Peertechz

Engineering Group

Trends in Computer Science and Information Technology

DOI http://dx.doi.org/10.17352/tcsit.000004

Ransikarn Ngambusabongsopa[1], Vincent Havyarimana[2]* and Zhiyong Li

[1]College of Computer Science and Electronic Engineering, Hunan University, 410082, Changsha, China

[2]Departemnt of Applied Sciences, Ecole Normale Supérieure, 6983, Bujumbura, Burundi

Dates: Received: 28 February, 2017; **Accepted:** 06 March, 2017; **Published:** 07 March, 2017

***Corresponding author:** Havyarimana Vincent, Departemnt of Applied Sciences, Ecole Normale Supérieure, 6983, Bujumbura, Burundi, Tel: (+257) 79830258; Fax: (+257) 22258948; E-mail: havincent12@gmail.com ; havicent14@hnu.edu.cn

Keywords: Chemical reaction optimization; Maximization; Mutation operator; Polynomial mutation

https://www.peertechz.com

Research Article

A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms

Abstract

Several approaches are proposed to solve global numerical optimization problems. Most of researchers have experimented the robustness of their algorithms by generating the result based on minimization aspect. In this paper, we focus on maximization problems by using several hybrid chemical reaction optimization algorithms including orthogonal chemical reaction optimization (OCRO), hybrid algorithm based on particle swarm and chemical reaction optimization (HP-CRO), real-coded chemical reaction optimization (RCCRO) and hybrid mutation chemical reaction optimization algorithm (MCRO), which showed success in minimization. The aim of this paper is to demonstrate that the approaches inspired by chemical reaction optimization are not only limited to minimization, but also are suitable for maximization. Moreover, experiment comparison related to other maximization algorithms is presented and discussed.

Introduction

Optimization [1] is prevalent in various field of science, engineering, economics and other related topics. Since the past few decades, plenty optimization frameworks were proposed to solve existing optimization problems. They depend on using a predefined constraint which is involved in the area of the search space to find variables of the functions to be optimized. In general, an optimization problem includes minimizing or maximizing a function systematically by selecting input values from a given feasible set [2]. The most famous framework is the evolution algorithm (EA), EA is heuristic algorithm which is inspired by the nature of the biological evolution and the social behavior of species. Several evolutionary algorithms have been addressed to optimization including Simulated Annealing (SA). It is inspired by annealing in metallurgy or physical process of increasing the crystal size of a material and reducing the defects through a controllable cooling procedure [3]. Furthermore, the genetic algorithm (GA) [4], is affected by Darwinian principle of the 'survival of the fittest' and the natural process of evolution through reproduction. Memetic algorithm (MA) [5], is inspired by Dawkins' notion of a meme. However, particle swarm optimization (PSO) [6], is developed from social behavior of bird flocking or fish schooling by Eberhart and Kennedy. Ant colony optimization (ACO) [7], mimics ants which are able to discover the shortest route between their nest, and a source of food. Shuffled frog leaping algorithm (SFL) [8], is introduced to combine the benefits of the genetic based and the social behavior-based PSO. Bat algorithms (BAs) are inspired by the echolocation behavior of bats [9]. Harmony search (HS) [10], which is based on natural musical performance processes, occurs when a musician searches for an optimal state of harmony. Finally, chemical reaction optimization (CRO) is developed and proposed by Lam and Li [11,12], which simulate the effective drive to molecules in chemical reaction.

Although there are abundant approaches suggested to solve optimization problems, researchers still measure the capability of algorithms by comparing the optimal results generated with the previous methods in minimization aspect. They have corroborated that the published algorithms are suitable for solving minimization problems. Nevertheless, these are unwarranted to be appropriate for absolutely all optimization problems. As a result, the simulation based on maximization is considered in this paper. We also intend to solve maximization problems by experimenting several optimization algorithms based on CRO framework. These algorithms include an effective version of RCCRO (i.e., RCCRO4), HP-CRO2 a best version of hybrid algorithm based on PSO with CRO, OCRO which is a hybrid orthogonal crossover with CRO, and a recent established algorithm MCRO which is hybrid polynomial mutation operator with CRO.

The rest of this paper is organized as follows: a brief review of optimization algorithms based on CRO is introduced

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

in Section 2. In Section 3, optimization problem functions and evaluation methods which are used for measuring the performance of algorithms are presented. The experimental results and discussions for different experiments are illustrated in Section 4. Finally, Section 5 highlights conclusion and our future work.

Optimization algorithms based on CRO

CRO [11,12], is a framework that mimics molecular interactions in chemical reactions to reach a low-energy stable state. Potential energy is the energy stored in a molecule with respect to its molecular configuration; the system becomes disordered when potential energy is converted to other forms [11]. Molecules stored in a container are vital to the manipulation of agents. Each molecule contains a profile that includes several attributes, such as molecular structure (ω), current potential energy (PE), and current kinetic energy (KE). In CRO reaction, the initial reactants in high-energy states incur a sequence of collisions. Molecules collide either with other molecules or with the walls of the container, pass through energy barriers, and become the final products in low-energy stable states. Typically, CRO includes three phases: initialization, iteration, and the final phase. In each run, the algorithm begins with initializing the population or the molecules contained in a container, and a particular number of iterations are, then, performed. When it satisfies the stopping criteria, the global optimal solution is presented.

This section provides a brief of RCCRO which is represented as the best research of original CRO, and several hybrid optimization algorithms based on CRO have practiced the performance in minimization aspect: HP-CRO, OCRO and MCRO.

Real-coded chemical reaction optimization (RCCRO): The concept of CRO is captured from the phenomenon of driving high-energy molecules to stable states through various types of elementary reactions. In CRO algorithm, four types of elementary reactions can take place: On-wall ineffective collision, Intermolecular ineffective collision, Decomposition, and Synthesis. On-wall and intermolecular ineffective collisions are local searches, whereas decomposition and synthesis are global searches. The characteristics and descriptions of these four elementary reactions are as follow [12,13].

On-wall ineffective collision is the reaction created when a molecule hits the wall of a container and then bounces back. This reaction only slightly changes the molecular structure (ω) when $PE_{\omega} + KE_{\omega} \ge PE_{\omega}$ occurs. The new molecular structure ($\acute{\omega}$) is produced by neighborhood search operator as $\acute{\omega}$ = neighborhood (ω). The central energy buffer is updated by extracting and storing a certain portion of its *KE*. The profile of the molecule is updated as $KE_{\acute{\omega}} = (PE_{\omega} - PE_{\acute{\omega}} + KE_{\omega}) \times r$, where r is a random number that r ω [*KELossRate*,*1*]. The new PE of new fitness is calculated by the new ω *or* $\acute{\omega}$ as $f(\acute{\omega})$.

