n\right), \quad i = 1, 2, \cdots, N \tag{1}$$

where x_i^d is the position of i th mass in the d th dimension and D is dimension of the search space.

Step 2. Evaluation of agent's fitness: Evaluate each agent's fitness value.

Step 3. Update $G(t)$, $\text{best}(t)$, $\text{worst}(t)$, $M_i(t)$ for $i = 1, 2, \cdots, N$: In GSA, the gravitational constant, G, will take an initial value, G_0, and it will be reduced by time:

$$G(t) = G(G_0, t) \tag{2}$$

Based on [1], the mass of each agent is calculated after computing current population's fitness as follows:

$$q_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \tag{3}$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^{s} q_j(t)} \tag{4}$$

where $M_i(t)$ and $\text{fit}_i(t)$ represent the mass and the fitness value of the agent i at t, and, $\text{worst}(t)$ and $\text{best}(t)$ are defined as follows (for a maximization problem):

$$\text{best}(t) = \max_{j \in \{1, \cdots, s\}} \text{fit}_j(t) \tag{5}$$

$$\text{worst}(t) = \min_{j \in \{1, \cdots, s\}} \text{fit}_j(t) \tag{6}$$

Step 4. Calculation of the total force in different directions: The total forces from a set of heavier masses that apply on agent i is calculated based on law of gravity as follow:

$$F_i^d(t) = \sum_{j \in k\text{best}, j \neq i} \text{rand}_j G(t) \frac{M_j(t) M_i(t)}{R_{ij}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t)\right) \tag{7}$$

where rand_j is a uniform random in the interval $[0,1]$, ε is a small value, and $R_{ij}(t)$ is the Euclidian distance between two agents i and j defined as $R_{ij}(t) = \left\| X_i(t), X_j(t) \right\|_2$. $k$best is the set of first K agents with the best fitness value and biggest mass. $k$best is a function of time, initialized to K_0 at the beginning and decreasing with time. Here, K_0 is set to N (total number of agents) and is decreased linearly to 1.

Step 5. Calculation of acceleration and velocity: Agent acceleration is calculated by using law of motion as follow:

$$\begin{aligned} a_i^d(t) &= \frac{F_i^d(t)}{M_i(t)} \\ &= \sum_{j \in k\text{best}, j \neq i} \text{rand}_j G(t) \frac{M_j(t)}{R_{ij}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t)\right) \end{aligned} \tag{8}$$

where rand_i is a uniform random in the interval $[0,1]$.

Then, the next velocity of an agent is calculated as a fraction of its current velocity added to its acceleration.

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \tag{9}$$

Step 6. Updating agents' position: Change the agent's position, x_i, according to Equation (10).

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{10}$$

Step 7. Repeating the evolutionary cycle: Repeat steps 2 to 6 until the stop criteria is reached, usually a sufficiently good fitness or a maximum number of iterations (generations).

3. Quantum Formulation of the Agent Dynamics

As per classical mechanics, an individual is depicted by its position vector x_i and velocity vector v_i, which determine the trajectory of the individual. The individual moves along a determined trajectory in Newtonian mechanics, but this is not the case in quantum mechanics. In quantum world, the term trajectory is meaningless, because x_i and v_i of an individual cannot be determined simultaneously according to uncertainty principle. Therefore, if individual agent in a GSA system has quantum behavior, the GSA algorithm is bound to work in a different fashion.

In quantum time-space framework, the quantum state of an individual is depicted by wave function (ψ) instead of position. Q is the probability that measurement of the individual's position finds it about the point whereby $|\psi|^2 = Q$.

