



Chemical reaction optimization: survey on variants

Md. Rafiqul Islam¹ · C. M. Khaled Saifullah² · Md. Riaz Mahmud¹

Received: 14 April 2018 / Revised: 3 May 2019 / Accepted: 16 May 2019 / Published online: 7 June 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

Chemical Reaction Optimization (CRO) is a recently established population based metaheuristic for optimization problems inspired by the natural behavior of chemical reactions. Optimization is a way of ensuring the usability of resources and related technologies in the best possible way. We experience optimization problems in our daily lives while some problems are so hard that we can, at best, approximate the best solutions with heuristic or metaheuristic methods. This search (CRO) algorithm inherits several features from other metaheuristics like Simulated Annealing and Genetic Algorithm. After its invention, it was successfully applied to various optimization problems that were solved by other metaheuristic algorithms. The robustness of CRO algorithm was proved when the comparisons with other evolutionary algorithms like Particle Swarm Optimization, Genetic Algorithm, Simulated Annealing, Ant Colony Optimization, Tabu Search, Bee Colony Optimization etc. showed the superior results. As a result, the CRO algorithm has been started to use for solving problems in different fields of optimization. In this paper, we have reviewed the CRO based algorithms with respect to some well-known optimization problems. A brief description of variants of CRO algorithm will help the readers to understand the diversified quality of CRO algorithm. For different problems where CRO algorithms were used, the study on parameters and the experimental results are included to show the robustness of CRO algorithm.

Keywords Optimization · Chemical Reaction Optimization · Meta-heuristic · NP-hard · Maximization · Minimization

1 Introduction

In the universe, every natural process has its own pattern or form. But all the processes tend to maintain a common pattern and that uses the least resource to give maximum production. In scientific terminology, we describe this incident as an optimization. The events of optimization can be found from searching food of ants to the ecosystem. Researchers developed many systems for the betterment of mankind where optimization is a key factor. Optimization plays a very crucial role both in industrial and scientific world [1].

The scientists mimic different natural optimized phenomena to use their properties in scientific studies to lessen the time, money and error. These methods lack behind optimal results but can solve NP-hard problems near optimally in polynomial time. The benefit attracts the industries and thus becomes popular to industries and research communities. In general, these methods are termed as nature-inspired computing [2].

The most of optimization related problems are considered as non-deterministic polynomial hard or in short NP-hard problems [3]. Exact algorithms fail to solve most of them in polynomial time. For small instances, exact algorithms like Dynamic programming, Greedy method, Tree search, Graph search etc. may give optimal results but they take exponential time for the large instances or for a large number of instances. Besides Greedy method, Tree and Graph searches may misdirect for large instances and thus lose the optimal characteristics. That is why, the scientists become inspired to use heuristic or metaheuristic algorithms for solving such NP-hard problems. Basically, there is a small difference between heuristic and metaheuristic where both are non-exacts. Heuristic algorithms are designed to solve specific

✉ Md. Rafiqul Islam
dmri1978@cseku.ac.bd

C. M. Khaled Saifullah
khaledkucse@gmail.com

Md. Riaz Mahmud
riazmahmudcse@gmail.com

¹ Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

² Department of Computer Science, University of Saskatchewan, Saskatoon, Canada

problems while metaheuristics are problem independent. Metaheuristic has the advantage over heuristic that it does not get stuck in local optima. Metaheuristic algorithms can be classified as population-based and trajectory-based. For example, Genetic Algorithm, Particle Swarm Optimization are population-based as they use multiple agents or solutions while Simulated annealing is trajectory-based because of using single agent or solution [4].

At present most of the new metaheuristic algorithms are based on a metaphor of some natural or man-made processes as they have been developed by taking inspiration from nature or man-made process [5]. Many algorithms have been designed by capturing inspiration from biological, physical and chemical systems. However, majority of them are based on biological systems [6]. Evolutionary Computing (EC) is special field in research which draws inspiration from the Darwinian natural evolution. EC relates the powerful process of natural evolution to a particular way to solve such kind of problems by trial and error [7]. The algorithms which are involved in EC referred as Evolutionary Algorithms (EA) [8]. An Evolutionary Algorithm is inspired by the mechanism of biological evolution which may include reproduction, mutation, recombination and selection.

The main advantage of these algorithms is that they can return the optimal or the near-optimal results within polynomial time. The industrial world also wants to use approximation solutions since time plays an important role, cost as well as error rate is very much connected to time.

For last 2 decades, different nature-inspired algorithms gained enormous popularity in solving different NP-hard problems. Some of the familiar algorithms are Genetic Algorithm (GA) [9] inspired by Darwin's theory of evolution, Particle Swarm Optimization (PSO) [10] inspired by the behavior of bird flocking or fish schooling, Ant Colony Optimization (ACO) [11] mimics the behavior of ant while searching food, Simulated Annealing (SA) [12] encouraged by the physical process of heating a material and then slowly lowering the temperature to decrease defects.

In 2010 Lam and Li [13] introduced a nature-inspired metaheuristic algorithm named Chemical Reaction Optimization (CRO) that mimics the interaction behavior of molecules participated in a chemical reaction. Using the theory of thermodynamics CRO algorithm showed promising results for solving different optimization problems. The different algorithms based on CRO were proposed for various combinatorial and continuous optimization problems for finding optimal or near optimal solutions. Some problems for which optimal or near optimal solutions may not be gotten by the few renowned approaches where CRO may potentially solve those problems and thus the popularity of application of this algorithm increases. The CRO algorithm was applied to some NP-hard problems such as: Quadratic Assignment [13, 14], Grid Scheduling [15], Cognitive Radio Spectrum

Allocation [16], 0–1 Knapsack [17], Multiple-choice Knapsack [18], Stock Portfolio selection [19], Global Numeric Optimization [20], Longest Common Subsequence [21], Shortest Common Supersequence [22], Printed Circuit Board Drilling [23], Directed Acyclic Graph Scheduling [24], Optimal cluster analysis [25], Capacitated Arc Routing [26], Stock Market Forecasting [27], Static Bike Repositioning [28], Vehicle Routing [29], SCM Transportation Scheduling with TPL [30] etc.

In this paper, we have reviewed the problems which were solved by CRO. During the review process, we noticed that the researchers tried to achieve better results when they made some changes in the basic CRO algorithm. These changes might modify the basic framework of the algorithm to suit different problems. In most cases, the changes are made according to the definitions and properties of the problems. Such variants of tasks are very fine for the problems and give the readers and the researchers a new direction to work on. The energy conversion and transfer are maintained in different entities that makes the CRO unique among other metaheuristics. Other attributes can easily be comprised with the molecule that gives the advantage and flexibility in designing different operators to achieve the best performance. In addition, the advantages of the GA and SA are deployed in CRO [31–33].

Some of the common variants of the algorithm might be hybridized with other algorithms, adding or reducing search operators, adapting CRO algorithm for parallel or multi-objective problems etc. The main motivation for working on these variants is to help readers so that they can find some common approaches used by the researchers to solve a particular problem using CRO algorithm. A general review of CRO algorithm in [31] gives the basic idea about how the search operators work along with the theoretical background. But recently it can be observed that researchers prefer to make changes to the basic structure of the algorithm. These changes help to get the expected results earlier. So, the future research or study on a metaheuristic algorithm should include the variants and their usefulness. Besides parameters play an important role in the optimization algorithm. It is a big challenge to find out suitable values of parameters. Our study on parameters values would give the readers idea about how the parameters were set and used previously. Lastly, our novel approach of getting some statistical results of the experimental values of the reviewed paper will help the readers to catch the effect of variants. The statistical analysis of the results will give a clear picture how the future research will be derived from CRO and other metaheuristic algorithms.

The paper organizes as follows: Section 2 briefly discusses optimization. Section 3 gives the motivation of CRO. Section 4 gives a theoretical description of CRO algorithm along with its components. Section 5 is for energy

conversion. Section 6 illustrates main CRO algorithm framework along with its pseudo-code. Section 7 narrates the variants of CRO algorithm whereas Sect. 8 reviews different applications solved by CRO algorithm. In Sect. 9 a brief study on parameters and experimental results are given, and Sect. 10 is for conclusions.

2 Optimization

Single objective optimization is a study on selecting the best value from a set of feasible values. In our practical life, we see a lot of optimization problems. In general, we can say that, if we have more than one solution to a problem then selecting the best solution is the optimization. So, if S be the set of all possible solutions then we have to find a solution, s that is best among all the solutions, where $s \in S$. For an optimization problem, a solution or search space can be constituted by all possible solutions where each point represents a feasible solution. Now optimization is finding the optimal point (minimum or maximum depending on the nature of the problem) in the solution (search) space. An optimization problem constitutes three components: a set of variables, a set of constraints and a function which is known as objective or fitness function.

If V be the set of variables where $V = \{v_1, v_2, v_3, \dots, v_n\}$ and C be the set of constraints where $C = \{c_1, c_2, c_3, \dots, c_n\}$. The constraints are used to limit the values of variables. Here n denotes problem dimensions whereas, m represents the number of constraints. Now a solution, s is defined as the values of variables V where the variables are restricted by C . Besides S is the search space that defines set of all feasible solutions. Now, if the objective functional value of each solution is f then for a minimization problem, the objective function can be described as in [13] shown below.

$$\begin{aligned} \text{ObjectiveFunction} &= \min_{T \in R^n} f(T) \\ \text{subject to} &\begin{cases} c_i(T) = 0 & i \in EQ \\ c_i(T) \leq 0 & i \in IEQ \end{cases} \end{aligned} \quad (1)$$

Here R denotes a set of real numbers, EQ and IEQ mean index sets of equalities and inequalities respectively. Equation (1) shows the objective function for a minimization problem. The objective function will search the maximum value of $f(T)$ if we consider maximization problem.

On the other hand, the optimization problem with more than one objective functions or target solutions is called multi-objective optimization problem (MOOP). Most of the real-world problems and their solutions are required to be treated as multi-objective optimizations. Different from a single objective problem, a MOOP has more than one target or solution. MOOP does not mean to find one solution

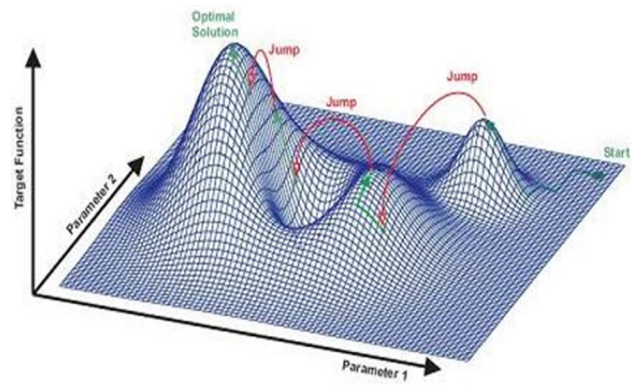


Fig. 1 A solution/search space of a maximization optimization problem

corresponds to each objective function. The air ticket reservation can be represented as a MOOP. There are many varieties of tickets available at different prices. The ticket with higher price has higher facilities for the customer. If the problem is considered simply, it has two objectives: prices and facilities. Instead of finding only one optimal solution, there is a set of optimal solutions (known as Pareto-optimal solutions) by trade-off between prices and facilities. In MOOP, the target is to find these optimal solutions. There is only one optimal solution corresponding to each amount of price for this problem and each customer has to choose one such solution according to their abilities. Though there is a set of optimal solutions, practically each customer wants one optimal solution. These problems may be solved in two ways: one is ideal multi-objective optimization and another one is preference-based multi-objective optimization. To deal with the multi-objective optimization, a set of trade-off optimal solutions are determined and then one solution is selected among all trade-off solutions according to higher level information (such as customer ability in air ticket problem). On the other hand, in preference-based multi-objective optimization, multi-objective functions are converted to a single-objective function using higher level information and then the problem is solved in the same way as single-objective optimization [34].

Figure 1 represents the solution space for an optimization problem that seeks the maximum objective value. Each point on the three-dimensional graph defines the objective function value for each feasible solution depending on parameters 1 and 2. Since the problem is maximization problem so the maximum value of z index is the optimal solution. The optimal solution can be easily identified by the naked eye from the Fig. 1, but finding the optimal solution is a very tough task since it can be at any point in a large solution space. Besides, any approach can be trapped by the local optima. An exact algorithm generally traverses the whole solution space and then gives the optimal solution

which is a very much time and space inefficient approach. Heuristic algorithms start from a position in the space and then calculating the problem-related information. They traverse the space that takes less time but might get stuck in local optima. Moreover, the heuristic algorithm is very much problem dependent. That means the way they follow for a problem will be different for another problem due to the different problem information. For example, a person unknown in a city wants to go to a point “Y” from another point “X”. When he looks for a path, people will help him with path related information like “Go to the train”, “walk in that direction” etc. But when the same person wants to go to point “Z” from “X”, the direction of the path would be different. On the other hand, meta-heuristic algorithms are the problem independent approaches where they start from a random position and jump into a different portion of the solution space. Besides, they have no prior information about the problem and thus there is a higher possibility to find near optimal solution. Meta-heuristic algorithms are designed inspiring from the different optimized phenomena occurred in nature. Almost all meta-heuristic algorithms accept the worse values and thus can skip the trap of local optima. Recent studies show that evolutionary multi-objective optimization basically focuses those portions of the front which satisfy the preference of the decision maker (DM) rather than consider the whole Pareto front. It can be observed that the uniqueness of the DM which is not the case for several decision making situations. A negotiation can help to the several DMs to adjust their preferences by following some negotiation rounds. However, it can be possible to modify several DMs to incorporate the CRO to satisfy the preferences of the DMs [35].