Intermolecular ineffective collision refers to two or more molecules that collide with one another and then separate. This reaction also slightly changes the molecular structure similar to on-wall ineffective collision. The profiles of the molecules and the central energy buffer are updated when $PE_{\acute{\omega}1} + PE_{\acute{\omega}2} + KE_{\acute{\omega}1} + KE_{\acute{\omega}2} \ge PE_{\acute{\omega}1} + PE_{\acute{\omega}2}$. The molecular structures are produced from their own neighborhood by neighborhood search operator. The number of molecules are unchanged after the collision.

Decomposition represents the situation when a molecule hits the wall of a container and then splits into two or more molecules. This elementary reaction is applied to finish local search and explore other regions (i.e., global search). The profiles of the molecules and the central energy buffer are updated when $PE_{\omega} + KE_{\omega} \ge PE_{\acute{\omega}1} + PE_{\acute{\omega}2}$ and when the energy buffer is sufficient. This reaction significantly processes new molecular structures of the resultant molecules.

Synthesis occurs in the situation that two or more molecules collide and transform to a single molecule. The profiles of the molecules and the central energy buffer are updated when $PE_{\omega1} + PE_{\omega2} + KE_{\omega1} + KE_{\omega2} \ge PE_{\acute{\omega}}$. This reaction strongly and significantly alters the resultant molecular structure. The number of molecules are reduced after the collision.

RCCRO is the most powerful algorithm of original CRO. In addition, RCCRO4 is the best version of RCCRO. The algorithm of RCCRO4 is represente as algorithm 1.

Algorithm 1: RCCRO4 Algorithm

```
Input: Objective function f , constraints, and the dimensions of the problem
/* initialization phase*/
Assign parameter value to PopSize, InitialKE, StepSize, buffer, KELossRate,
    On-wallColl,
DecThres, SynThres, The set of molecules in this Container are molecules
    1, 2, . . . , PopSize
for each of the molecule  do
    Assign a random solution to the molecular structure ω. Calculate the PE
    by f (ω) and evaluate.
    Assign the KE with InitialKE
end for
/* Iterations  phase*/
while (the stopping criteria not met) do
    Get r randomly in interval  [0,1]
    if (r >  On-wallColl) then
        select  a molecule M from Container Randomly
        if (Decomposition criterion met) then
            Decomposition
            if Success then Add a new molecule M′ to Container
        else
            On-wallineffectiveCollision
        end if
    else
        select two molecule M1 and M2 from Container  Randomly
        if (Synthesis criterion met) then
            Synthesis
            if Success then Remove molecule M2 from Container
        else
            Inter-molecular Collision
        end if
    end if
    evaluate and keep any new optimal solution
end while
/* The Output phase */
Output the best solution and its function value
```


Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

Hybrid algorithm based on particle swarm and chemical reaction optimization (HP-CRO): The HP-CRO [14], is an approach for optimizing functions based on PSO and CRO algorithm. This algorithm presents the combination between PSOupdate operator and local search operators which make algorithm efficient. In addition, structure of the algorithm can easily control the whole search space to find global minimum based on the difference between the two boundary handling constraints. HP-CRO contains two elementary reactions: On-wall ineffective collision and Intermolecular ineffective collision. Several parameters are included in HP-CRO: inertia weight w = 0.729, local weight c_1 = 1.49445, global weight c_2 = 1.49445, r_1 and r_2 randomizes. There are two versions of HP-CRO which are HP-CRO1 and HP-CRO2, whereas the best version is HPCRO2. Algorithm of HP-CRO is demonstrated as Algorithm 2. For more details, HP-CRO is represented in [14], we note that we have changed the condition of algorithm to maximization operation.

Algorithm 2: HP-CRO Algorithm

Input: Objective function *f*, constraints, and the dimensions of the problem
1: \\ Initialization
2: Assign parameter values to *PopSize*, *KELossRate*, *Stepsize*, *buffer*, *InitialKE*, γ, *InterRate*,
w(inertia weight), c_1 (cognitive/local weight), c_2 (social/global weight).
3: Let *Pop* be the set of molecule (particle) 1, 2,. . ., *Popsize*
4: for each of molecules (particles) do
5: Assign random solution to the molecular structure (particle position) *w*
6: Calculate the fitness by *f*(*w*)
7: Set *PSOCoe* = 0
8: end for
9: \\ Iterations
10: while (*the stopping criteria not met*) do
11: Select a molecule M_w from *Pop* randomly
12: if $PSOCoe_{Mw}$ > γ then
13: $PSOUpdate(M_w)$
14: $PSOCoe_{Mw}$ = 0
15: else
16: Generate *r* randomly in interval [0,1]
17: if *r* > *InterRate* then
18: Randomly select molecule M_{w1}
19: IntermolecularIneffectiveCollision(M_w,M_{w1})
20: $PSOCoe_{Mw} = PSOCoe_{Mw} + 1$
21: $PSOCoe_{Mw1} = PSOCoe_{Mw1} + 1$
22: else
23: OnwallIneffectiveCollision(M_w)
24: $PSOCoe_{Mw} = PSOCoe_{Mw} + 1$
25: end if
26: end if
27: Check for any new optimal solution
28: end while
29: // The final phase
30: Output the best solution found and its objective function value

Orthogonal chemical reaction optimization (OCRO): In general, the OCRO [15], is an algorithm that hybrids quantization orthogonal crossover (QOX) and CRO. It creates new molecules by two types of elementary reaction on-wall ineffective collision and intermolecular ineffective collision as original CRO. Moreover, it uses QOX search operator to create new molecules. The two elementary reactions in CRO serve as local search while QOX is provided to work as a global search operator. However, synthesis and decomposition elementary reactions are not included in OCRO. The algorithm has three main phases including: 1) initialization, 2) iteration, and 3) the final phase as original CRO. In the iteration phase, the type of search is identified by judging whether a random number *t* is bigger than *Molecoll*. QOX search takes place when *t* is smaller than *Molecoll*. Otherwise, it may result in a local search included on-wall ineffective collision and intermolecular ineffective collision. Any new local minimum found are checked and recorded as a new global minimum. Later on, the judge determines whether it is an intermolecular collision. The final global minimum is presented as the optimal result. OCRO Algorithm is shown in Algorithm 3. See more description of OCRO in [15].