In quantum mechanics, the governing equation is the general time-dependent Schrödinger equation,

$$j\hbar \frac{\partial}{\partial t} \psi(r,t) = H(r) \psi(r,t) \tag{11}$$

where $H(r)$ is a time-independent Hamiltonian operator given by,

$$H(r) = -\frac{\hbar^2}{2m} \nabla^2 + V(r) \tag{12}$$

where $\hbar$ is Planck's constant, m is the mass of the individual, and $V(r)$ is the potential energy distribution. Its amplitude squared is the probability that measurement of the individual's position at the time t finds it in the volume element about the desired point. By imposing the following normalization condition we can justify such a measure,

$$\int_{-\infty}^{+\infty} |\psi|^2 \,\mathrm{d}r = \int_{-\infty}^{+\infty} Q \mathrm{d}r = 1 \tag{13}$$

Now, we hypothesize that the GSA system is a quantum system, each mass of which is of quantum state formulated by wave function. Inspired by analysis of convergence of the classical GSA in [1], we assume that an individual mass moves in a Delta potential well in feasible search space, of which the center is weighted average of all *k*best. For simplicity, we consider an agent in one-dimensional space, the potential energy of the mass in one-dimensional Delta potential well is represented as:

$$V(r) = -\gamma\delta(r) \tag{14}$$

where γ is a positive number proportional to the "depth" of the potential well. The depth is infinite at the origin and zero elsewhere.

The Schrodinger equation for the model is,

$$E\psi(r) = \left(-\frac{\hbar^2}{2m}\nabla^2 - \gamma\delta(r)\right)\psi(r) \tag{15}$$

For $r \neq 0$ the equation can be written as,

$$\frac{\mathrm{d}^2\psi}{\mathrm{d}r^2} + \frac{2m}{\hbar^2}E\psi = 0 \tag{16}$$

In order to prevent diverging of agents the following boundary conditions is applied

$$|r| \to \infty \leftrightarrow \psi \to 0 \tag{17}$$

To satisfy the bound condition, Equation (16) is calculated as follow:

$$\psi(r) = \mathrm{e}^{\frac{\sqrt{-2mE}}{\hbar}|r|} \text{ when } r \neq 0 \tag{18}$$

QGSA is the integration of quantum computing and GSA. The QGSA is based on the representation of the quantum state vector.

To evaluate the fitness value, we need to learn of precise information of position of the agent. However, the quantum wave function can only give the probability density function that the mass appears at the desired position. So we have to gauge the position of the individual, which is called collapsing the quantum state to the classical state. According to Monte Carlo Method of uncertainty, it is possible to simulate the process of measurement. The procedure of simulation is described as follows.

To this aim, a random variable rand is generated uniformly distributed between 0 and 1, so Equation (18) can be simplified with substituting a random number instead of it,

$$Q(r) = |\psi(r)|^2 = \mathrm{e}^{\frac{2\sqrt{-2mE}}{\hbar}|r|} = \mathrm{rand} \tag{19}$$

where rand is a random number uniformly distributed on $(0,1)$.

$$r = x - \mathrm{best} = \pm\frac{\hbar}{2\sqrt{-2mE}}\ln(\mathrm{rand}) \tag{20}$$

Thus, the position of agent accurately is measured as follows,

$$x = \mathrm{best} \pm \frac{\hbar}{2\sqrt{-2mE}}\ln(\mathrm{rand}) \tag{21}$$

where $\frac{\hbar}{2\sqrt{-2mE}}$ is the only parameter of the algorithm.

The procedure for implementing the QGSA is given by the following steps:

Step 1. Initialization of mass positions: Initialize a population (array) of masses with random positions in the D-dimensional problem space using a uniform probability distribution function.

Step 2. Evaluation of particle's fitness: Evaluate the fitness value of each agent.

Step 3. Selecting *k*best and update *P*best: *k*best is the set of first K agents with the best fitness value and biggest mass. In order to updating *P*best, compare each agent's fitness with the agent's *P*best. If the current value is better than *P*best, then set the *P*best value equal to the current value and the *P*best location equal to the current location in D-dimensional space.

Step 4: Updating the $M\mathrm{best}_i$ using Equation (23):

$$\mathrm{distance}_{i,l} = \left\|X_i - k\mathrm{best}_l\right\| \tag{22}$$

$$M\mathrm{best}_i = \frac{\sum_{j=1}^{K}\frac{1}{\mathrm{distance}_{i,j}}\cdot k\mathrm{best}_j}{\sum_{l=1}^{K}\frac{1}{\mathrm{distance}_{i,l}}} \tag{23}$$

Step 5. Updating of masses' position: Change the position of the mass where c_1 and c_2 are two random numbers generated using a uniform probability distribution in the range $[0,1]$.