3 Motivation

Chemical Reaction Optimization is a nature-inspired metaheuristic algorithm which has solved many optimization problems by outperforming many other existing optimization algorithms. Though there are many metaheuristics algorithms, the number of optimization problems are many times bigger compare to the number of successful metaheuristics. In most of the cases, metaheuristics solve some definite classes of problems while get poor results for other classes of problems compare to other metaheuristics. In other words, no metaheuristic can solve all types of optimization problems efficiently.

That is why, by realizing the need of a new optimization technique Lam and Li proposed Chemical Reaction Optimization (CRO) in 2010. CRO is based on the phenomena of chemical reaction with optimization. Already it has shown its supremacy over various existing algorithms by solving many popular optimization problems and has the ability

to solve such problems that have been failed to solve successfully by other metaheuristics. The diversification and intensification characteristics help CRO to take the favor of both SA and GA. CRO can be easily modified for different problems to get better optimization results.

4 Chemical Reaction Optimization

Chemical reaction optimization (CRO) is a technique which loosely pairs chemical reactions with optimization [31, 36]. The functions of CRO operate on two laws of thermodynamics. The first law declares that energy cannot be produced or destroyed; energy can be transformed from one kind to another and transferred from one entity to another. In a chemical reaction, there are chemical substances and surroundings around these. Every chemical substance has potential energy (P_E), kinetic energy (K_E) and the energy of surroundings is symbolically representing as the buffer energy.

A chemical reaction can be either endothermic or exothermic. The initial buffer size ($S_B O$) can characterize these two types of chemical reactions. If ($S_B O$) > 0, then the reaction is endothermic and if ($S_B O$) = 0, then it is exothermic. The second law says that the entropy of a system always tends to increase over time. The energy stored in a molecule is referred to as potential energy. When it is converted into kinetic energy, then the system becomes more disordered and entropy increases. The algorithm is a step-wise searching process for minimum energy (optimal solution) [31].

A chemical reaction is accomplished by some sub-reactions and after every sub-reaction, a more stable product is generated. At the same time, a checking is done whether the product is at the optimal point or not. It is a multi-step optimal point searching. This behavior is very similar to many real-world optimization problems. Researchers mimic this natural phenomenon to solve the optimization problem. The two factors that have taken the researchers interests in a chemical reaction are:

- Chemical Reaction always gives a more stable product with minimum energy in it.
- The chemical reaction is a stepwise searching for an optimal solution.

The effectiveness and potential distinction of CRO regarding other metaheuristics can be determined by the following terms [31, 36].

- CRO is a framework which facilitates to exploit the different operators (extra operators) along with its basic four operators to solve different optimization problems. One

can also modify the working procedures of the basic four operators of the CRO according to the requirements.

- CRO gives the opportunity to use variable size population, which allows the system to adapt automatically the problem being solved. When diversification is required decomposition is triggered to produce more molecules due to explore the solution space for finding out the optimal solution. In the case of intensification, the algorithm triggers synthesis for merging molecules. This increases the probability of resultant molecules to be selected for manipulation.
- The basic principle of CRO is that it follows the law of conservation of energy. Energy can be converted and transferred from one form to different entities forms. The total amount of energy held by the molecules and the buffer remains same which makes the CRO unique than other metaheuristics.
- Problem specific heuristics can easily be incorporated into elementary reactions. One can design a molecule for different attributes that suit the problem to be solved as well as give the flexibility to design and manage different operators.
- CRO takes the advantages of both Simulated Annealing and Genetic Algorithm.
- It is easy to implement using object-oriented programming where a molecule is defined as a class and elementary reactions are as methods.
- Since population size does not need to be synchronized, it is easy to adapt CRO to run in parallel.

The objective function of the optimization problem is represented by potential energy in CRO. Besides, some other terminologies of the chemical reaction are represented in CRO algorithm such as molecules, kinetic energy, the number of hits etc. The meanings of these terminologies used in CRO are given below:

- Molecule: A solution in the search space. The structure of the molecule or solution can be an array, vector or matrix.
- Popsiz: Set of all feasible solutions. A 2-D matrix where each row represents the value of a solution.
- Potential Energy: The objective function value related to corresponding molecule.
- Kinetic Energy: Numerical value signifies the amount of tolerance to accept the worst solution.
- NumHit: Number of collisions by a particular molecule.
- KE Loss Rate: Percentage of the upper limit of reduction of KE.
- MoleColl: A parameter to choose whether the chemical reaction is uni-molecular or inter-molecular.
- Initial KE: Initial value of the kinetic energy assigned to each molecule in the initialization stage.

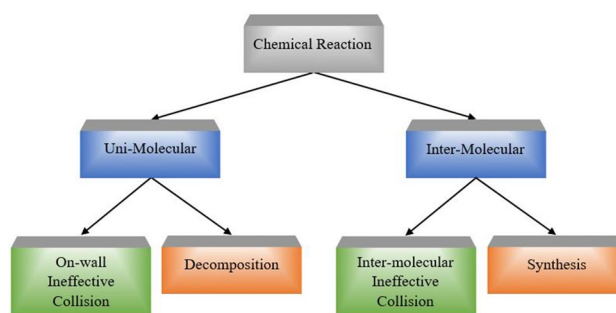


Fig. 2 Chemical collision in CRO

- buffer: Energy of surroundings.
- α, β : Threshold values for the intensification and diversification.
- MinStruct: The molecule structure that has minimum potential.
- MinPE: When a molecule attains its MinStruct, MinPE is its corresponding potential energy value.
- MinHit: It is the number of hits when a molecule has MinStruct.

Collision is the basic element for a chemical reaction. Without any collision, no chemical reaction will occur. Even in CRO one of the main components is a collision. Four basic collisions are considered here. CRO has some features that differentiate the algorithm from other meta-heuristic algorithms. The manipulated agents are molecules which represent the solution space. Only one elementary reaction takes place in each iteration depending on the conditions of the selected molecules. Decompositions are triggered as well as generates more molecules to explore the solution space while diversification is required. In the case of intensification, syntheses are triggered to merge molecules that results in higher probability for selection of the resultant molecules for manipulation. CRO also follows the energy conservation rule. During each of basic four reactions, the total amount of energy (PE and KE) held by the molecules and the buffer remains unchanged. CRO also facilitates the parallel processing [36]. But one of the features of CRO that differentiates the algorithm from other meta-heuristic algorithms is, besides basic four collisions, researchers can add new collision(s) or deduce any of the basic collisions. It can be pointed out that GA also gives us such facilities however CRO gives better results with less execution time. Such results can be observed in [17, 30, 37, 38].

Figure 2 describes the basic chemical collisions for CRO algorithm. The four basic chemical collisions are deployed in CRO inspired from both GA and SA which are primarily divided into two categories depending on the participants in the collision. These two categories are uni-molecular and bi-molecular. Uni-molecular collisions are the collisions

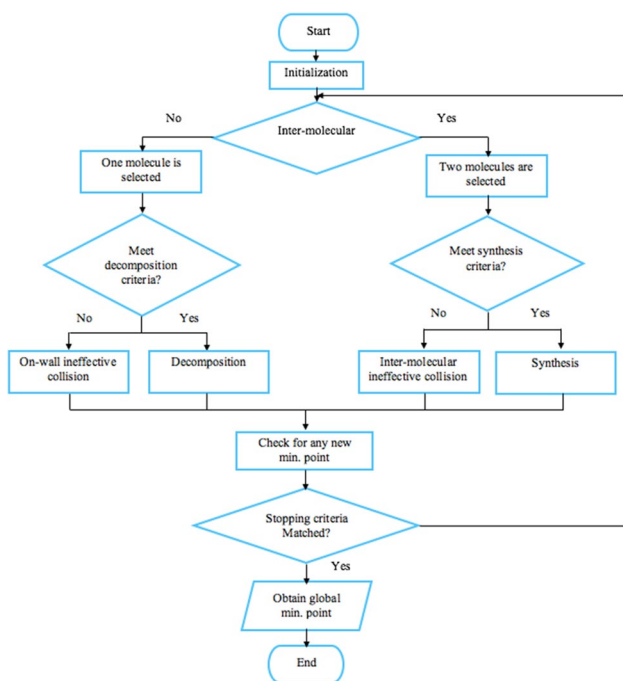


Fig. 3 Flowchart of CRO

where single molecule collides with the surrounding surface and inter-molecular are the collisions where two or more molecules collide with each other. In the case of uni-molecular collision, if the criterion of decomposition is met the decomposition is executed, otherwise on-wall ineffective collision is performed. On the other hand, in the case of an inter-molecular collision if the criterion of synthesis is met then synthesis is done, otherwise, inter-molecular ineffective collision is performed. To observe the difference, flowcharts of CRO and GA are shown in Figs. 3 and 4 respectively.

4.1 On-wall ineffective collision

Single molecule collides with the wall of the surface and returns the same number of product in the on-wall ineffective collision. The molecularity remains in this reaction. This collision is used to implement local search in the search space. Basically, a very small change occurs in the molecular structure of the participant molecule and thus traverses neighborhood space. In this collision, the molecule m slightly changes to m' , i.e., $m \rightarrow m'$. Figure 5 represents the on-wall ineffective collision where molecule, m collides with the surface and produces m' [31].

Let $N(\cdot)$ is neighbourhood search operator where m' is obtained by perturbing m by the operator. Thus, the structure of new molecule (solution) is similar to original one. Now, we have $m' = N(m)$ and $PE_{m'} = f(m')$ where $f(\cdot)$ is an objective function [31].

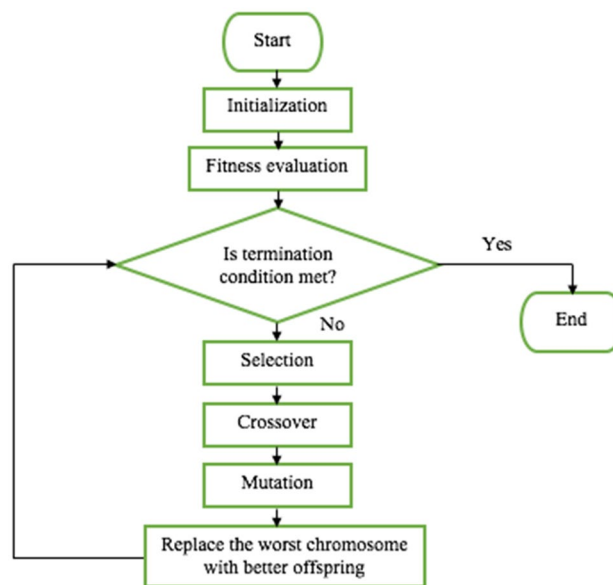


Fig. 4 Flowchart of GA



Fig. 5 On-wall ineffective collision

Let $t \in [KELossRate, 1]$ be a random number, uniformly distributed from $KELossRate$ to 1. Now, we get, $KE_{m'} = (PE_m - PE_{m'} + KE_m) \times t$ and the remaining energy $(PE_m - PE_{m'} + KE_m) \times (1 - t)$ is transformed to *buffer*.

$$PE_m + KE_m \geq PE_{m'} \quad (2)$$

Now, m' is accepted as a new molecule (solution) by replacing m only when Eq. (2) is satisfied. Let $PE_m = 8$, $PE_{m'} = 10$ and $KE_m = 4$, i.e., although the new solution m' is worse than the original one m , the variation is accepted due to $PE_{m'} \leq PE_m + KE_m$ is satisfied. We can say that the PE_m has allowed the acceptance of a worse individual in order to escape the local optima at an early stage. After getting acceptance, a certain portion of KE of the transformed molecule m' is withdrawn to the *buffer*. In this case, when a molecule experiences more of this elementary reaction, it will have more KE transferred to the *buffer*. Hence, the chance of getting a worse solution is lower in a subsequent change [31, 33].