Algorithm 3: OCRO Algorithm

Input: Objective function *f*, constraints, and the dimensions of the problem
1. Assign parameter values to *PopSize*, *KELossRate*, *MoleColl*, *buffer*, *Stepsize* and *InitialKE*
2.Let *Pop* be the set of molecule 1, 2,.. ., *PopSize*
3.for each of the molecules do
4. Assign a random solution to the molecular structure *x*
5. Calculate the *PE* by *f*(*x*)
6. Assign the *KE* with *InitialKE*
7. *FE* = *PopSize*
8.end for
9.Assign buffer = 0
10. while the stopping criteria not met do
11. Get t randomly in interval [0,1]
12. if *t* < *Molecoll* then
13. select molecules M_1 and M_2 from *Pop* randomly
14. QOX(M_1, M_2)
15. *FE* = *FE* + 10;
16. else
17. Get *w* randomly in interval [0,1]
18. if *w* > 0.5 then
19. Select a molecules *M* from *Pop* randomly
20. On-wall ineffective collision (*M*)
21. *FE* = *FE* + 1;
22. else
23. select molecules M_1 and M_2 from *Pop* randomly
24. Intermolecular ineffective Collision (M_1,M_2)
25. *FE* = *FE* + 2;
26. end if
27. end if
28. Check for any new optimal solution
29. end while
30. Output the best solution found and its objective function value

A hybrid mutation chemical reaction optimization algorithm (MCRO): A hybrid optimization algorithm MCRO [16] combines mutation operator with CRO in which turning operator. The latter is also included in this algorithm. The most appreciate mutation operator in MCRO framework is polynomial mutation operator. The exclusivity of MCRO compared to the other two hybrid CRO algorithms (i.e., HP-CRO2 and OCRO) is that MCRO focuses on hybrid new contents specific into four elementary reactions of CRO. However, HP-CRO2 and OCRO interest hybrid new contents to core procedure in order to work instead of two elementary reactions (i.e., synthesis and decomposition) which serve as global search in RCCRO. Hence, the main algorithm of MCRO is similar to RCCRO4 but different in the sub-algorithm. The main procedures such as on-wall ineffective collision, intermolecular ineffective collision, decomposition, synthesis, and neighborhood search operator, are discussed as follow:

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

Turning operator is invented and merged into neighborhood search operator which works in three types of elementary reactions of RCCRO: on-wall ineffective collision, intermolecular ineffective collision, and decomposition. Turning operator transforms the molecular structure from the neighborhood of the operand to highly improve the optimal quality and reliability of the algorithm.

A mutation operator is applied to each new molecule in the initialization phase of algorithm and incorporated in changing the molecular structure. As a consequence, the results are judged before and after studied the performance of mutation operator and selected the better result to be recorded as a new global minimum. Mutation operator can spread the search space by randomly sampling new points and increases the opportunity of generating more ideal result not less than twice of RCCRO for every elementary reaction. Related algorithms are represented in algorithms 4, 5, 6 and 7, further information can be found in [16].

Algorithm 4: On-wall in effective Collision of MCRO

Input: a molecule *M* and buffer
1: Obtain $\acute{\omega}$ = *Neighbor* (ω) with turning operator
2: Calculate $PE_{\acute{\omega}}$ by $f(\acute{\omega})$
3: if $PE_{\omega} + KE_{\omega} \geq PE_{\acute{\omega}}$ then
4: Get *r* randomly in interval [*KE Loss Rate,* 1]
5: $KE_{\acute{\omega}} = (PE_{\omega} + KE_{\omega} - PE_{\acute{\omega}}) \times r$
6: Update *buffer* = *buffer* - $KE_{\acute{\omega}}$
7: do Mutation of ω to *temp*ω
8: Calculate the *tempPE* by *f*(*temp* ω)
9: if *tempPE* better than $PE_{\acute{\omega}}$ then Replace $\acute{\omega}$ with *temp*ω, $PE_{\acute{\omega}}$ with *tempPE*
11: Update the profile of *M* by $\omega = \acute{\omega}$, $PE_{\omega} = PE_{\acute{\omega}}$ and $KE_{\omega} = KE_{\acute{\omega}}$
12: end if
Output *M* and *buffer*

Algorithm 5: Inter-molecular Collision of MCRO

Input: molecules M_1, M_2
1: Obtain $\acute{\omega}_1$ = *Neighbor*(ω_1) and $\acute{\omega}_2$ = *Neighbor*(ω_2) with turning operator
2: Calculate $PE_{\acute{\omega}1}$ and $PE_{\acute{\omega}2}$
3: $temp_2 = (PE_{\acute{\omega}1} + PE_{\acute{\omega}2} + KE_{\acute{\omega}1} + KE_{\acute{\omega}2}) - (PE_{\acute{\omega}1} + PE_{\acute{\omega}2})$
4: if $PE_{\acute{\omega}1} + PE_{\acute{\omega}2} + KE_{\acute{\omega}1} + KE_{\acute{\omega}2} \geq PE_{\acute{\omega}1} + PE_{\acute{\omega}2}$ then
5: Get *r* randomly in interval [0, 1]
6: $KE_{\acute{\omega}1} = temp_2 \times p$ and $KE_{\acute{\omega}2} = temp_2 \times (1 - p)$
7: end if
8: do Mutation of ω_1 to *temp*ω_1 and ω_2 to *temp*ω_2
9: Calculate the *tempPE*$_1$ by f(*temp*ω_1) and *tempPE*$_2$ by f (*temp*ω_2)
10: If *tempPE*$_1$ better than $PE_{\acute{\omega}1}$: Replace $\acute{\omega}_1$ with *temp*ω_1 and $PE_{\acute{\omega}1}$ with *tempPE*$_1$
11: If *tempPE*$_2$ better than $PE_{\acute{\omega}2}$: Replace $\acute{\omega}_2$ with *temp*ω_2 and $PE_{\acute{\omega}2}$ with *tempPE*$_2$
12: Update the profile of M_1 by $\omega_1 = \acute{\omega}_1$, $PE_{\acute{\omega}1} = PE_{\acute{\omega}1}$ and $KE_{\acute{\omega}1} = KE_{\acute{\omega}1}$
13: Update the profile of M_2 by $\omega_2 = \acute{\omega}_2$, $PE_{\acute{\omega}2} = PE_{\acute{\omega}2}$ and $KE_{\acute{\omega}2} = KE_{\acute{\omega}2}$
14: Output M_1 and M_2