$$\mathrm{Best}_i^d = \frac{c_1 M\mathrm{best}_i^d + c_2 P\mathrm{best}_i^d}{c_1 + c_2} \tag{24}$$

Employing the Monte Carlo method, the masses move according to the following iterative equation:

$$\begin{cases} X_i(t+1) = \text{Best}_i + \delta \cdot \left|\text{Best}_i - X_i(t)\right| \cdot \ln\left(\dfrac{1}{\text{rand}}\right) & \text{if } S \geq 0.5 \\ X_i(t+1) = \text{Best}_i - \delta \cdot \left|\text{Best}_i - X_i(t)\right| \cdot \ln\left(\dfrac{1}{\text{rand}}\right) & \text{if } S < 0.5 \end{cases} \tag{25}$$

where δ is a design parameter called contraction-expansion coefficient, rand and S are values generated according to a uniform probability distribution in range $[0,1]$.

Step 6. Repeating the evolutionary cycle: Loop to Step 2 until a stop criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

4. Numerical Results

In this study, to evaluate the performance of the QGSA, 7 minimization benchmark functions [1,4] listed in **Table 1** are used for comparison with classical GSA. In **Table 1**, D is the dimension of search space (function), f_{opt} is the minimum value of the function. The minimum value (f_{opt}) of the functions of **Table 1** are zero. The test functions are unimodal high-dimensional functions which make them suitable for evaluating performance of the algorithm in point of view convergence rate.

We evaluate the unimodal optimization algorithm in two aspects: quality, and precision of solutions.

Quantity: Number of needed fitness function evaluations to reach the global optimum. It means the convergence rate of search algorithm.

Precision or accuracy: The difference between the global minimum and one but lowest minimum determines the possibility to detect the global minimum point.

Table 1. Benchmark functions used in this study.

Test functions	Feasible spaces	D	f_{opt}
$f_1 = \sum_{i=1}^{n} x_i^2$	$[-100,100]^n$	30	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10,10]^n$	30	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_i \right)^2$	$[-100,100]^n$	30	0
$f_4(x) = \max\{\lvert x_i \rvert, i = 1,2,\cdots,n\}$	$[-100,100]^n$	30	0
$f_5 = \sum_{i=1}^{n-1} \left[100\left(x_i^2 - x_{i+1}\right)^2 + \left(x_i - 1\right)^2 \right]$	$[-5,10]^n$	30	0
$f_6 = \sum_{i=1}^{n} \left(x_i + 0.5\right)^2$	$[-100,100]^n$	30	0
$f_7(x) = \sum_{i=1}^{n} x_i^4 + \text{random}[0,1)$	$[-1.28,1.28]^n$	30	0

Table 2. Minimization result of functions in table 1. Results are averaged over 30 runs and the average best-so-far and standard deviation of best solution obtained at last iteration are given.

		Classical GSA	Quantum GSA
f_1	The average best-so-far solution median of the best solution the standard deviation	7.3e−11 7.1e−11 4.4e−25	0 0 0
f_2	The average best-so-far solution median of the best solution the standard deviation	4.03e−5 4.07e−5 1.38e−013	0 0 0
f_3	The average best-so-far solution median of the best solution the standard deviation	0.16e+3 0.15e+3 1.05e+4	0 0 0
f_4	The average best-so-far solution median of the best solution the standard deviation	3.7e−6 3.7e−6 1.4e−4	0 0 0
f_5	The average best-so-far solution median of the best solution the standard deviation	25.27 25.17 181.83	1.112 0 33.156
f_6	The average best-so-far solution median of the best solution the standard deviation	8.3e−11 7.7e−11 2.41e−32	1.176e−3 0 7.87e−3
f_7	The average best-so-far solution median of the best solution the standard deviation	0.018 0.015 1.02e−4	0 0 0

In unimodal functions, the convergence rate of search algorithm is more important than the final results because there are other methods which are specifically designed to optimize unimodal functions.

The results are averaged over 30 runs and the average best-so-far solution, median of the best solution and the standard deviation in the last iteration are reported for unimodal functions in **Table 2**. As this table illustrates QGSA provides better results than GSA for all functions. In these function QGSA tends to find the global optimum faster than classical GSA and hence has a higher convergence rate.