4.2 Decomposition

Decomposition is the uni-molecular collision where a single molecule collides with the wall of the surface and produces

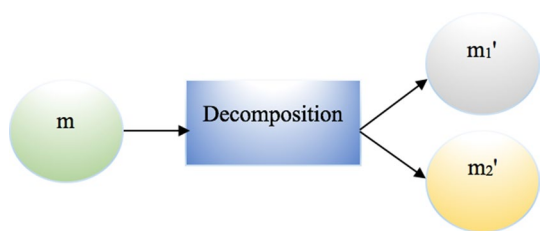


Fig. 6 Decomposition collision

two new molecules. Assume that m produces m'_1 and m'_2 , i.e., $m \rightarrow m'_1 + m'_2$. In Fig. 6, we can see that a molecule m collides with the wall of the container and produces two new molecules m'_1 and m'_2 [31].

Decomposition is launched in order to explore the other parts of the solution space when selected molecule does not experience any new minimum value in terms of number of hits α as well as by triggering On-wall ineffective collision. Massive change occurs in the structure of the molecule and thus, the system can jump into another region of the solution/search space and implements global search. Any mechanism is allowed to produce m'_1 and m'_2 from m . Since more molecules are created by decomposition, the total sum of PE_m and KE_m may not be sufficient [31, 33]. In other words, we may have

$$PE_m + KE_m < PE_{m'_1} + PE_{m'_2} \tag{3}$$

But due to violation of the energy conservation rule this decomposition is rejected. To accept this decomposition, a small portion of energy from *buffer* is drawn randomly. Let σ_1 and σ_2 be two independent and identically distributed numbers uniformly produced in the the range of [0, 1]. We modify the energy conservation condition for decomposition as follows:

$$PE_m + KE_m + \sigma_1 \times \sigma_2 \times buffer \geq PE_{m'_1} + PE_{m'_2} \tag{4}$$

Here, some energy from buffer is transferred to the molecule when it collides with the wall of the surface. The existing molecule m is replaced by the two newly generated ones when Eq. (4) is satisfied. Let $PE_m = 8$, $KE_m = 12$, $PE_{m'_1} = 9$ and $PE_{m'_2} = 10$. In this example, we can see that both new molecules are worse than the original one. However, the new molecules are accepted due to Eq. (4) is satisfied [31, 33].

The remaining energy, $E_{dec} = (PE_m + KE_m + \sigma_1 \times \sigma_2 \times buffer) - (PE_{m'_1} + PE_{m'_2})$ is randomly shared by the $KE_{m'_1}$ and $KE_{m'_2}$ where,

$$KE_{m'_1} = E_{dec} \times \sigma_3 \quad \text{and} \tag{5}$$

$$KE_{m'_2} = E_{dec} \times (1 - \sigma_3) \tag{6}$$

Here δ_3 is a random number generated in the range of [0, 1] and also the energy of buffer is updated by the following equation [31].

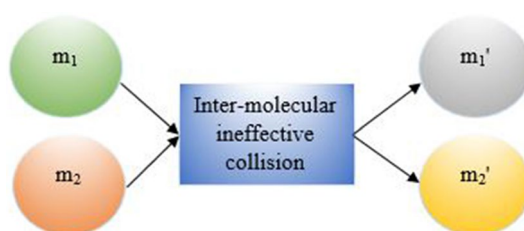


Fig. 7 Inter-molecular ineffective collision

$$buffer' = (1 - \sigma_1 \sigma_2) buffer \tag{7}$$

Molecularity changes in decomposition reaction and thus it proves that CRO algorithm gives the feature of variable population size which is helpful for finding near optimal solution.

4.3 Inter-molecular ineffective collision

The inter-molecular ineffective collision is a reaction where two or more molecules participate in the collision to produce the same number of products. Such that two molecules m_1 and m_2 slightly change to m'_1 and m'_2 respectively, i.e., $m_1 + m_2 \rightarrow m'_1 + m'_2$. Molecularity does not change by this type of collision. Like on-wall ineffective collision, inter-molecular ineffective collision implements local search. A small change in the molecular structure appears in the participants of the collision. We get m'_1 and m'_2 by $m'_1 = N(m_1)$ and $m'_2 = N(m_2)$ [31].

Figure 7 shows the inter-molecular ineffective collision where two molecules m_1 and m_2 collide each other and produce two new solutions m'_1 and m'_2 .

The energy conservation condition can be stated as:

$$PE_{m_1} + PE_{m_2} + KE_{m_1} + KE_{m_2} \geq PE_{m'_1} + PE_{m'_2} \tag{8}$$

As more molecules are participated in this reaction, the total sum of energy of the molecular sub-system is larger than that of the On-wall ineffective collision. Thus, the probability of the molecules to explore their immediate surroundings is increased. In other words, the molecules get higher flexibility to be transformed to more diverse molecular structures. Same operator for On-wall ineffective collision can be used to produce new molecules (solutions). The existing molecules m_1 and m_2 are replaced by the two newly generated ones when Eq. (8) is satisfied. Let $PE_{m_1} = 4$, $PE_{m_2} = 5$, $KE_{m_1} = 3$, $KE_{m_2} = 2$, $PE_{m'_1} = 6$ and $PE_{m'_2} = 7$. Here, it can be seen that both new molecules are worse than the two originals. But the changes are accepted because of satisfying Eq. (8). Then, the *KEs* of the two newly generated ones share the remaining energy $E_{inter} = (PE_{m_1} + PE_{m_2} + KE_{m_1} + KE_{m_2}) - (PE_{m'_1} + PE_{m'_2})$ in the sub-system, i.e.,

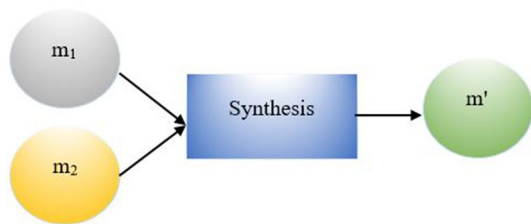


Fig. 8 Synthesis reaction

$$KE_{m'_1} = E_{inter} \times \sigma_4 \quad \text{and} \quad (9)$$

$$KE_{m'_2} = E_{inter} \times (1 - \sigma_4) \quad (10)$$

where σ_4 is a random number generated in the range of [0, 1].

4.4 Synthesis

Two or more molecules collide each other and produce only one molecule in the synthesis reaction. From more than one molecule the algorithm returns only one molecule and suggests a massive change in structure. It is opposite of the decomposition reaction and performs a global search like decomposition. Assume that m_1 and m_2 collide with each other and produce a new one m' , i.e., $m_1 + m_2 \rightarrow m'$. In Fig. 8, two molecules m_1 and m_2 collide with each other and synthesize into a single molecule m' .

The energy conservation can be stated as:

$$PE_{m_1} + PE_{m_2} + KE_{m_1} + KE_{m_2} \geq PE_{m'} \quad (11)$$

Let $PE_{m_1} = 4$, $PE_{m_2} = 5$, $KE_{m_1} = 3$, $KE_{m_2} = 2$ and $PE_{m'} = 12$. Here, we can observe that the new molecule is worse than the two originals. However, the new molecule is accepted because of the role of KEs of the original molecules [31, 33]. If Eq. (11) is satisfied, the resulting $KE_{m'}$ just keeps all the remaining energy, i.e.,

$$KE_{m'} = (PE_{m_1} + PE_{m_2} + KE_{m_1} + KE_{m_2}) - (PE_{m'}) \quad (12)$$

Because of synthesis a greater change has been done on m' with respect to m_1 and m_2 . For this reason, $KE_{m'}$ is usually higher than KE_m . So, the resulting molecule m' has higher ability to explore a new solution region as well as the system can skip the trap of local optima and jump into another portion of the solution or search space.

In general, the principles of a chemical reaction are controlled by the first two laws of thermodynamics [39]. These two laws of thermodynamics have been considered in order to design the algorithm efficiently and effectively. Researchers treat potential energy (PE) and kinetic energy (KE) as the energy of a molecule and central energy buffer as the energy of the surroundings. If the reaction is endothermic then the energy from surroundings is transformed into

energy of molecule and if the reaction is exothermic then energy from a molecule is transformed into the energy of surroundings. The potential energy of a reaction is referred to the objective function of an optimization problem and kinetic energy as a numeric value that quantifies how much a molecule can tolerate the worst value. So the acceptance of a change in a chemical reaction is checked using Eq. (13). Potential energy is considered as the energy which is stored in the molecules against molecular configuration. When the mentioned energy is converted into other pattern, the system becomes more disordered. As an example, when the molecules having higher kinetic energies (basically altered from potential energy) varies in a faster way, the whole system becomes more disordered and thus its entropy rises. Thus, all proceeding systems show a symptom to reach a state of equilibrium, where the level of potential energy reaches to a minimum scale.

5 Energy conservation of CRO

CRO is a population based meta-heuristic algorithm that imitates the behavior of chemical reactions and the algorithm was designed by Lam and Li [31]. Basically, CRO is a technique that loosely couples with chemical reactions [36]. In a chemical reaction, each chemical substance holds potential energy (PE) and kinetic energy (KE) itself and the energy of the surroundings is symbolically representing as the buffer. So, from the first law of thermodynamics we can write,

$$\sum_{i=1}^{PopSize(t)} (PE_i(t) + KE_i(t)) + buffer(t) = Z \quad (13)$$

where $PE_i(t)$ and $KE_i(t)$ denote the potential and kinetic energy of the molecule i at time t respectively, $buffer(t)$ is the energy of the surrounding as well as the energy of the central buffer at time t , and Z is a constant.

However, each molecule has some essential attributes such as the molecular structure, potential energy, kinetic energy, and other parameters. If any molecule holds excessive energy that means the molecule is in the unstable condition. An unstable molecule always tries to get a stable condition with lower energy by undergoing the four basic mentioned reactions. The two ineffective collisions represent a small change in the molecular structure that refers to the effect of intensification as well as local search. On the other hand, a massive change in the molecular structure is noticed in the case of decomposition and synthesis reactions that give the effect of diversification as well as global search. As CRO follows the energy conservation rule so, any of the four reactions will only be triggered when the following equation is satisfied [31].

$$\sum_{i=1}^t (PE_{z_i} + KE_{z_i}) \geq PE_{z'_i} \quad (14)$$

where t denotes the number of reactants, z and z' is the structures of the molecule before and after the reaction. Some problems may conduct negative potential energy (PE) which is obtained from the objective function of the problem. But theoretically, energy can not be a negative value as well as it is not accepted at all. In this case, one can add an offset to the negative objective function value to make each negative PE positive. So, there is no violation of the law of conservation of energy [31, 36].

6 Chemical Reaction Optimization algorithm

Like all other evolutionary algorithms, CRO has three stages: Initialization, iteration, and finalization [31]. In the initialization stage, molecules are generated. Each molecule has following characteristics:

- Molecular Structure.
- Potential Energy.
- Initial Kinetic Energy.
- Initial NumHit.

The initialization stage begins with the initialization of the different parameters of CRO algorithm. The iteration stage consists of some operations between molecules. In the iteration stage, if any of stopping criteria does not meet then the algorithm proceeds to the final stage. The final stage determines a globally optimal solution from the local optimal solutions using its objective function value and terminates the algorithm.

In the iteration stage, a conditional loop starts executing until the stopping criterion is not satisfied. The molecule has lost its kinetic energy according to $KE_{LossRate}$ that is initialized at very first. A threshold value (t) is generated between 0 and 1 randomly to determine whether the reaction will be uni-molecular or inter-molecular. Every molecule of the population is iterated and modified by one of the four operators of CRO algorithm. The value of a variable t decides which operator will be used to modify the solutions. If $t > MoleColl$, a uni-molecular collision is selected, otherwise, inter-molecular collision is selected. In the case of uni-molecular collision, if the criterion of decomposition is met the decomposition is executed, otherwise on-wall ineffective collision is performed. On the other hand, in the case of an inter-molecular collision if the criterion of synthesis is met then synthesis is done, otherwise, inter-molecular ineffective collision is performed to modify the selected molecules. After each

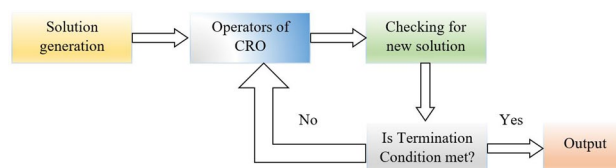


Fig. 9 Steps of CRO algorithm

iteration $NumHit$ of any particular molecule increases if the reaction is unimolecular or $NumHit$ of two or more molecules increases if the reaction is inter-molecular. The two ineffective collisions perform local search (intensification) while synthesis and decomposition perform global search (diversification) [32].

Besides, after each iteration, the resultant molecule is checked with the original one to observe whether the molecule is best or not. The algorithm undergoes through these different reactions until it meets the stopping criteria. In the finalization stage, the global best solution with the objective function value is returned as output. The steps of CRO algorithm are shown in Fig. 9. The pseudo code of CRO algorithm is given in Algorithm 1.