Algorithm 6: Decomposition of MCRO

Input: A molecule *M* and buffer
1: Obtain ω_1 and ω_2 from ω
2: Obtain $\acute{\omega}_1$ = *Neighbor* (ω_1) and $\acute{\omega}_2$ = *Neighbor* (ω_2) with turning operator
3: Calculate $PE_{\acute{\omega}1}$ and $PE_{\acute{\omega}2}$
4: $temp_1 = PE_{\omega} + KE_{\omega} - PE_{\acute{\omega}1} - PE_{\acute{\omega}2}$
5: Success=TRUE
6: if $PE_{\omega} + KE_{\omega} \geq PE_{\acute{\omega}1} + PE_{\acute{\omega}2}$ then
7: Get *k* randomly in interval [0, 1]
8: $KE_{\acute{\omega}1} = temp_1 \times k$ and $KE_{\acute{\omega}2} = temp_1 \times (1 - k)$
9: Create new molecules *M'*
10: else if *temp*1 + *buffer* ≥ 0 then
11: Get *r*1, *r*2, *r*3, and *r*4 randomly in interval [0, 1]
12: $KE_{\acute{\omega}1} = (temp1 + buffer) \times r1 \times r2$
13: $KE_{\acute{\omega}2} = (temp1 + buffer - KE_{\acute{\omega}1}) \times r3 \times r4$
14: Update *buffer* = *temp*1 + *buffer* − $KE_{\acute{\omega}1}$ − $KE_{\acute{\omega}2}$
15: Create new molecules *M'*
16: end if
17: else
18: *Success* = FALSE
19: end if
20: If Success = TRUE
21: do Mutation of $\acute{\omega}_1$ to temp$\acute{\omega}_1$ and $\acute{\omega}_2$ to temp$\acute{\omega}_2$
22: Calculate the *tempPE*$_1$ by *f*(temp$\acute{\omega}_1$) and *tempPE*$_2$ by *f*(temp$\acute{\omega}_2$)
23: If *tempPE*$_1$ better than $PE_{\acute{\omega}1}$ then Replace $\acute{\omega}_1$ with *temp*$\acute{\omega}_1$ and $PE_{\acute{\omega}1}$ with *tempPE*$_1$
24: If *tempPE*$_2$ better than $PE_{\acute{\omega}2}$ then Replace $\acute{\omega}_2$ with *temp*$\acute{\omega}_2$ and $PE_{\acute{\omega}2}$ with *tempPE*$_2$
25: Assign $\acute{\omega}_1$, $PE_{\acute{\omega}1}$, $KE_{\acute{\omega}1}$ to the profile of *M*
26: Assign $\acute{\omega}_2$, $PE_{\acute{\omega}2}$, $KE_{\acute{\omega}2}$ to the profile of *M'*
27: end if
28: Output *M* and *M'*, *Success* and *buffer*

Algorithm 7: Synthesis of MCRO

Input: molecules M_1, M_2
1: Obtain $\acute{\omega}_1$ from ω_1 and $\acute{\omega}_2$ from ω_2
2: Calculate $PE_{\acute{\omega}1}$
3: if $PE_{\omega1} + PE_{\omega2} + KE_{\omega1} + KE_{\omega2} \geq PE_{\acute{\omega}1}$ then
4: *Success* = TRUE
5: $KE_{\acute{\omega}1} = PE_{\omega1} + PE_{\omega2} + KE_{\omega1} + KE_{\omega2} - PE_{\acute{\omega}1}$
6: do Mutation of $\acute{\omega}_1$ to temp$\acute{\omega}_1$
7: Calculate the *tempPE* by f (*temp*$\acute{\omega}_1$)
8: if *tempPE* better than $PE_{\acute{\omega}1}$ then Replace $\acute{\omega}_1$ with *te mp*$\acute{\omega}_1$ and $PE_{\acute{\omega}1}$ with *tempPE*
9: Assign $\acute{\omega}_1$, $PE_{\acute{\omega}1}$ and $KE_{\acute{\omega}1}$ to the profile of M_1
10: else
11: *Success* = FALSE
12: end if
13: Output M_1 and *Success*

The performance in terms of minimization among these four algorithms has been discussed in [16]. The ranking by the best powerful algorithm to the worst algorithm are MCRO, OCRO, HP-CRO2, and RCCRO4, respectively.

Problem functions and evaluation methods

Maximization Operation: As previously mentioned, discovering the most powerful optimal solution of any problem is the main purpose of optimization. In general, the function is called an objective function, cost function (i.e., minimization), or utility function (i.e., maximization). An optimal solution is considered to be the minimum / maximum of the objective function which is known as a global optimal solution. An optimization problem includes minimizing or maximizing a function systematically by selecting input values from a given

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

feasible set. Problem function *f(x)* is a scalar, where a variable *x* represents a particular solution and is usually a vector of *n* components. An optimization problem can be subject to nominated constraints *C*, defined as $C = \{c_1, c_2, . . . , c_m\}$ which limits the feasible region. In the literature, the standard formulation of an optimization problem is largely stated in terms of minimization. Generally, unless both the feasible region and the objective function are convex in a minimization problem, there may be more than one local minimal. A local minimum $x*$ is defined as a point for which the following expression holds [13,17].

$$(x*) \leq (x) \tag{1}$$

The goal of minimization is to find the minimum solution ś ∊ S and (ś) ≤ *f* (*s*), ∀ *s*∊ *S*.

Mathematically, a minimization problem has the following form:

$$\max f(x) \underset{x \epsilon R^n}{subject\ to} = \begin{cases} C_i(x) = 0 i \epsilon E \\ C_i(x) \leq 0 i \epsilon I \end{cases} \tag{2}$$

Where *R*,*E* and *I* symbolize the real number set, the index set for equality constraints, and the index set for inequality constraints, respectively.

With the same concept, the aim of maximizing operation is to generate the maximum solution of *f*(*x*). ś ∊ S and. (ś) ≥ *f* (*s*), ∀ *s*∊ *S*

Mathematically, a maximization problem has the following form:

$$\max f(x) \underset{x \epsilon R^n}{subject\ to} = \begin{cases} C_i(x) = 0 i \epsilon E \\ C_i(x) \leq 0 i \epsilon I \end{cases} \tag{3}$$

Benchmark functions and Parameters: The benchmark functions in this paper are similar to the previous CRO publication [13-16], all experiments are simulated to solve the 23 objective problem functions. Such benchmark functions are classified into three categories as shown in Table 1. Category I is the high-dimensional unimodal functions, category II is the High-Dimensional Multimodal Functions, and category III is the Low-Dimensional Multimodal Functions, More details are contrasted in [13-16].

This research obtains the main parameters provided in CRO framework [13], as shown in Table 2. Moreover, there are several individual parameters for each algorithm that are obtained as its original work, and more description are contained in [14-16].

Evaluation methods: There are three evaluation methods that are used in our experiments to compare the performance of competition algorithms. These three methods are optimal solution quality evaluation, convergence speed, and statistical hypothesis testing.

(1) Optimal solution quality evaluation: In each computing time of optimization algorithm when solving an objective function, this will output a most potential result named optimal solution. Since almost previous publications were focused on solving minimization, the authors considered the result the best global minimum. Different from this paper, we call the optimum result as the best global maximum because we are interested in solving maximization problem.

For algorithms based on CRO in our simulation, each round of iteration contains the local maximum as the best result of round. Accordingly, the current global maximum is replaced by the local maximum when the new local maximum is better than current global maximum. The best global maximum at each computing time is generated when the program meets the stop condition of algorithm. The optimal solution quality is evaluated by comparing the mean of the best global maximum or the optimal solution of each function at all computing times (i.e., in this paper is 25). If the optimal solutions of the competitors are equal, then, the second key for comparison is the standard deviation, the lower standard deviation is winner.

(2) Convergence Speed Evaluation: Beside optimal solution quality, convergence speed is an essential issue that indicates the capability of competition algorithms in terms of speed to reach the global optimal solution. In our experiment, the convergence speed of the algorithm is calculated by counting the number of iterations (FEs) before the algorithm converges into the acceptable solution. Since there is no any acceptable solution that has been published for maximizing, we determine the acceptable solution by averaged the mean of objective function's optimal solution for these four comparison algorithms. The strong algorithms have to generate the result at least not worse than the acceptable solution. The algorithm with fewer FEs is more outstanding than that with greater FEs. Furthermore, we draw a convergence curve for specific functions in a particular run. The ideal curve should begin with the lowest number, then, growing to the highest number, opposite minimization.