The numerical results in **Table 2** show that almost the QGSA could hit the optimal solution with high precision.

There are two exception test functions which QGSA cannot tune itself and have not a good performance.

In Rosenbrock function, the global optimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, so converging to the global minimum, however, is difficult. Hence QGSA could not tune itself.

Also, the good convergence rate of QGSA could be concluded from **Figures 1** to **7**. According to these figures, QGSA tends to find the global optimum in an acceptable time hence has a high convergence rate.

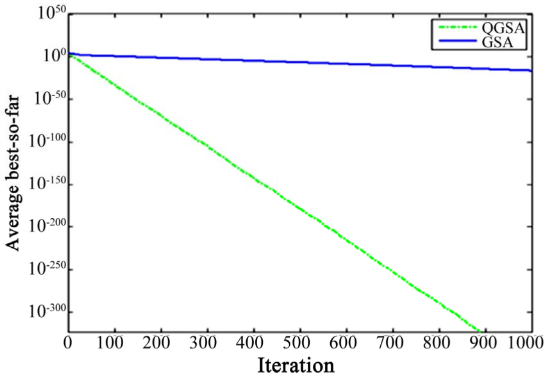

Figure 1. Comparison of performance of QGSA and GSA for minimization of $f_1(x)$.

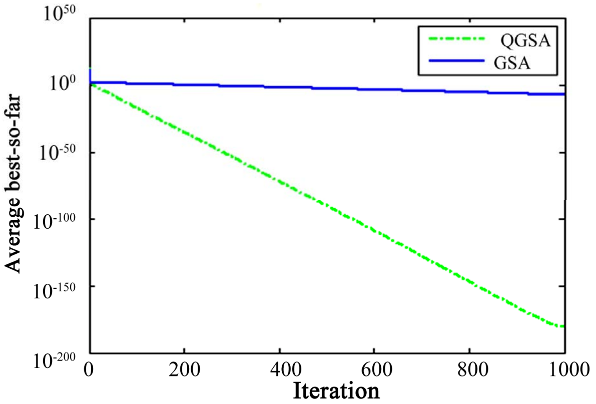

Figure 2. Comparison of performance of QGSA and GSA for minimization of $f_2(x)$.

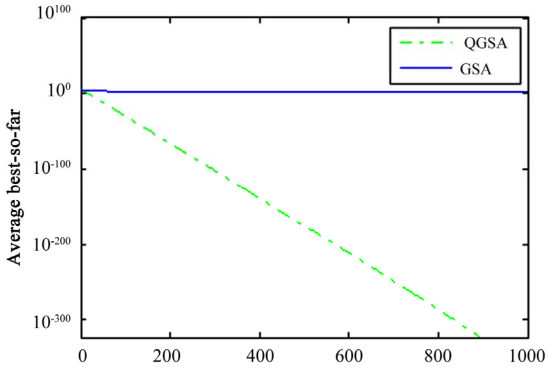

Figure 3. Comparison of performance of QGSA and GSA for minimization of $f_3(x)$.

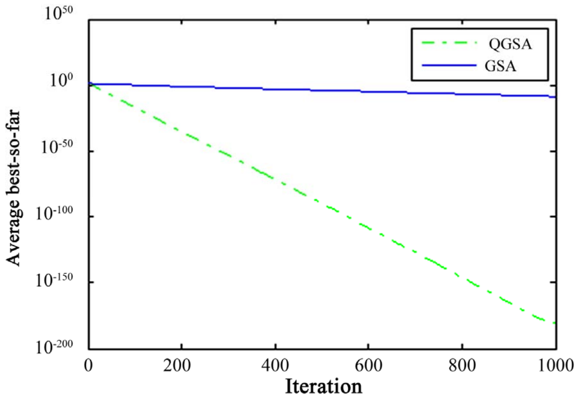

Figure 4. Comparison of performance of QGSA and GSA for minimization of $f_4(x)$.