Algorithm 1 Chemical Reaction Optimization

```

1: Input: Problem specific information (Objective Function, constraints and the dimension of the problem)
2: Initialize population with random solutions and set the parameters
3: Compute the fitness value of each molecule as  $PE$ 
4: Let the central energy buffer be  $buffer$  and initialize  $buffer = 0$ 
5: while the stopping criteria not met do
6:   Generate  $t \in [0, 1]$ 
7:   if  $t > MoleColl$  then
8:     Randomly select one molecule  $m_x$ 
9:     if Decomposition criterion met then
10:       $(m_{x'_1}, m_{x'_2}) = Decomposition(m_x)$ 
11:     else
12:       $m_{x'} = OnWallIneffectiveCollision(m_x)$ 
13:     end if
14:   else
15:     Randomly select two molecules  $m_{x_1}$  and  $m_{x_2}$ 
16:     if Synthesis criterion met then
17:       $m_{x'} = Synthesis(m_{x_1}, m_{x_2})$ 
18:     else
19:       $(m_{x'_1}, m_{x'_2}) = InterMolecularIneffectiveCollision(m_{x_1}, m_{x_2})$ 
20:     end if
21:   end if
22:   if the corresponding collision is triggered then
23:     Replace the original molecule(s) with the new one(s) and update the  $buffer$ 
24:   else
25:     Keep the original molecule(s) in the population
26:   end if
27:   Check for any new best solution
28: end while
29: Output: Overall best solution and its objective function value
  
```

Algorithm 2 Generalized Form of CRO Algorithm

```

1: Input: Datasets of the problem, objective function
2: Parameter initialization
3: Solution set generation
4: while the stopping criteria not met do
5:   At first we choose whether the reaction will be
     unimolecular or intermolecular
6:   if the reaction is unimolecular then
7:     We choose a reaction between on-wall ineffective collision
     and decomposition randomly. In general,
     on-wall ineffective collision is used for local search and de-
     composition reaction is used for the global search
8:   end if
9:   if the reaction is intermolecular then
10:    We select a reaction between inter-molecular ineffective col-
    lision and synthesis randomly. In this case, too, ineffective
    collision implements local search whereas synthesis is used
    for the global search
11:   end if
12:   if collision is triggered then
13:    Replace the original solution(s) with the new one(s) in
    the solution set and update the buffer
14:   else
15:    Keep the original solution(s) in the solution set
16:   end if
17: end while
18: Output: Best solution in the set and its
    objective function value

```

In general, if we describe the CRO algorithm using general words as used in other evolutionary algorithms then we can see from the Algorithm 1 that CRO algorithm starts with setting up parameters and creating solution set. Solution set generation technique and parameters are very much problem dependent. Then in the iteration phase, searching of a better solution is done locally or globally which is determined by the values of parameters. In CRO, search operators might include very well-known operators like mutation or crossover used in Genetic Algorithm [9]. After each searching, a new solution is generated and will be included in the solution set if that has better solution quality than the current solution. Iteration phase is regulated by different factors. When the iteration phase includes the best solution in the solution set with the respective objective function value then it will be returned as output. The generalized form of CRO algorithm is shown in Algorithm 2. Some implemented codes are provided here.^{1,2,3} One can follow this for better understanding as well as to get idea for implementation of the CRO algorithm.

¹ <https://sites.google.com/site/slimbechikh/sourcecodes>.

² <http://cro.eee.hku.hk>.

³ <https://github.com/oayld/cro>.

7 Variants of CRO algorithm

Chemical Reaction Optimization (CRO) algorithm has a feature that is flexibility to change the basic framework for any sort of problem. Considering the fact, researchers have designed variants of CRO for different problems. Some of them are discussed below.

7.1 CRO algorithm with Greedy strategy

CRO algorithm with Greedy strategy was used to solve 0–1 knapsack problem [17]. The name signifies that with the CRO algorithm a greedy-based function is used. The authors use basic CRO algorithm and a repair function which is invoked after every reaction. The Repair function has been built based on the greedy strategy and it has two phases: Add phase and Drop phase. Before Add phase, items are sorted on the basis of profit to weight ration.

Now in the Add phase, the sorted items are checked from the beginning and changed the values from zero to one until the feasibility is violated. Then in the Drop phase, a random item is picked and if the value is zero then changed to one if the feasibility is not violated. According to the authors, Add phase improves the fitness of the feasible solution whereas Drop phase confirms the feasibility of a solution. Besides, a Repair operator is used to validate the balance between CPU time and cost, skip the local optima trap. Figure 10 represents the steps of CROG algorithm.

7.2 Artificial chemical reaction optimization

The multiple-choice knapsack problem is solved using Artificial CRO [18]. The concept of ACRO is very similar to CRO wherein CRO we have to consider potential energy as the objective function for both minimization and maximization problem. But in ACRO enthalpy is considered as an objective function if the problem is minimization problem and entropy is considered as the objective function for maximization problem. Now like CRO, ACRO has three stages where, in the initialization stage, reactant/solution is generated defined by *reactantnum* which is referred as *popsiz*e in CRO. Then in iteration phase, one reaction is executed until the termination condition is met. Then in the finalization stage, the best solution is returned as output. Now, ACRO has two types of reactions: monomolecular and bimolecular. Monomolecular reactions are redox1 and decomposition. Here one molecule works as a reactant. Bimolecular reactions are synthesis, redox2 and displacement reactions. ACRO has one more basic reaction than CRO but the basic change in ACRO is that after each reaction if enthalpy is not decreased or entropy is not increased then a reversible

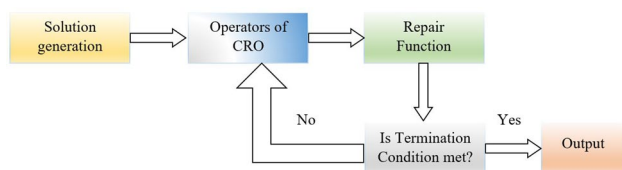


Fig. 10 Steps of CROG algorithm

reaction occurs. This incident differentiates ACRO from CRO. Figure 11 shows the execution process of ACRO algorithm.

7.3 Parallel CRO

One of the distinguishing features of CRO is that the algorithm can be parallelized without much synchronization [31]. Like other evolutionary algorithms, CRO algorithm does not need to maintain the flow if it is run on multiple processors. Each CRO maintains its own population as well as reaction and synchronization can be made after few iterations or computational time. Parallel CRO has been used to solve quadratic assignment problem [14]. Here the best solution for every process is sent to the central node after a certain time. The central node removes all the solutions except the best one among the received solutions. Then the copy of the global best solution is sent to every node where CRO is running and the global best solution replaces any random solution. Thus, the parallelization occurs. It is very clear from the process that no prior synchronization is needed among the processors. After a couple of time, the processor selects its best solution and sends it to the central node and after selecting global best solution one copy is received and a random solution is replaced by the global best solution. Advantages of PCRO over other parallel versions of evolutionary algorithms are varying molecules (population) in each processor and the energy in each buffer is transferred during communication and that effect on the local and global searches of the process [14]. Figure 12 shows the block diagram of PCRO algorithm where each processor works with its own CRO and through communication string, they communicate with the central node.

7.4 Real coded CRO

The concept of Real Coded Chemical Reaction Optimization (RCCRO) algorithm has been given in [40] for the continuous optimization problem. Three modifications have been reported for RCCRO. The first modification is the solution representation. Every solution of the continuous optimization problem is a real number vector where the molecular structure is formed of floating points having specific upper and lower bounds. Besides, continuity of the solution is a

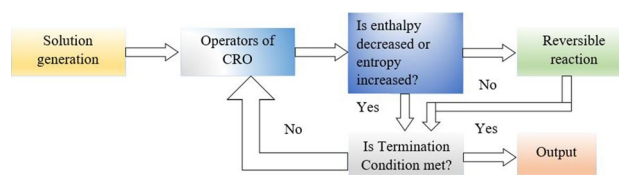


Fig. 11 Steps of ACRO algorithm

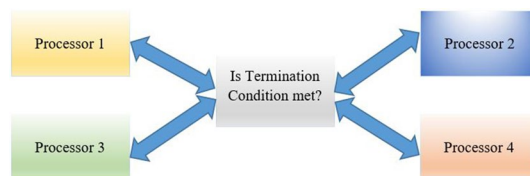


Fig. 12 Block diagram of PCRO algorithm

key factor for this type of problem. The continuity does not effect on global search but for local search, continuity remains a key problem. Therefore, the two local search reaction operators are modified to ensure the continuity of the solution. In general, evolutionary computation adds perturbation to an existing solution to generate new one [41]. Here in RCCRO, the perturbation is used for neighborhood search. There are many probability distributions such as Gaussian, logarithm, exponential etc. where neighborhood search can be incorporated to produce probabilistic perturbation. The third modification is boundary constraints handling. Since the continuous optimization problem is bounded so boundary constraints need to be handled. There are many approaches to handling them whereas most popular are the random and reflecting approaches.

7.5 Mutation CRO

Mutation CRO (MCRO) was proposed in order to solve global numerical optimization problem [20]. Mutation CRO is a hybridization of CRO algorithm and two operators: Turning and Mutation operators. The turning operator is merged to neighborhood search. In MCRO, three reactions (on-wall ineffective collision, decomposition, and intermolecular ineffective collisions) are considered for neighborhood. Turning operator improves the quality of solution and reliability of the algorithm [20]. Mutation operator is incorporated into solution generation and every reaction operator. The authors considered three types of mutation operators: uniform, non-uniform, and polynomial. They claimed that mutation operator increases the probability of finding optimal solution and can skip the trap of local optima. The steps of Mutation CRO shows in Fig. 13.

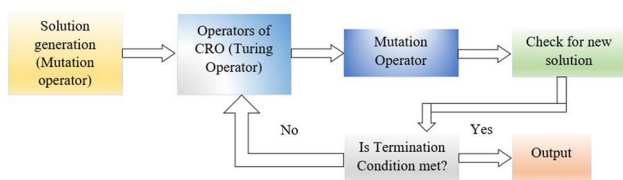


Fig. 13 Steps of MCRO algorithm

7.6 Hybridization of CRO and PSO

Hybridization of CRO and PSO was used to solve continuous optimization problem [42]. Here criteria of both the CRO and PSO algorithms are preserved. Out of four reaction operators of CRO, only two neighborhood operators and a PSO operator (*PSOUpdate*) are used. The algorithm consists of three stages: initialization, iteration, and finalization. In initialization stage, the population which is referred as a molecule in CRO and particle in PSO is generated along with the assignment of parameters of both PSO and CRO algorithms.

Then in the iteration stage, a checking is done with a parameter and a random variable. If the random variable is greater than the parameter then the *PSOUpdate* operator is executed or one of the neighborhood operators of CRO is executed. Iteration stage continues until termination condition is met and then the best solution is returned as output in finalization stage. A pictorial view of steps of the hybrid CRO and PSO is shown in Fig. 14.

7.7 Multi-objective CRO

In real world application, there are many problems which consist of more than one objective interdependent on each other. Solving one objective does not generate the solution. That sort of problem is a multi-objective problem (MOP) [32]. The general form of MOP is described as follows:

$$\begin{aligned}
 \text{Min } f(x) &= [f_1(x), f_2(x), f_3(x), \dots, f_N(x)]^T \\
 g_j(x) &\geq 0 \quad j = 1, 2, \dots, P \\
 h_k(x) &= 0 \quad k = 1, 2, \dots, Q \\
 x_i^L &\leq x_i \leq x_i^U \quad i = 1, 2, \dots, N
 \end{aligned} \tag{15}$$

Here N is the number of objective functions, P is a number of inequality constraints and Q is the number of equality constraints. x_i^L and x_i^U are the lower and upper bounds of variable x . If a solution satisfies $(P + Q)$ constraints then the solution is a feasible one.

Two types of CRO were reported to propose for the multi-objective problem. One was non-dominated sorting CRO

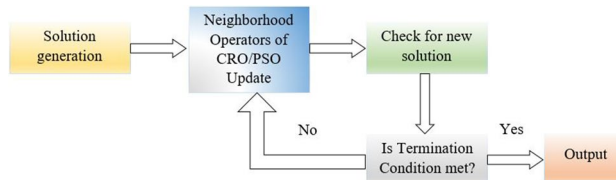


Fig. 14 Steps of hybrid CRO and PSO algorithm

(NCRO) [32] and other was decomposition based CRO (MOCRO/D) [43].

NCRO algorithm works as follows: firstly, offspring population P_q is generated by applying CRO's operators. Next, the population P_q is evaluated. During the evaluation, adaptation in potential energy assignment and offspring generation are done. After the evaluation, the population is combined therefore the new combined population is greater than initial population size but it is not twice as the initial since update procedure of CRO rejects the non-promising population. Now the combined population is sorted according to the non-domination criteria. Therefore, a solution having least potential energy is the best solution remains at the top of the new combined population. In NCRO potential energy is calculated using Pareto rank and crowding distance measure.