(3) Statistical Hypothesis Testing: In order to strengthen our experiments over above two evaluation methods, we verify aptitude of competition algorithms by using statistical hypothesis testing names Friedman test [18]. It is a famous non-parametric statistical testing. Friedman rank is processed by transforming the quality results of contestant algorithms to ranks for each objective function, the average ranks are selected when ranks are equal.

Experimental results and discussion

The proposal for the conducted experiments is to evaluate the performance of competitor algorithm RCCRO4, HP-CRO2, OCRO, and MCRO by running each objective function for 25 times in maximization aspect. We note that, we have changed the condition of algorithms to maximization. Our simulation is built by using C# programming language and the experiments are executed on a computer with Intel Corei7, CPU 3.10 GHz, and Ram 4GB specifications. According to the evaluation method mentioned in section 3.3, the simulation results are demonstrated in the following, proved that the best results are marked in bold.

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

Table 1: The benchmark objective functions

Objective function	No. of Molecules	Lower-Upper bound	Name
Category I			
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	Sphere model
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]^n$	Schwefel's problem 2.22
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100, 100]^n$	Schwefel's problem 1.2
$f_4(x) = max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	$[-100, 100]^n$	Schwefel's problem 2.21
$f_5(x) = \sum_{i=1}^{n-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right)$	30	$[-30, 30]^n$	Generalized Rosenbrock's function
$f_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	$[-100, 100]^n$	Step function Quartic
$f_7(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0,1)$	30	$[-1.28, 1.28]^n$	Function with noise
Category II			
$f_8(x) = -\sum_{i=1}^{n} \left(x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right)$	30	$[-500, 500]^n$	Generalized Schwefel's problem 2.26
$f_9(x) = \sum_{i=1}^{n} \left(x_i^2 - \cos(2\pi x_i) + 10\right)$	30	$[-5.12, 5.12]^n$	Generalized Rastrigin's function
$f_{10}(x) = -20exp(-0.2)\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^n$	Ackley's function
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	Generalized Griewank function
$f_{12}(x) = \frac{\pi}{n}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{29} (y_i - 1)^2 \left[1 + 10sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\}$ $+\sum_{i=1}^{30} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{1}{4}(x_i + 1)$	30	$[-50, 50]^n$	Generalized penalized funtions
$f_{13}(x) = 0.1\left\{sin^2(\pi 3x_1) + \sum_{i=1}^{29} (x_i - 1)^2 \left[1 + sin^2(\pi 3x_{i+1})\right] + (x_n - 1)^2 \left[1 + sin^2(2\pi 30)\right]\right\}$ $+\sum_{i=1}^{30} u(x_i, 5, 100, 4)$	30	$[-50, 50]^n$	

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

Remark: In f_{12} and f_{13} ; $u(x_i,a,k,m)=\begin{cases}(k(x_i-a)^m, & x_i>a\\ 0, & -a\le x_i\le a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$			
Category III			
$f_{14}(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right]^{-1}$	2	[-50, 50]^n	Shekel's Foxholes function
$f_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5, 5]^n	Kowalik's function
$f_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5, 5]^n	Six-hump camel-back function
$f_17(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})cos\ x_1+10$	2	[-5, 10] x [0, 15]	Branin function
$f_{18}(x)=\left[1+(x_1+x_2+1)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ x$[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-32x_1x_2+27x_2^2)]$	2	[-2, 2]^n	Goldstein-Price function
$f_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{4}a_{ij}\left(x_j-p_{ij}\right)^2\right]$	3	[0, 1]^n	Hartman's family
$f_{20}(x)=-\sum_{i=1}^{4}c_i\exp\left[-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right]$	6	[0, 1]^n	
$f_{21}(x)=-\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	[0, 10]^n	Shekel's family
$f_{22}(x)=-\sum_{i=1}^{7}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	[0, 10]^n	
$f_{23}(x)=-\sum_{i=1}^{10}\left[(x-a_i)(x-a_i)^T+c_i\right]^{-1}$	4	[0, 10]^n	

This research obtains the main parameters provided in CRO framework [13] as shown in Table 2. Moreover, there are several individual parameters for each algorithm that are obtained as its original work, and more description are contained in [14, 15, 16].

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

(1) The results of optimal solution quality evaluation are illustrated in Table 3-5. Table 3 represents the optimal solution quality of MCRO, OCRO, HP-CRO2, and RCCRO4 for category I which contains seven high-dimensional unimodal functions (*f*1-*f*7). MCRO conducts the best for *f*2, *f*3, *f*4, *f*5, *f*6 and *f*7, except the *f*1 that generates the best result by RCCRO4.The ranking of optimal solution quality for category I functions from best to worst as follows: MCRO, HP-CRO2, RCCRO4 and OCRO respectively.

Table 4 compares the optimal solution quality for 6 functions in Category II which are high-dimensional multimodal functions. MCRO operates the best for f11, f12 and f13 while HP-CRO2 has the best results for f8, f9, and f10. Therefore, the ranking of optimal solution quality of Category II functions is led by MCRO and HP-CRO2, followed by RCCRO4 and OCRO respectively.

Table 5 compares the results of Category III functions or low-dimensional multimodal functions. MCRO is the most outstanding in this category because it generates the best results for all 10 functions: f14 - f23. MCRO is followed by RCCRO4, while HP-CRO2 is the third rank, and OCRO is the fourth rank.

The overall ranking of optimal solution quality is represented in Table 6, showing that MCRO performs the best in the optimal solution quality evaluation. MCRO is followed successively by HP-CRO2, RCCRO4 and OCRO.

(2) The results of convergence speed evaluation for 4 comparison algorithms are presented in Table 7, all algorithms perform the best convergence of *f*14 when converges the acceptable solution at 1 FE; MCRO, OCRO and HP-CRO2 are leaders of *f*18 when converges the acceptable solution at 1 FE; MCRO and OCRO output the most advantageous convergence speed of *f*20 when converges the acceptable solution at 1 FE; MCRO and HP-CRO2 report the best results for *f*16 .Moreover,

Table 2: MCRO parameters

Parameter \ Category	Category I (f1 - f7)	Category II (f8 – f13)	Category III (f14 – f23)
popsize	10	20	100
InitialKE	1000	10000000	1000
Buffer	0	100000	0
KELossRate	0.1	0.1	0.1
On-wallColl	0.2	0.2	0.2
DecThres	150000	150000	150000
SynThres	10	10	10

Table 3: Results of Optimal solution quality for category I (f1 – f7)