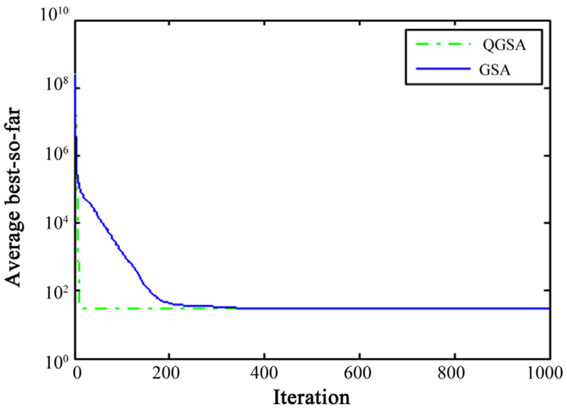

Figure 5. Comparison of performance of QGSA and GSA for minimization of $f_5(x)$.

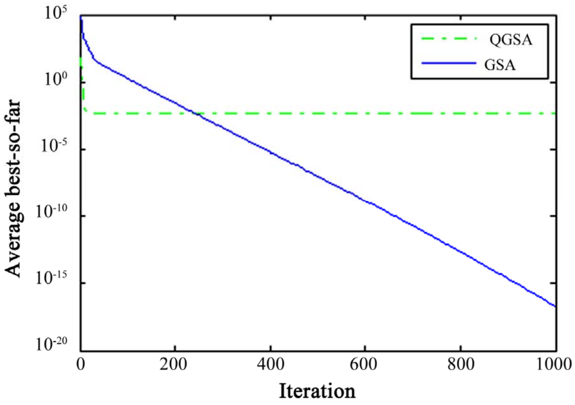

Figure 6. Comparison of performance of QGSA and GSA for minimization of $f_6(x)$.

The reported results for dimension 30 by for different test functions are summarized in **Table 2**. According to **Figures 1** to **7**, we can see that QGSA provides better solutions except for $f_6(x)$.

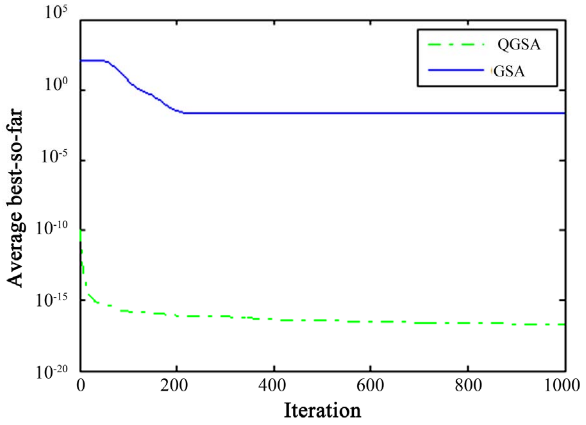

Figure 7. Comparison of performance of QGSA and GSA for minimization of $f_7(x)$.

5. Conclusion

In this paper, a quantum version of gravitational search algorithm is introduced called as QGSA. It is tasted on different benchmark functions to investigate the efficiency of QGSA. The results show that QGSA is performing much better than GSA in finding the optimum result. Although the used functions are unimodal functions but it could expand it to multimodal functions. Currently, the authors are working on the improvement of the proposed QGSA.

REFERENCES

[1] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Science*, Vol. 179, No. 13, 2009, pp. 2232-2248. doi:10.1016/j.ins.2009.03.004

[2] K. S. Tang, K. F. Man, S. Kwong and Q. He, "Genetic Algorithms and Their Applications," *IEEE Signal Processing Magazine*, Vol. 13, No. 6, 1996, pp. 22-37 doi:10.1109/79.543973

[3] F. V. D. Bergh and A. P. Engelbrecht, "A Study of Particle Swarm Optimization Particle Trajectories," *Information Sciences*, Vol. 176, No. 8, 2006, pp. 937-971. doi:10.1016/j.ins.2005.02.003

[4] X. F. Pang, "Quantum Mechanics in Nonlinear Systems," World Scientific Publishing Company, River Edge, 2005. doi:10.1142/9789812567789

[5] W. Schweizer, "Numerical Quantum Dynamics," Hingham, 2001.

[6] M. Dorigo, V. Maniezzo and A. Colorni, "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, Vol. 26, 1, 1996, pp. 29-41. doi:10.1109/3477.484436