During calculation, the solutions having crowding distance zero are removed or replaced by randomly generated different objective vector since the solutions having zero crowding distance are the duplicate one having the same number of objective functions for other solutions. When the combined population is sorted and evaluated then the algorithm chooses the best N number of populations for the further iteration. Figure 15 shows the steps of NCRO algorithm. Decomposition of a multi-objective optimization problem can be done in many ways. The authors in [43] choose Tchebycheff Aggregation approach.

Therefore, the scalar optimization problem [32] can be defined as:

$$\begin{aligned}
 \text{Minimize } g(x|w, z^*) &= \max_{i \in \{1, 2, \dots, m\}} \{w_i |f_i(x) - z_i^*|\} \\
 \text{Subject to } x &\in \Omega
 \end{aligned} \tag{16}$$

Here z^* is the reference point and w is the weighting vector having m points in it. Decomposition based CRO (MOCRO/D) works as decomposing the multi-objective into a single one. Then using CRO each singular objective function is solved. Neighborhood relations among these types of single objective subproblems are defined in accordance with the distances between their weight vectors [43]. Besides Extended CRO based on Decomposition (MOECRO/D) was also proposed for MOP. The main difference between MOCRO/D and MOECRO/D is the different optimization operators.

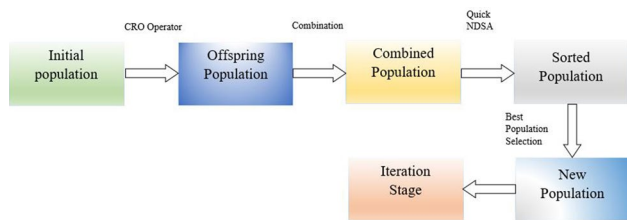


Fig. 15 Steps of NCRO algorithm

8 Application

Chemical Reaction Optimization is a more up-to-date meta-heuristic approach than other evolutionary algorithms like a genetic algorithm or ant colony optimization. But it gains much popularity within a very short period of time. In [31] some applications of CRO algorithm were described. In this paper, we will discuss some new problems those were solved by CRO or variants of CRO algorithm.

8.1 0–1 Knapsack problem

The 0–1 knapsack problem is very well known problem which is a NP-hard. The problem is as follows: given a weight and a profit of each item, the task is to find the optimal combination of items where maximum profit will be gained within a limit of weight. If it is the i th item with profit value p_i and weight capacity is C then the objective function of 0–1 knapsack problem [17] is as below:

$$\begin{aligned} \text{Objective Function} &= \text{Maximize} \sum_{i=1}^n x_i p_i \\ \text{Subject to} \quad &\sum_{i=1}^n x_i p_i \leq C, x_i \in \{0, 1\}, \forall_i \in \{1, 2, \dots, n\} \end{aligned} \quad (17)$$

Dynamic programming approach was proposed to solve this problem but since the problem is NP-hard, so for the large instances, this approach takes exponential time. CRO algorithm with greedy strategy was proposed to solve 0–1 knapsack problem for large instances by Truong et al. [17]. Mutation and crossover operators are used for neighborhood operator whereas half-total exchange operator is used for decomposition and synthesis operator for synthesis reaction. A new Repair function is incorporated with every reaction operator. CROG was compared with Genetic Algorithm, Ant colony optimization, and Quantum-Inspired Evolutionary algorithm. For comparison profit, execution time and the standard deviation are used. The problem instances include 100, 250 and 500 items and the results of profit show that average result of 20 run by CRO has the highest profit than all

other cited algorithms. Besides, the CROG takes the least time and standard deviation for all these instances. Although the outcome of CROG shows promising results, reform function needs some modification for proper implementation. In future, different knapsack problems can be implemented using CROG and detailed study on parameters and reaction operators can enhance the algorithm.

8.2 Multiple choice Knapsack problem

Artificial Chemical Reaction Optimization (ACRO) algorithm was proposed by Truong et al. [18] to solve multiple-choice knapsack problem. The problem is a hard combinatorial problem that states: given the two-dimensional number of items having a particular weight and a profit of each item. The task is to select one item from each row such that maximum profit can be gained having a weight less or equal a particular weight. So, for m classes of items, where each item has a cost c_{ij} and a weight w_{ij} , the objective function [18] is:

$$\begin{aligned} \text{Objective Function} &= \text{Minimize} \sum_{i=1}^m \sum_{j=1}^{n_i} c_{ij} x_{ij} \\ \text{Subject to} \quad &\sum_{i=1}^m \sum_{j=1}^{n_i} w_{ij} x_{ij} \leq W \sum_{j=1}^{n_i} x_{ij} = 1, \\ &\forall_i \in \{1, 2, \dots, m\}, \quad x_j \in \{0, 1\}, \\ &\forall_i \in \{1, 2, \dots, m\}, \quad j \in N_i \end{aligned} \quad (18)$$

An integer string represents the solution and crossover operator is used for neighborhood search and the two-exchange operator is used for decomposition and variants of synthesis operator for synthesis reaction. ACRO algorithm was compared with the Genetic algorithm using three problem instances having class size 10 and 100 and item size 10, 100 and 1000.

Profit, computational time and the standard deviation were used for the comparison. The outcomes of GA and ACRO show that ACRO algorithm has better performance in profit, time and standard deviation for all instances. In the future, parallel version of ACRO can be implemented for better results (Table 1).

8.3 Quadratic assignment problem

Quadratic Assignment problem (QAP) is an abstract principle of an unquestionable renowned Travelling Salesman Problem. The problem states as n facilities are assigned to n locations so that each facility must be located in only one location. Now the task is to minimize the cost which is the summation of

multiplication of distance and the flow generated by any location to all possible locations. The objective function of QAP [13] is:

$$\min \sum_{i,j=1}^n \sum_{p,q=1}^n f_{ij} d_{pq} x_{ip} x_{jq} \quad (19)$$

$$\text{Subject to } \sum_{i=1}^n x_{ij} = 1, i \leq j \leq n; x_{ij} \in \{0, 1\},$$

$$1 \leq i, j \leq n$$

where f_{ij} is the flow between the facilities i and j , d_{pq} is the distance between location p and q . Lam and Victor proposed CRO algorithm for quadratic assignment problem [13] and Parallel CRO was proposed by Xu et al. [14] to solve the same problem. Since the review of CRO has been given in [31], so in this paper, we are giving a review of PCRO. Parallel CRO has same solution structure and reaction operators but uses a parallel algorithm and it performs using 1, 2, 4 and 8 processors at a time. When the number of processors is one then it performs like simple sequential CRO. The outcome of PCRO was compared with sequential CRO and results show that with the involvement of more processors the solution quality has improved along with low computational time consumption. The authors are planning to design and trying more molecular communication patterns. Besides, for a heterogeneous environment, the asynchronous transfer will be implemented and thus, the algorithm will be redesigned. Decisively, PCRO needs to be reviewed provisionally.

8.4 Continuous optimization

Hybridization of CRO and PSO algorithm was proposed by Nguyen to solve 23 benchmark objective functions [42]. Both the searching properties of CRO and PSO apply in the algorithm and that is why, the power of hybrid PSO and CRO increases. Neighborhood operators of CRO and *PSOUpdate* operator of PSO are used here. Since two different algorithms are hybridized so parameters selection was a key factor. For CRO, synthesis threshold is deducted since global searching operators are not needed. Besides, for *PSOUpdate* operator, a new parameter γ is used for checking PSO coefficient. That means after having γ times of local search *PSOUpdate* operator is used. Two versions of hybrid CRO and PSO algorithms were proposed. The versions are categorized according to the boundary constraints. The algorithm terminates when it reaches the maximum function evaluations. Continuous optimization problem was solved by CRO and Real-coded CRO. The outcomes of the hybrid CRO and PSO were compared with three versions

of RCCRO. Objective function value and standard deviation for multiple runs were used for the comparison of HP-CRO1, HP-CRO2, RCCRO1, RCCRO2, and RCCRO4. The experimental results show that HP-CRO outperforms RCCRO for maximum cases. Especially HP-CRO2 has the best results in 18 cases whereas HP-CRO1 ranks 1st in five cases. Besides, the graphical representation of functional evaluation and computational time between HP-CRO and RCCRO show that HP-CRO outperforms RCCRO. In future minimization of some parameters with HP-CRO for more objective functions can be used. Besides, HP-CRO algorithm can be utilized for the multi-objective problem as well.

8.5 Global numerical optimization

Mutation CRO (MCRO) algorithm was applied to solve global numerical optimization problem by Ngambusabongsopa et al. [20]. Basically, 23 objective functions are divided into three categories where category I is termed as high dimensional unimodal function. Category II is preferred to as high-dimensional multi-modal position and category III as low-dimensional multi-modal position. Solutions/molecules are represented as a real vector. The turning operator is used in neighborhood searching and decomposition reaction. Synthesis operator is used in the synthesis reaction. Besides, with each reaction operator and in the population generation mutation operator is used. Authors tried three types of mutation operators: uniform, non-uniform and polynomial, where polynomial mutation operator shows much better performance than other two. Hence for comparing the outcome of MCRO with hybrid CRO and PSO and Real Coded CRO, the authors use polynomial mutation operator. The value of objective function, convergence speed and computational time were compared among the algorithms. The experimental results show that out of 23 instances, only in five cases MCRO algorithm is not better than all other algorithms in case of fitness values whereas in the case of convergence rate MCRO ranked first for 20 instances. Computational times of all algorithms for low-dimensional multi-modal functions are almost same whereas for other two categories OCRO have best results. For future MCRO can be implemented for other numerical optimization problems and practical engineering optimization problems.

8.6 Printing circuit board drilling problem

Printed Circuits Board (PCB) production has a major role in computers and electronic equipments number of holes of various diameters differs a lot [23]. The time to drill the holes depends on the order by which the numerically controlled drilling machine drills the holes. The problem is considered as an instance of the Traveling Salesman Problem (TSP) where time matrix is comparable with distance matrix

Table 1 Application of CRO [31]

Problem	References	CRO type	Years	Solution structure	Neighborhood operator	Decomposition operator	Synthesis operator	Additional function/operator
0–1 Knapsack problem	[17]	CROG	2013	Binary string	Mutation, Crossover	Half-total exchange	Synthesis	Repair function
Multiple choice Knapsack problem	[18]	ACRO	2013	Integer string	Crossover	Two-exchange	Variant of synthesis	Reversible reaction
Quadratic assignment problem	[14]	PCRO	2010	Permutation vector	Two-exchange	Circular shift	Distance preserving	None
Continuous optimization problem	[42]	HP-CRO	2014	Real vector	Gaussian perturbation	Not applied	Not applied	<i>PSOUpdate</i>
Global numerical optimization	[20]	MCRO	2015	Real vector	Turning	Turning	Synthesis	None
Printing circuit board drilling problem	[23]	CRO	2015	Integer vector	Two-exchange, crossover	Half-total exchange	Alternative-select	None
Neural network for effective prediction of stock market	[44]	ACRO	In press	Binary vector	Random masking	Two-exchange	Probabilistic select	Redox1, Redox2
Longest common subsequence	[21]	CRO	2015	Position matrix	Random deletion, crossover	Random deletion	Probabilistic select	Correction function
Shortest common supersequence	[22, 45]	CRO	2016	Integer vector	Two-exchange, crossover	Half-total exchange	Probabilistic select	Reform function
DAG scheduling problem	[24]	CRO	2013	Integer vector	Crossover, circular shift	Circular shift	Probabilistic select	None
Environmentally sustainable NDP	[46]	CRO	2014	Binary string	One-difference	Random-exchange	Synthesis operator	Repair function
Artificial neural network	[47]	CRO	2015	Integer string	Gaussian perturbation	Gaussian perturbation	Probabilistic select	None
Multi-objective problem	[32, 43]	NCRO, MOCRO/D, MOECRO/D	2015	Single objective sub-problem	Polynomial mutation	Polynomial mutation	Multiple molecular collision	None
Set covering problem	[48]	hCRO	2015	Solution vector	Perturbation heuristics	Customized two-exchange	Probabilistic combination scheme	None
Static bike repositioning problem	[28]	Enhanced CRO	2016	Integer string	Mutation, crossover	Half-total exchange	Synthesis	None
Vehicle routing problem	[29]	hCRO	2016	Integer string	Single swap mutation	Order crossover	Partially mapped crossover	None
SCM transportation scheduling	[30]	CRO	2017	Two dimensional matrix	Mutation	Half-total exchange	Synthesis	None

and cities are compared with holes. Therefore, the objective function [13] can be associated with the objective function of TSP and extrapolated in Eq. (20) where $c(i, j)$ defines the time needed to shield holes between i and j .

$$\begin{aligned} \text{Objective Function} &= \min \sum_{i=1}^n \sum_{j=1}^n x(i, j) c(i, j) \\ \text{Subject to } \sum_{i=1}^n x(i, j) &= 1, 1 \leq j \leq n; \sum_{j=1}^n x(i, j) = 1, \\ 1 \leq i \leq n; x(i, j) &\in \{0, 1\} \end{aligned} \quad (20)$$

CRO algorithm was applied to solve the above problem by Eldos et al. [23] where the reactions were renamed as on-wall ineffective collision as deformation, inter-molecular ineffective collision as collaboration and synthesis as a combination. An integer vector is used as the solution representation where the numbering of each hole appears once. The integer vector shows the order of holes to drill. Deformation reaction is done having one or two or three elements swapping to create a new solution. The two-exchange operator is used in decomposition reaction where two solutions are produced from the one reactant. The alternative select operator is applied for conflation and crossover operator is applied for confederation. A random data set having problem size of 22 to 130 holes and PCB442 benchmark instances were used for the experiment. The outcome of the CRO was compared with the optimal value. Authors show that the efficiency of the algorithm increases with the increasing number of iterations. Additionally, decomposition and synthesis reaction rates play a vital part. The authors are thinking to modify the reaction operators to upgrade the performance of CRO.