Algorithm	Result of	f1	f2	f3	f4	f5	f6	f7	Average Rank
MCRO	Mean	2.95E+05	**2.87E+02**	**2.85E+05**	**100.00**	**2.23E+09**	**2.96E+05**	**9.52E+02**	
	StDev	3.61E+02	9.00E-01	4.53E+02	0.00	1.97E+07	6.40E+02	1.72E+01	
	Rank	2	**1**	**1**	**1**	**1**	**1**	**1**	**1.14**
OCRO	Mean	1.30E+05	1.79E+02	1.25E+05	9.96E+01	6.65E+08	1.24E+05	3.77E+02	
	StDev	9.88E+03	8.56E+00	1.05E+04	4.48E-01	7.64E+07	7.50E+03	4.61E+01	
	Rank	4	4	4	4	4	4	4	4.00
HP-CRO2	Mean	2.39E+05	2.65E+02	2.37E+05	1.00E+02	1.66E+09	2.41E+05	8.78E+02	
	StDev	2.66E+04	1.48E+01	1.86E+04	1.50E-04	2.50E+08	2.20E+04	1.54E+02	
	Rank	3	2	2	2	2	2	2	2.14
RCCRO4	Mean	**1.28E+07**	1.80E+02	1.28E+05	9.98E+01	6.82E+08	1.26E+05	4.44E+02	
	StDev	**5.80E+05**	1.01E+01	1.30E+04	3.13E-01	9.11E+07	1.00E+04	5.67E+01	
	Rank	**1**	3	3	3	3	3	3	2.71

Table 4: Results of Optimal solution quality for category II (f8 – f13)

Algorithm	Result of	*f*8	*f*9	*f*10	*f*11	*f*12	*f*13	Average Rank
MCRO	Mean	5.79E+03	9.10E+02	2.20E+01	**2.65E+03**	**6.45E+09**	**1.05E+10**	
	StDev	3.23E+02	8.57E+00	1.53E-02	4.18E+00	8.22E+07	1.35E+08	
	Rank	2	2	2	**1**	**1**	**1**	**1.50**
OCRO	Mean	2.50E+03	6.81E+02	2.16E+01	1.16E+03	1.93E+09	3.28E+09	
	StDev	6.02E+02	2.77E+01	8.02E-02	5.68E+01	2.21E+08	2.97E+08	
	Rank	4	4	4	4	3	4	3.83
HP-CRO2	Mean	**9.32E+03**	**1.13E+03**	**2.21E+01**	2.25E+03	5.17E+09	9.21E+09	
	StDev	4.82E+02	3.04E+01	4.07E-02	1.84E+02	9.24E+08	1.23E+09	
	Rank	**1**	**1**	**1**	2	2	2	**1.50**
RCCRO4	Mean	2.89E+03	6.96E+02	2.16E+01	1.22E+03	1.89E+09	3.35E+09	
	StDev	5.27E+02	2.45E+01	5.60E-02	9.89E+01	1.96E+08	4.87E+08	
	Rank	3	3	3	3	4	3	3.17

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

Table 5: Results of Optimal solution quality for category II (f14 – f23)

Algorithm	Result of	f14	f15	f16	f17	f18	f19	f20	f21	f22	f23	Average Rank
MCRO	Mean	**5.00E+02**	**2.69E+15**	**6.46E+03**	**5.05E+02**	**1.02E+06**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**9.99E+01**	**9.99E+01**	
	StDev	**1.36E-08**	1.10E+16	3.92E+01	2.71E-01	4.08E+02	**8.86E-06**	**2.49E-07**	**1.43E-04**	**2.01E-04**	**3.42E-04**	
	NRank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1.00**
OCRO	Mean	5.00E+02	2.65E+06	5.02E+03	3.56E+02	7.00E+05	1.00E+02	1.00E+02	9.99E+01	9.99E+01	9.99E+01	
	StDev	7.71E-05	4.57E+06	6.51E+02	7.54E+01	1.44E+05	1.35E-04	4.38E-05	4.50E-03	6.90E-03	9.96E-03	
	NRank	3	4	4	4	4	3	3	4	4	4	3.70
HP-CRO2	Mean	5.00E+02	1.74E+14	5.81E+03	4.88E+02	9.98E+05	1.00E+02	1.00E+02	9.99E+01	9.99E+01	9.99E+01	
	StDev	1.57E-05	5.01E+14	3.25E+02	1.00E+01	1.43E+04	1.63E-04	4.50E-05	4.31E-03	5.81E-03	8.33E-03	
	NRank	2	2	2	2	2	4	4	3	3	3	3.00
RCCRO4	Mean	5.00E+02	1.12E+09	5.28E+03	3.89E+02	7.24E+05	1.00E+02	1.00E+02	1.00E+02	9.99E+01	9.99E+01	
	StDev	9.76E-05	5.44E+09	6.52E+02	7.03E+01	1.57E+05	6.87E-05	1.75E-05	2.23E-03	5.48E-03	6.89E-03	
	NRank	4	3	3	3	3	2	2	2	2	2	2.29

Table 6: Overall ranking of optimal solution quality

Algorithm	Average Rank				Overall Ranking
	Category I f1-f7	Category II f8-f13	Category III f 14-f23	Three Categories	
MCRO	**1.14**	**1.50**	**1.00**	**1.21**	**1**
OCRO	4.00	3.83	3.70	3.84	4
HP-CRO2	2.14	**1.50**	2.70	2.11	2
RCCRO4	2.71	3.17	2.60	2.82	3

Table 7: Results of Convergence Speed

Function	Acceptable	MCRO		OCRO		HR-CRO2		RCCRO4	
	solution	FEs	Rank	FEs	Rank	Fes	Rank	FEs	Rank
f1	3.35E+06	*	2	*	2	*	2	1	**1**
f2	2.27E+02	163	2	*	3	15	**1**	*	3
f3	1.93E+05	252	2	*	3	37	**1**	*	3
f4	9.98E+01	1	**1**	*	4	22	3	1	**1**
f5	1.30E+09	168	2	*	3	47	**1**	*	3
f6	1.96E+05	287	2	*	3	59	**1**	*	3
f7	6.62E+02	71	2	*	3	21	**1**	*	3
f8	5.12E+03	18382	2	*	3	608	**1**	*	3
f9	8.54E+02	5069	2	*	3	411	**1**	*	3
f10	2.18E+01	138	**1**	*	3	1959	2	*	3
f11	1.82E+03	572	2	*	3	47	**1**	*	3
f12	3.85E+09	371	2	*	3	140	**1**	*	3
f13	6.58E+09	238	**1**	*	3	706	2	*	3
f14	4.999998E+02	1	**1**	1	**1**	1	**1**	1	**1**
f15	7.16E+14	205186	**1**	*	2	*	2	*	2
f16	5.64E+03	1	**1**	*	2	1	**1**	*	2
f17	4.34E+02	115	**1**	248	3	195	2	*	4
f18	8.59E+05	1	**1**	1	**1**	1	**1**	*	2
f19	9.99998E+01	19	**1**	*	3	*	3	2232	2
f20	9.999997E+01	1	**1**	1	**1**	120	2	149	3
f21	9.9952E+01	1	**1**	*	3	*	3	132	2
f22	9.9935E+01	137	2	1	**1**	*	3	*	3
f23	9.99E+01	1	**1**	*	2	*	2	*	2
	Average rank		**1.48**		2.52		1.65		2.52
	rank		**1**		3		2		3

MCRO and RCCRO4 generate the best results of *f4* when converges the acceptable solution at 1 FE. Summary, MCRO generates the fastest convergence of 11 functions: *f4*, *f10*, *f14*, *f15*, *f16*, *f17*, *f18*, *f19*, *f20*, *f21* and *f23*, and 7 of 11 functions converge the acceptable solution at 1 FE. In addition, ranking of other 12 functions for MCRO are second; HP-CRO2 is the best convergence of 12 functions: *f2*, *f3*, *f5*, *f6*, *f7*, *f8*, *f9*, *f11*, *f12*, *f14*, *f16* and *f18* .Although HP-CRO2 performs best for a number of functions there are only 3 of 12 functions that converge the acceptable solution at 1 FE. Moreover, ranking of the rest 11 functions are second for 7 functions and the third for 4 functions. OCRO generates the best results of 4 functions: *f14*, *f18*, *f20* and *f22*; and RCCRO4 generates the best convergence speed of 3 functions: *f1*, *f4* and *f14*. The average convergence speed ranking of comparison algorithms for all functions from fastest to slowest is led by MCRO, the second is HP-CRO2, followed by HP-CRO2 and RCCRO4 which are the same order.

Besides, when evaluating the convergence speed based on the iteration number (FEs), we evaluate the similar results of competition algorithm: MCRO, OCRO, HP-CRO2, and RCCRO4 by drawing a convergence curve of specific functions which select 2 functions from each category: category I (*f3*, *f7*), category II (*f11*, *f12*) and category III (*f16*, *f22*) in a particular run. We note that appreciate curve for maximizing operation should be grown, as opposed to minimization. Figure 1-3 shows that MCRO remains the most outstanding among the four algorithms.

(3) As mentioned above, we provide Freidman test for statistical hypothesis testing .Friedman rank is processed by transforming the results of each function comparison algorithms to ranks. The average ranks are provided when ranks are equal. Table 8 presents the results of Freidman test for optimum solution quality and convergence speed. The results of Friedman rank test in terms of the comparison of optimal solution quality MCRO achieves the best rank, followed by HP-CRO2, RCCRO4 and OCRO respectively. In terms of the competition of convergence speed, MCRO also performs the outstanding and HP-CRO2 is the second best similar to optimal solution quality comparison, while OCRO is the third, then, RCCRO4 is the worst different from optimal solution quality comparison.

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 2(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

In the statistic test, comparing both the optimal solution quality and the convergence speed on the basis of the corresponding Friedman ranks 1.59 and 1.22, the statistic Ff values are 21.76398 and 35.96694, and the p-values are 7.30387E-05 and 7.60987E-08. In addition, the results explicitly display significant differences across the competing algorithms.

The performance measured by these three methods concludes that MCRO absolutely outstanding in order to solve maximization problem.

Conclusion

This paper is concerned with solving optimization problem in maximization aspect. We provided several approaches

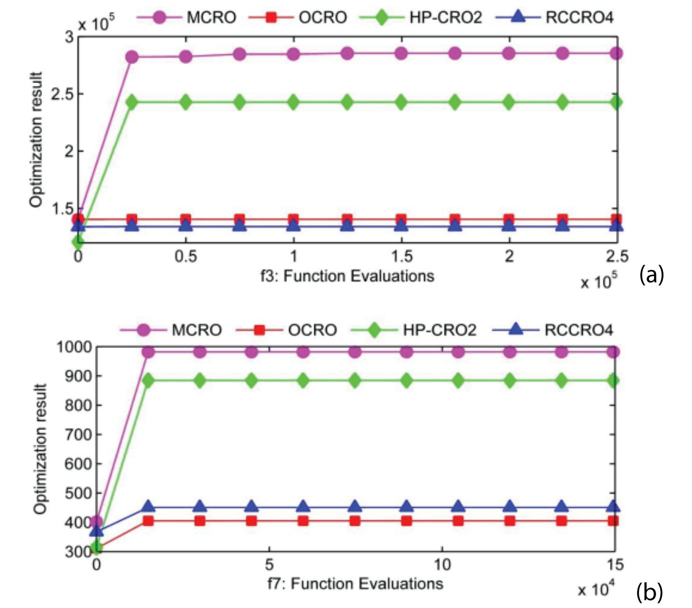

Figure 1: Convergence Curves of function f3 and f7 (category I).

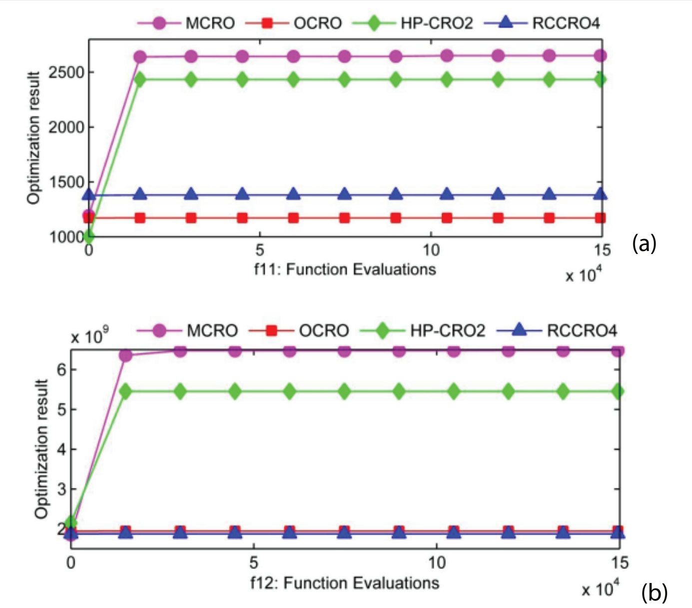

Figure 2: Convergence Curves of function f11 and f12 (category II).

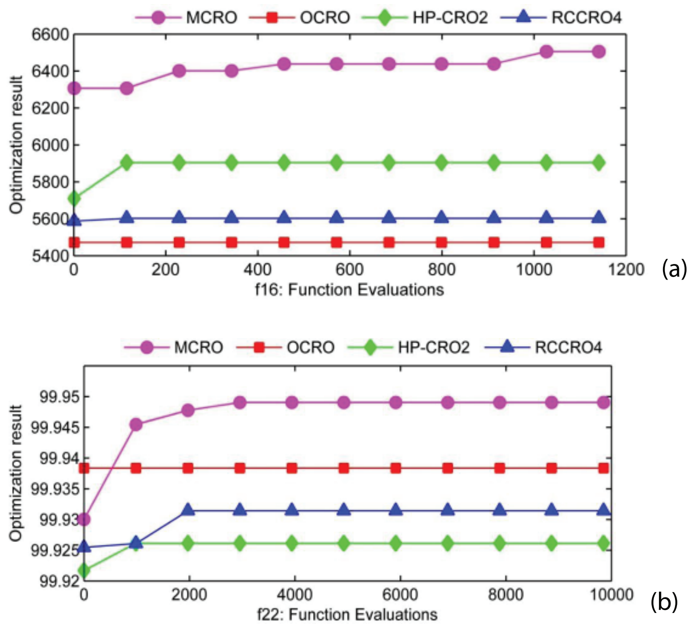

Figure 3: Convergence Curves of function f16 and f22 (category III).