8.7 Neural network for efficient prediction of stock market indices

The uncertainty of stock market indices rises the difficulties in prediction. The characteristics of stock indices are nonlinear, highly volatile, discontinuous and that is why it needs an artificial neural network to predict. Artificial CRO was proposed by Nayak et al. to learn the artificial neural network [44]. Binary strings are used for solution representation. The probabilistic select operator is applied for synthesis reaction. Crossover operator is used for displacement reaction. Exchange operator is applied for Redox2 reaction and one-difference operator is used for the Redox1 reaction. Besides two-exchange operator is applied in a decomposition reaction. A maximum number of iteration or minimum error signal is used for the termination condition. ACRO algorithm is executed to search optimal weight and bias set for the artificial neural network. The outcomes of ACRO

were compared with MLP, RBFNN and MLR algorithms. Seven benchmark instances of stock indices are used as data-set where prediction is done for 1-day ahead, 1-week ahead and 1 month ahead. Sliding window concept is applied for training and testing the data. Accuracy, POCID and Error Rate are used for the comparison. The experimental results show that ACRO shows better error rate for all instances in all types of predictions. Besides, the gain percentage in the accuracy of ACRO over all other algorithms is between 10 to 50 percentages and POCID values of ACRO are better than all other algorithms for all types of predictions. In future, higher order neural networks might be considered and hybridization of ACRO and another evolutionary algorithm to strengthen the searching capabilities can be achieved.

8.8 Longest common subsequence

Longest Common Subsequence (LCS) is one of the renowned NP-hard combinatorial problems. The problem states that given a set of strings, the task is to find the common subsequence of all strings that is longest in the length. This problem is a maximization problem. A lot of approaches were proposed to solve the problem. CRO algorithm was proposed by Islam et al. [21] to solve LCS problem when the number of strings was two. Position matrix of a valid solution is represented as the molecule. Random deletion and the crossover are used for the neighboring operator. Random deletion operator is used for decomposition reaction and the probabilistic select operator is used for synthesis reaction. With all the basic four reaction operators a new correction function is used to solve the resultant subsequence if the sequence is broke down. The authors of the paper [21] did not solve the LCS problem for multiple string. The work has been extended by Islam et al. [37] and solved the LCS problem for multiple string. The results of the proposed CRO for multiple string were compared with results of some related works.

8.9 Shortest common supersequence

Another renowned NP-hard problem is Shortest Common Supersequence (SCS). The definition of the problem is, given a set of strings formed by a particular alphabet; the task is to find the shortest string in length that is the supersequence of all strings. Supersequence is the string where the sequence of every string of a given set of strings is embedded. Objective function of SCS problem derived from [45] can be written as:

$$\begin{aligned} \text{Objective Function } F(l) &= \min(l_m) \\ \text{Subject to } w_m &\in W, s_m \in S \text{ and} \\ w_m &> s_m \text{ for } m = 0, 1, 2, \dots, k \end{aligned} \quad (21)$$

Here S be the set of input strings where every instance is referred to s_i , W is the set of supersequences and L be the set of the distances of supersequences. Exact, approximate, heuristic and meta-heuristic algorithms were proposed to solve the problem. CRO algorithm was also applied to this problem by Saifullah et al. [22, 45]. In [22] they applied CRO for small instances like the number of strings 500 and length of each string 100. Along with the entire four reaction operators a new Reform function is used after each reaction. The task of Reform function is to check whether constraints of SCS are violated in new supersequence or not. The outcomes of CRO were compared with Dynamic programming and other three approximation algorithms. CRO gives better near-optimal results than all other approximate algorithms in less execution time. In [45] population generation function is improved and Reform function is updated where Reform function contains two phases. In the first phase it checks the validity of new supersequence whereas in the second phase, it repairs the violated supersequences. Thus, CRO algorithm is applied for higher instances. CRO algorithm was compared with ant colony optimization, deposition and reduction, an artificial bee colony and Enhanced Beam Search algorithm. For SCS Enhanced Beam Search was the previous state-of-art. For comparison length of the supersequence, execution time and standard deviation are used. The outcomes show that CRO outperforms all existing algorithms both in length and time. Thus, CRO is proved to be the state-of-art for the SCS problem. In the future, parallel version of CRO can be implemented for SCS problem. Besides, the parameters of CRO can be tuned to get better results.

8.10 DAG scheduling

A connection consisting of a class of duties with precedence constraints is consistently modeled as a Directed Acyclic Graph (DAG) [24]. The major criteria for DAG scheduling is to optimize the *makespan*, which is known as the gap between the time when the first task in the DAG executes and the time when the final task completes execution. Heuristic and meta-heuristic algorithms were preferred for finding a proper solution of DAG scheduling. CRO algorithm was proposed by Xu et al. to solve DAG scheduling [24]. A scheduling solution is considered as the solution or molecule. The molecule is encoded as an integer string where the first half of the solution describes scheduling order and last half describes the resource allotment. Overall schedule length of all tasks named as *makespan* is the potential energy. For neighborhood search crossover operator, circular shift for decomposition and probabilistic select operator for synthesis are used. The outcomes of the CRO were compared with heuristic algorithm HEFT_B and HEFT_T and meta-heuristic algorithm Genetic algorithm (GA). *Makespan*

was used as the comparison criteria. Average *makespan* of all algorithms shows that CRO has the least and best values than all other existing methods. In future authors are planning to implement three things. One is studying parameters for better performance, the second is implementation of parallel CRO for DAG scheduling and lastly implementation of Dynamic Voltage Scaling in CRO for bi-objective optimization.

8.11 Environmentally sustainable network design problem

The bilevel Network Design Problem (NDP) is a two-level problem where the upper level works finding the optimal decision on selecting either link improvements or link additions to an existing road network and lower level accounts for the route choice behavior of network users. Previously the only cost was considered during designing a road network. But in [46] environment sustainable NDP problem is designed where the cost is incorporated with noise and vehicle emission. The bilevel NDP that incorporates with the noise and vehicle emission costs was solved using enhanced CRO by Szeto et al. [46]. The problem states in two levels where the upper-level deals with improving the routes by minimizing total cost (TC) that includes Total System Travel Time Cost (TSTC), Total Emission Cost (TEC) and Total Noise Cost (TNC). Therefore, the objective function of multi-objective upper level [28] is:

$$\min TC = TSTC + TEC + TNC \quad (22)$$

The lower level of the problem captures the behavior of transportation network users. The authors emphasize the different environment issues for example noise and multi-types of emission cost for designing road network. The proposed problem is nonlinear, non-convex and NP-hard. Thus to get global optimum value is difficult. CRO algorithm is designed keeping in mind that the problem is bi-level. A binary string is used for representing the solution. One-difference is used for the neighborhood search whereas decomposition and synthesis operators are used for decomposition and synthesis reaction respectively. Besides a repair function is used if the solution found after the reaction is infeasible. Authors demanded three basic differences between basic CRO and their enhanced CRO algorithm. At first, they applied several conditions for occurring on-wall ineffective collision and decomposition reactions. It ensures that in every reaction a new solution is generated that does not exist in CRO and that affects in computational time. Besides due to using different conditions, one parameter is deducted. Secondly, enhanced CRO can handle multiple level problems whereas the CRO can solve the single level problem. Lastly, a repair function is used for an infeasible solution which is also unique than main CRO. Two benchmark road networks with the different

zones, nodes, links and budgets were used for comparison. The comparison was done in two ways. One is between CRO and GA and another is between inclusions of different costs in the objective function. Objective function value and computation time are the performance comparison criteria. Standard deviation is used as statistical measures for stochastic algorithms like GE and CRO. Comparison results between CRO and GE show that CRO has better objective function value than GA in most cases and in case of computational time CRO outperforms GE in all areas. In future other meta-heuristic algorithms can be compared with CRO for this problem and some other factors can be incorporated with objective during designing networks.

8.12 Artificial neural network

Artificial Neural Network is the family of a model that is inspired by the central nervous system of human to realize the parallel information processing and transformation. Evolutionary algorithm shows better performance than traditional back propagation method [47]. Chemical Reaction Optimization was proposed by Yu et al. for training phase of Artificial Neural Network (ANN) [47]. An integer string with limiting value is used to represent the solution. Besides, Gaussian perturbation method is used for neighborhood search and decomposition reaction whereas probabilistic select operator is used in the synthesis reaction. Two stopping criteria are introduced. One is maximum function evaluation and another is with fitness detection. Three datasets such as Iris classification, Wisconsin breast cancer classification and Pima Indian diabetes datasets are used. The outcomes of CRO algorithm were compared with some well-known meta-heuristic approaches such as particle swarm optimization, evolutionary programming, group search optimizer etc. Error rate was considered as the performance comparison criteria. For the three benchmark datasets, CRO algorithm takes least error rate than all other algorithms. The evolution of the network structure and on the weight adaptation there is no restriction in CROANN and that is why, the algorithm does not suffer from the “structural hill-climbing” problem conformed in the constructive and pruning applications of ANN. The authors have a plan to get a systematic analysis of variance on the parameters and implement a Student’s t-test to show significance of the results. Furthermore, CROANN can be applied to solve several real-world classification problems, and other problems including function approximation, data processing, and robotics.

8.13 Multiobjective problem using NCRO

Multiobjective problems (MOP) are the real world problem where various conflicting and incommensurable objectives

are there. Main challenge of multiobjective problem is that at the same time some objectives are minimizing and some are maximizing with respect to several constraints. Different evolutionary algorithms have gained popularity to solve different MOP. In [32] Chemical Reaction Optimization algorithm was proposed by Bechikh et al. to solve MOP and the authors name the algorithm as non-dominated sorting CRO (NCRO). Along with NCRO, they proposed a new Non-Dominated sorting algorithm (NDSA) named as quick-NDSA (Q-NDSA). For generating the population, the optimized CRO algorithm is applied. Besides, energy management law of CRO is applied to update the population and to determine the solutions remain in the population. To do that each molecule is marked with number of offsprings, their children and collision types so that energy management law of CRO can easily be applied. Next, after applying the rules the children those have violated the rules are removed from the population. Besides, law of conservation of energy is strictly maintained. The results of CRO were assimilated with Non-dominated Sorting Genetic Algorithm (NSGA-II), Indicator-based Evolutionary Algorithm (IBEA), Approximation-Guided EMO (AGE) and Multiobjective Evolutionary Algorithm (MOEA/D). Test problem within EMO community are used as benchmark problem. First seven test problems of DTLZ suite, WFG5 problem and variation of DTLZ-1 and DTLZ-2 that called as scaled-DTLZ1 and scaled-DTLZ2 are used as problem instances. Hypervolume indicator (HV) that means the portion of objective space that is denoted by the Pareto Front approximation returned by the algorithm is used for performance comparison. Besides Wilcoxon Rank sum test in pairwise fashion is used along with HV where if the median of two algorithms are equal then hypothesis H_0 is proved and if not then hypothesis H_1 is proved. Finally computational time calculated for all algorithms and compared with each other. Though HV indicator shows promising results for the NCRO, but the NCRO is time consuming than MOEA/D algorithm. Similarly, no prior performance metric is used for performance measures and parameters tuning has no mathematical explanation. Authors cannot assert that their results can be generalized for real world multi-objective problem which lead to some future investigation.