Table 8: Results of Friedman Ranks

Friedman Rank / Function	Optimal Solution quality				Convergence Speed			
	MCRO	OCRO	HP-CRO2	RCCRO4	MCRO	OCRO	HP-CRO2	RCCRO4
*f*1	2	4	3	**1**	3	3	3	**1**
*f*2	**1**	4	2	3	2	3.5	**1**	3.5
*f*3	**1**	4	2	3	2	3.5	**1**	3.5
*f*4	**1**	4	2	3	**1.5**	4	3	**1.5**
*f*5	**1**	4	2	3	2	3.5	**1**	3.5
*f*6	**1**	4	2	3	2	3.5	**1**	3.5
*f*7	**1**	4	2	3	2	3.5	**1**	3.5
*f*8	2	4	**1**	3	2	3.5	**1**	3.5
*f*9	2	4	**1**	3	2	3.5	**1**	3.5
*f*10	2	4	**1**	3	**1**	3.5	2	3.5
*f*11	**1**	4	2	3	2	3.5	**1**	3.5
*f*12	**1**	3	2	4	2	3.5	**1**	3.5
*f*13	**1**	4	2	3	**1**	3.5	2	3.5
*f*14	**1**	3	2	4	**2.5**	**2.5**	**2.5**	**2.5**
*f*15	**1**	4	2	3	**1**	3	3	3
*f*16	**1**	4	2	3	**1.5**	3.5	**1.5**	3.5
*f*17	**1**	4	2	3	**1**	3	2	4
*f*18	**1**	4	2	3	**2**	**2**	**2**	4
*f*19	**1**	3	4	2	**1**	3.5	3.5	2
*f*20	**1**	3	4	2	**1.5**	**1.5**	3	4
*f*21	**1**	4	3	2	**1**	3.5	3.5	2
*f*22	**1**	4	3	2	2	**1**	3.5	3.5
*f*23	**1**	4	3	2	**1**	3	3	3
Average Friedman rank	**1.17**	3.83	2.22	2.78	**1.72**	3.13	1.98	3.20
Rank	**1**	4	2	3	**1**	3	2	4

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 1(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004

based on CRO which had already promised the perspective of minimization in previous publications to the experimental such as RCCRO4, HP-CRO2, OCRO and MCRO. The evaluation results have proved that MCRO which is a hybrid algorithm CRO and polynomial mutation operator is the most excellent in maximizing and minimizing problems among the comparison algorithms for both optimal solution quality and convergence speed. But, there are some variations on ranking of other competitors such as HP-CRO2, which is the third on minimization and the second on maximization for both optimal solution quality and convergence speed. OCRO, which is the second best on minimization, is the third in the part of convergence speed and the worst in optimal solution quality on maximization. Finally, RCCRO4, which the worst on minimization, is the third in the part of optimal solution quality and the worst in convergence speed of maximization.

The results forcefully verified that MCRO is the best optimization approach to be considered as a promising algorithm for solving optimization (minimization or maximization) problems.

References

1. Kita H, Yabumoto Y, Mori N, Nishikawa Y (1996) Multi-objective optimization by means of the thermodynamical genetic algorithm. Lecture Notes in Computer Science 1141: 504–512. Link: https://goo.gl/i2jSYU
2. Wikipedia Mathematical optimization. Link: https://goo.gl/iYWRMx
3. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21: 1087–1092 Link: https://goo.gl/gc15MC
4. Holland J (1992) Adaptation in natural and artificial systems: An introductory analysis with applications to biology. Control and Artificial Intelligence, Cambridge Link: https://goo.gl/tfWTEq
5. Dawkins R (1976) The selfish gene. Oxford University press Link: https://goo.gl/iU0gqm
6. Kennedy J, Eberhart R (1995) Particle swarm optimization. IEEE International Conference on Proceedings Neural Networks 4: 1942-1948 Link: https://goo.gl/5nIs4E
7. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans SystMnCybern 26: 29-41 Link: https://goo.gl/rBSyyw
8. Liong SY, Atiquzzaman M (2004) Optimal design of water distribution network using shuffled complex evolution. J InstEng Singapore 44: 93-107 Link: https://goo.gl/iM0i1h
9. Yang XS (2010) A new metaheuristic bat-inspired algorithm. University of Cambridge, Department of Engineering. Link: https://goo.gl/NAcNAc
10. Geem ZW, Kim JH, Loganathan G V (2001) A new heuristic optimization algorithm: harmony search. Simulation 76: 60–68 Link: https://goo.gl/xJYIaO
11. Lam AY, Li VO (2010) Chemical-reaction-inspired metaheuristic for optimization. IEEE Transactions on Evolutionary Computation 14: 381–399.
12. Lam AY, Li VO (2012) Chemical reaction optimization: A tutorial. Memetic Computing 4: 3–17 Link: https://goo.gl/vptngX
13. Lam AY, Li VO, Yu JJ (2012) Real-coded chemical reaction optimization. Evolutionary Computation, IEEE Transactions 16: 339-535 Link: https://goo.gl/Y790Cx
14. Nguyen TT, Li ZY, Zhang SW, Truong TK (2014) A Hybrid algorithm based on particle swarm and chemical reaction optimization. Expert System with Applications 41: 2134-2143 Link: https://goo.gl/9pq2Aj
15. Li ZY, Li Z, Nguyen TT, Chen SM (2015) Orthogonal chemical reaction optimization algorithm for global numerical optimization problems. Expert System with Applications 42: 3242–3252 Link: https://goo.gl/fxkWDt
16. Ngambusabongsopa R, Li ZY, Eldesouky E (2015) A Hybrid Mutation Chemical Reaction Optimization Algorithm for Global Numerical Optimization. Mathematical Problems in Engineering 2015: 1-18 Link: https://goo.gl/ypoEWw
17. Wang G, Guo L (2013) A novel Hybrid Bat algorithm with Harmony Search for Global Numerical Optimization. Applied Mathematics 2013: 1-21 Link: https://goo.gl/39XmFP
18. Derrac J, Garcia S, Molina D, Herrara F (2011) Swarm and Evolutionary Computation. Swarm and Evolutionary Computation 1: 3-19.d as algorithm 1.

Citation: Ngambusabongsopa R, Havyarimana V, Li Z (2017) A Study of Global Numerical Maximization using Hybrid Chemical Reaction Algorithms. Trends Comput Sci Inf Technol 1(1): 001-011. DOI: http://dx.doi.org/10.17352/tcsit.000004