8.14 Multiobjective problem using MOCRO and MOECRO

Multiobjective problems are efficiently solved by decomposing into singular objective sub-problems by different evolutionary algorithms. Chemical Reaction Optimization (CRO) and Extended Chemical Reaction Optimization (ECRO) algorithms were proposed to solve multi-objective problem (MOP) by Li et al. [43]. The algorithms are MOCRO/D and MOECRO/D respectively. The basic idea of MOCRO/D and

MOECRO/D algorithms is decomposing MOP into single objective sub-problems and solving each subproblem separately. Each molecule of CRO algorithm is considered as a single objective sub-problem. Polynomial mutation is used for inter-molecular ineffective collision that uses three molecules as a reactant and thus enhance more molecular information interactive each other at a certain probability. Multiple molecular collisions are used for synthesis reaction that could increase global search ability. Tchebycheff approach is used for MOCRO/D which is decomposing MOP into number of scalar optimization problem. CRO is used for each sub-problem. Here an MOP is decomposed into N single objective sub-problems by selecting N weight vectors. Besides, the molecules and their neighbors create a group during chemical reaction process. MOCRO/D minimizes all N sub-problems simultaneously in a single run. Depending on the weight vector the neighborhood of a molecule is defined. Each subproblem is optimized using the information of the neighbor molecule. In the first step, the algorithm searches locally by on-wall ineffective collision and inter-molecular ineffective collision. However global search is implemented by decomposition and synthesis. Balance in the energy is maintained using law of conservation of energy.

The difference between MOCRO/D and MOECRO/D is the improvement of global search ability. Here the polynomial mutation is used in decomposition operator for local search. The inter-molecular ineffective collision is used for global search using three molecules. The synthesis reaction is also used for global search using three molecules as a reactant. MOCRO/D and MOECRO/D were compared with NNIA, OMOPSO, NSGA-II, IM-MOEA, IBEA, NCRO algorithms. For performance calculations, inverted generational distance (IGD) was applied. For benchmark instance of ZDT, DTLZ, CEC09 experimental results show that MOECRO/D have better results than all other for complex PF whereas MOCRO/D obtains a better converged and diversified set of nondominated solutions of general PF. The authors have a plan to improve the distribution of the Pareto optimal solution set obtained by MOECRO/D algorithm and implement it for solving dynamic multi-objective optimization problems. Besides, many-objective optimization problems can be investigated further.

8.15 Set covering problem

Set covering problem is the classical NP-hard problem that finds the least cost subsets from a set of elements. The problem can be stated as: If there are n number of elements in a set S then collection of k number of subsets in the set X where $X = \{X_i \subset S, 1 \leq i \leq k\}$ and each subset is associated with a cost in a set of costs, $C = \{c_1, c_2, \dots, c_k\}$, we have to find the least cost prime cover of set S . Prime cover of S

means there will be no redundant subset in X for set S . Yu et al. [48] proposed CRO algorithm for solving set covering problem. Two different encoding schemes are used to represent the solution. One is binary vector and another is solution vector representing subset indices. Authors used the reverse cumulative scheme to assign subset indices. Perturbation heuristics are used for neighborhood searching, customized decomposition operator is used for decomposition reaction and probabilistic combination scheme is used for synthesis reaction. The proposed algorithm was tested on 65 non-unicost SCP instances where the instances were divided into 11 different sets. Parameters are generally set applying trial and error detestable methodology. The outcomes of the heuristic-based CRO (hCRO) algorithm were compared with Lagrangian heuristic, the genetic algorithm, a probabilistic greedy search heuristic an indirect genetic algorithm, a metaheuristic for randomized priority search and a metaheuristic algorithm based on gravity algorithm. Although for non-unicast instances almost all the algorithms have optimal solutions, but for unicast instances CRO algorithm has better near optimal than all other algorithms. The author also proposed some variants of CRO at the end of their paper, where random pic scheme is used in neighborhood operator to generate hCRO/IR algorithm, the remove-repair scheme is used in neighborhood operator to represent hCRO/NR algorithm and crossover operator is used in an intermolecular ineffective collision to make hGA algorithm. Simulation on the variants of hCRO algorithm shows that hCRO algorithm has the best results than all of its variants.

8.16 Static bike repositioning problem

Now-a-days another NP-hard problem called Static Bike Repositioning Problem (static BRP) becomes popular. This problem can be defined as, to lower the total unmet demand at night, the number of bicycles in each station are moved from stations with excess bikes to those with insufficient bikes and the number of bikes each station requires remain unchanged during the repositioning operation is called a static bike repositioning problem [28]. This repositioning problem can be formulated as follows:

$$\text{Min } x = \sum_{j \in m} \phi_j + \delta \sum_{j, k \in m_0, i \neq j} f_{jk} \cdot W_{jk} \quad (23)$$

Here δ is the weight associated with the vehicle's total operational time and ϕ_j is an auxiliary variable associated with each station. This function aims to minimize the weighted sum of two terms: the total number of the unsatisfied customers (or total unmet demand) reflected by $\sum_{j \in m} \phi_j$ and the vehicle's total operational time $\sum_{j, k \in m_0, i \neq j} f_{jk} \cdot W_{jk}$.

Previously, many approaches were proposed to solve this problem. In order to improve the traditional CRO,

some novel operators, new rules, an intensive search and two neighbor-node sets for limiting the search space was introduced to solve routing problems more efficiently [28]. Another subroutine technique has incorporated to determine the loading and unloading quantities at each visited station that makes this method superior to other methods. This technique invokes linear programming or max-flow for solving the sub-problems in each iteration. From the results it is clear that this enhanced version of CRO achieves superiority in solution quality and CPLEX in terms of speed. And the concept of neighbor-node is really a good idea that gives advantage to increase probability of running the intensive search. So this is safe to say that, this enhanced version of CRO has all the capabilities to outperform all existing algorithms.

8.17 Vehicle routing problem

The vehicle routing problem is to find possible optimal routes so that minimum number of vehicles used, cost is minimized and other environment defined constraints are to be satisfied. The problem is defined as an undirected graph, $G = (V, E)$ where $V = \{v_0, v_1, \dots, v_n\}$ is the vertex set and $E = \{v_p, v_q\} : v_p, v_q \in V$ is the edge set [29]. A hybrid algorithm of Chemical Reaction Optimization and Unified Tabu Search was proposed by Thu-Lan Dam et al. for solving vehicle routing problem [29]. Solution representation is a string where at first the route is encoded with “–” and finally customer index is added. Four operators of chemical reaction optimization are implemented but for good solution, in inter-molecular operator Unified Tabu Search is imported. Two datasets were used for the experiment. For benchmark of quality and computing time were compared with UTS, OCGA, PSO, HybPSO and GAB algorithm. Ranking of algorithms was done on using Friedman test. The experimental results show that CROUTS yields ten best results within fourteen instances with less computing time. One of the main contribution of this paper is showing the potential of CRO with other meta-heuristic algorithms.

8.18 SCM transportation scheduling problem with TPL

The transportation scheduling problem of supply chain management (SCM) network is a popular NP-hard combinatorial problem which is being studied in recent years. In this problem, any industry or manufacturing company tries to deliver products, raw materials or services within their distribution network with the help of limited vehicle resources in hand. The target of the vehicle scheduling problem of SCM network is to minimize the overall transportation cost within a wide territory. Recently in 2017, a model was proposed

to solve this problem using CRO by Mahmud et al. [30]. In the beginning, the transportation nodes of the SCM network were divided into three types according to their certain characteristics. In their proposed model, they considered two types of vehicles resources, one is the self-support vehicle of a company, and another is the third party logistics vehicles (TPL). At first, a transportation route combining two special kinds of nodes is created as a solution which is represented by a two-dimensional array. Using this structure, a large number of transportation routes throughout the whole network is generated as the population or solution space. Next, for another single kind of transportation destination node, a matrix is generated randomly where the two types of TPL vehicle resources are selected to serve in some particular nodes. Then, four reaction operators of CRO are redesigned to pick out the best sequences of nodes from both approaches. Finally, the best solution or in this case, the best transportation routes which consist of minimum transportation cost by maintaining all the constraints of the model is found. The experiment results of the proposed CRO in this paper was compared with two Ant Colony Optimization algorithms based model and show that CRO performs significantly better in case of minimizing the transportation cost than the other ACO algorithms.

8.19 RNA structure prediction

The RNA structure prediction (RSP) is a problem that anticipates RNA secondary structure. In this problem we have to compute the correct secondary structure from an given RNA sequence. The objective of the problem is to maximize the number of stem to produce RNA secondary structure and select the most stable structure. The stability of a structure relies on the Gibbs free energy (ΔG) where a structure with minimum energy is accepted. The total energy of different structures of the same sequence is calculated by ΔG . To calculate the free energy of a helix in the RNA secondary structure the individual nearest-neighbour hydrogen bond model is used. To predict RNA structure the following objective function is used [38].

$$R = \min\{\Delta G_j\} \quad (24)$$

where $1 \leq j \leq m; m =$ number of secondary structure for one sequence.

However, in this paper a repair operator is designed to verify and remove the repeated stem from the solution of an RNA sequence with basic four operators that makes the process more time efficient. The experiment results of the proposed CRO were compared with genetic algorithm, simulated annealing algorithm, coincidence algorithm, two-level particle swarm optimization algorithm and changing range bat algorithm and it can be noticed that CRO performs better than other algorithms [38].

9 Study on parameters and experimental results

The performance of every meta-heuristic algorithm mostly depends on the parameter settings of the algorithm. Mislead in the settings might cause a lot of computational time or divergence in the objective function value. So parameter settings and the experimental results have a very deep relation between them. Study on parameters and analysis of experimental results are discussed briefly below.

9.1 Study on parameters

Chemical Reaction Optimization has seven basic parameters. These are *Popsiz*e, *KELossRate*, *MoleColl*, α , β , *Buffer* and *InitialKE*. A number of parameters may vary depending on the designing of CRO for the problem. Even different variants of CRO include new parameters or deduct some basic parameters such as in hybridization of CRO and PSO (HP-CRO), both α and β parameters are deducted. Notations of basic parameters of CRO with their definitions are given in Table 2.

In Table 2 the term *Popsiz*e is referred to the number of molecules/solutions. The number may be changed when the algorithm undergoes decomposition or synthesis reaction. So, *Popsiz*e is the number of solutions at the initial stage and it is not mandatory that the number will be same for the whole process. *KELossRate* is the percentage of loss of kinetic energy after every reaction. Basically, we consider a certain amount of kinetic energy would be transferred from the molecule to the surroundings. The energy of the surroundings is *Buffer*. The transferred amount is defined as *KELossRate*. Another parameter is *MoleColl*. This parameter is used to determine whether the reaction is unimolecular or inter-molecular. Generally, the value of *MoleColl* is between zero and one. Now, in every iteration, a random value between zero and one is compared with *MoleColl*. If the value is more than the *MoleColl* then the reaction is unimolecular otherwise, the reaction is inter-molecular. The threshold α is for decomposition which is used to find out whether the reaction is decomposition or on-wall ineffective collision. Similarly, β is coined as synthesis threshold and used to determine a reaction between synthesis and inter-molecular ineffective collision. *InitialKE* is a value assigned to every molecule as kinetic energy at the initial stage. For the different problems discussed earlier, the parameter values are given in Table 3 so that a reader can get an idea about the values of the basic parameters of CRO.

Variations in values prove that for different problems the values would be different and it is a big challenge for the researchers to identify the best set of parameters for a

Table 2 Parameters of CRO

Notation of parameter	Definition
<i>Popsiz</i> e	Number of initial solutions
<i>KELossRate</i>	Percentage of loss of kinetic energy after each reaction
<i>MoleColl</i>	Used to decide the reaction to be unimolecular or inter-molecular
α	Decomposition threshold
β	Synthesis threshold
<i>Buffer</i>	Initial energy at the surroundings
<i>InitialKE</i>	Initial kinetic energy of each molecule

specific problem. A pattern might be found from the table that in the case of maximization problems like 0–1 knapsack problem or longest common subsequence the value of buffer will be 0 along with decomposition and synthesis thresholds will be as minimum as it can be. But in the case of minimization problems like DAG scheduling problem or Set Covering problem buffer will be as maximum as it can be along with the two thresholds.

9.2 Termination condition

Three types of termination conditions are used with CRO algorithm in the iteration phase. These are:

- Number of function evaluation.
- Computational time.
- Non-decreasing objective function value.
- Obtaining a fixed objective function value.

A number of function evaluations is a trial and error based approach where the number of iterations is determined based on the results obtained for a number of runs on a different number of iterations. For example, for multi-objective optimization problems in [32, 43], multiple-choice knapsack problem [18] all algorithms were compared having a fixed number of function evaluation values. It is a lengthy process but might give a better solution. Fixed computational time is used as stopping criteria too. Here same computational times for all algorithms are used and the performance is compared having better solution quality in fixed consumption of time. Besides non-decreasing objective function value and fixed objective function value defined by a threshold are also used as the termination criteria.

9.3 Experimental result analysis

The outcomes of CRO for different problems were compared with the different evolutionary algorithms like the genetic algorithm, particle swarm intelligence, ant colony

Table 3 Parametric values for different applications

Problem	<i>Popsiz</i> e	<i>KELossRate</i>	<i>MoleColl</i>	α	β	<i>Buffer</i>	<i>InitialKE</i>
0–1 Knapsack problem	20	0.8	0.2	[10–500]	[10–500]	0	100
Quadratic assignment problem	24	0.8	0.2	1300	10^4	0	10^6
Continuous optimization problem	[10–100]	0.1	0.2	–	–	$0, 10^5$	$1000, 10^7$
Global numerical optimization	[10–100]	0.1	0.2	150,000	10	$0, 10^5$	$1000, 10^7$
Printing circuit board drilling problem	150	0.2	0.2	1300	10	0	1000
Longest common subsequence	10	0.1	0.2	10	1000	0	1000
Shortest common supersequence	20	0.6	0.2	[10–100]	[10–100]	0	100
DAG scheduling problem	10	0.2	0.2	500	10	200	1000
Environmentally sustainable NDP	100	0.6	0.6	–	2	0	4
Artificial neural network	20	0.1	0.1	300	500	0	100
Multi-objective problem (MOCRO)	100,300	0.1	0.2	800	15	100	10,000
Multi-objective problem (NCRO)	100	0.6	0.7	15	10	0	10,000
Set covering problem	10	0.1	0.1	10,000	1000	10,000	1000
Static bike repositioning problem	[10–120]	[0.1–0.9]	[0.1–0.9]	[100–3700]	[0–1611]	–	[0–1790]
Vehicle routing problem	20	0.8	0.2	[10,100]	$[10, \frac{X}{20}]$	0	1000

X, total PE of initial pop

optimization, simulated annealing etc. The performance is measured on the basis of some criteria such as convergence speed, execution time, objective function value, the error rate in percentage etc. Besides, CRO being a stochastic algorithm multiple runs of CRO having same experimental setup are used for computing results. Average or the best of all runs is considered as the result of CRO. For reviewing the performance of CRO overall algorithms for the different problems, we consider two methods: average improvement percentage (APR) and average speed rate (ASR). Average improvement percentage is the average rate of better objective function value error rate of CRO in percentage than all other algorithms. Equation (25) shows the average improvement percentage (APR) where n is the number of algorithms compared with CRO and f_{ij} is the objective function value of i th algorithm for j th instance and f_{CROj} is the objective function value of CRO algorithm for j th instance.

$$APR(\%) = \frac{\sum_{i=1}^n \sum_{j=1}^m \frac{(f_{ij} - f_{CROj})}{f_{ij}}}{n} \times 100 \quad (25)$$

Besides, for the computational time, we apply the method average speed rate (ASR). Speed rate method computes the relative computational time between CRO and another algorithm. Mean value of all speed rates is the average speed rate. Equation (26) derives the average speed rate in mathematical notation.

$$ASR = \frac{\sum_{i=1}^n \sum_{j=1}^m \frac{t_{ij}}{t_{CROj}}}{n} \quad (26)$$

Here n is the number of algorithms, m is the number of instances, t_{CROj} is the computational time of CRO algorithm in j th instance and t_{ij} is the time consumed by the i th algorithm for j th instance.

In addition to average improvement rate (APR) and average speed rate (ASR), we calculate the standard deviation (μ) to show the statistical significance of the average results. Best and worst results are also included in the tables along with the average results and standard deviations.

In the case of 0–1 Knapsack problem [17], three problem instances of 100, 250 and 1000 numbers of items were used and outcomes of CRO were compared with Genetic Algorithm, Ant Colony Optimization, and quantum-inspired evolutionary algorithm. The datasets used for multiple-choice knapsack problem [18] had three instances: a number of classes 10 and number of items 10, the number of classes 100 and number of items 100, and the number of classes 1000 and number of items 100. Genetic Algorithm was used for comparison with ACRO. Three problem instances named Will100, Tho150, and Tai256c were used for Quadratic assignment problem [14]. Simulation results of singular CRO were compared with Parallel CRO. For continuous optimization problem, 23 benchmark functions were used which were classified into three categories [42]. The simulation results of HP-CRO were compared and ranked with real coded CRO. Twenty six benchmark objective functions were used as problem instances for global numerical optimization problem [20]. Then the objective function value of ACRO and canonical CRO were compared with HP-CRO, RCRO, and OCRO. For building model for effective prediction of the stock market daily closing prices of the different stock markets

like BSE, DJIA, NASDAQ, TAIEX, FTSE, S&P 500 and LSE were used [44]. Two third data were used for training and one third for testing. Sliding window technique was used for 1 day, 1 week and 1-month predictions. Error generation for prediction of ACRNN was compared with MLP, RBFNN and MLR algorithms. For LCS problem [21] two strings were used as input having lengths 128, 270 and 360. The lengths of LCS obtained by CRO were compared with DP and FLCS algorithms. In the case of SCS problem [45], two classes of datasets were used one of them was DNA and another was Protein. In both classes, a random and real datasets were used. For DNA, random dataset instances having the number of strings 5 to 5000 and lengths 10 to 1000 were used. In real dataset 6 DNA benchmark and 5 protein benchmark instances were used. The outcome of CRO was compared with ACO, beam search, some heuristic algorithm like deposition and reduction, enhanced beam search. Two road networks of USA were used for Environmental Sustainable Network Design Problem [46]. One is Sioux Fall network which is small and other is Anaheim Network which is relatively long. Besides for the simulation results with Genetic Algorithm three networks named SC, SD and AC were used. Three datasets named as Iris, Wisconsin Breast Cancer, and Pima Indian Diabetes datasets were used for neural network [47]. For Iris data set 75 training sample, 37 validation sample and 38 testing sample were used. In the case of Wisconsin Breast Cancer dataset 349 training sample, 175 validation sample and 175 testing sample were used. Lastly, in the case of Pima Indian Diabetes dataset 384 training samples, 192 validation samples, and 192 testing samples were used. The outcomes of CROANN for three datasets were compared with SGAANN, EPANN, ESANN, PSOANN, GSOANN. In the case of multiple objective optimization problem [32, 43], a dataset named CEC09 having four multiple objective functions were used as problem instance.

Table 4 shows the average improvement rate (APR) of CRO with the compared algorithms used to solve different problems. Here 1st column represents the problem name, 2nd column is for the algorithms those are compared with CRO, 3rd and 4th columns show the best and the worst improvement rates of CRO with all other comparing algorithms respectively, 5th column represents the average improvement rate (APR), and 6th column describes standard deviation. From the table we can see that Chemical Reaction Optimization has outperformed different evolutionary algorithms by over 0.1%. That means CRO algorithm has proven better than all other compared algorithms. Values of standard deviation demonstrate that improvement rate of CRO over all other algorithms is very much close to each other.

Table 5 represents the average speed rate (ASR) of CRO with all other compared algorithms. Similar to the Table 4, column 1 shows the problems, column 2 represents the compared algorithms, columns 3 and 4 show best and the worst values in case of computational time, column 5 shows the average speed rate (ASR) and column 6 is for the standard deviation values. Herein some problems like Multi-objective problem solved by MOCRO [43] and efficient stock prediction using artificial neural network [44], the authors did not consider computational time for comparison. So apart from these problems, average speed rates of other problems are shown in Table 5.

The results in Table 5 show that CRO has good efficiency in execution of the algorithms to solve the different kinds of problems. The CRO has average speed rate over five it means, CRO is five times faster (on average) than other algorithms. Besides from standard deviation we can see that speed rate of CRO over each algorithm is very much close to its average value. Having both local and global search capabilities, CRO algorithm can converge very fast and thus reduce the computational time. Such time efficiency makes CRO popular to the researchers those who want to solve different NP-hard and NP-completer problems.

10 Conclusions

This paper reviews one of the recent and well-known metaheuristics approaches named Chemical Reaction Optimization (CRO) algorithm along with its variants. The algorithm mimics one of the very popular nature inspired optimization incidents. The diversity and flexibility of design process of the algorithm make CRO very much interesting to the researchers. The energy conservation conducts the acceptance of new solution(s) as well as facilitates the scope of search. Its variable population size enables the algorithm to adapt to the problem with a fair combination of intensification and diversification. Variants of the algorithm prove the diversity in designing the algorithm for the problems. The current study shows the power of CRO to search solutions locally and globally in the search or solution space. Thus, CRO is much robust and efficient algorithm for solving the hard combinatorial problems. The dimensions of the problems solved by CRO algorithm are raising and hence the popularity of application of this algorithm is rising rapidly. CRO is now using in the filed of bioinformatics, data mining, computer visions, civil engineering etc. CRO has been developed for solving the multiobjective problems also. For learning artificial neural network to predict complex data like a stock portfolio or DNA sequencing, RNA structure prediction, CRO algorithm proves its robustness. A parallel

Table 4 Average improvement rate of CRO in %

Problem	Algorithms compared	Best improvement rate	Worst improvement rate	Average improvement rate (APR) %	SD (μ)
0–1 Knapsack problem	GA, ACO, QEA	1.1	0.02	0.36	0.004
Multiple choice Knapsack problem	GA	0.9	0.01	0.35	0.0007
Quadratic assignment problem	CRO	0.62	0.007	0.14	0.008
Continuous optimization problem	RCCRO1, RCCRO2, RCCRO4	1.2	0.1	0.43	0.01
Global numerical optimization	OCRO, HP-CRO2, RCCRO4	1.7	0.11	0.52	0.006
Neural network for effective prediction of stock market	MLP, RBFNN, MLR	2.1	0.1	0.68	0.007
Longest common subsequence	DP, FLCS	0.000003	0.0000005	0.000001	3×10^{-7}
Shortest common supersequence	ACO, ABC, DR, IBS_SCS	2.1	0.14	0.62	0.01
Environmentally sustainable NDP	GA	0.64	0.008	0.15	0.002
Artificial neural network	SGAANN, EPANN, ESANN, PSOANN, GSOANN	0.24	0.004	0.11	0.0004
Multi-objective problem (NCRO)	NSGA-II, IBEA, MOEA/D, AGE	1.2	0.07	0.34	0.0009
Multi-objective problem (MOCRO)	MOECRO, NCRO, MOEA/D-DE, NSGA-II, NNIA, MOEA/D-DRA, IBEA, SMPSO, IM-MOEA	1.3	0.1	0.37	0.0007

Table 5 Average speed rate of CRO

Problem	Algorithms compared	Best speed rate	Worst speed rate	Average speed rate (ASR) %	SD (μ)
0–1 Knapsack problem	GA, ACO, QEA	0.07	0.0006	0.03	0.0005
Multiple choice Knapsack problem	GA	3.1	1.1	1.7	0.04
Quadratic assignment problem	CRO	3.4	1.4	2.5	0.2
Continuous optimization problem	RCCRO4	3.1	1.1	2.3	0.09
Global numerical optimization	OCRO, HP-CRO2, RCCRO4	0.009	0.0008	0.006	0.0004
Longest common subsequence	DP, FLCS	4.1	0.2	2.3	0.3
Shortest common supersequence	ACO, ABC, DR, IBS_SCS	6.2	1.1	4.2	0.1
Environmentally sustainable NDP	GA	0.1	0.001	0.07	0.0005
Multi-objective problem (NCRO)	NSGA-II, IBEA, MOEA/D, AGE	4.1	1.4	3.1	0.3

version of CRO shows much effectiveness to solve problems having low computation time consumption as well as without prior synchronization. The study on parameters shows that for various types of problems the parameters settings were very much different from each other although we tried to find a pattern that might help the researchers to find a better set of parameters. The experimental analyses depict that CRO algorithm has very good ability to converge results having better average improvement rate in percentage in almost all the problems we reviewed. Besides, average speed rate of applicable cases shows that the CRO algorithm is much faster than other evolutionary algorithms including Genetic Algorithm, Ant Colony Optimization, Particle Swarm Optimization, Bee Colony Optimization etc. More study on CRO might help

researchers to fit the algorithm in other sides of computer science as well as different engineering sectors.

Compliance with ethical standards

Conflicts of interest The authors have no conflict of interest.

References

1. Blum C (2005) Ant colony optimization: introduction and recent trends. *Phys Life Rev* 2(4):353–373
2. Shadbolt N (2004) Nature-inspired computing. *IEEE Intell Syst* 19(1):2–3

3. Garey MR (1979) Computers and intractability: a guide to the theory of np-completeness. *Revista Da Escola De Enfermagem Da USP* 44(2):340
4. Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver Press, Osaka
5. Sörensen K (2015) Metaheuristics—the metaphor exposed. *Int Trans Oper Res* 22(1):3–18
6. Fister Jr I, Yang XS, Fister I, Brest J, Fister D (2013) A brief review of nature-inspired algorithms for optimization. arXiv preprint [arXiv:1307.4186](https://arxiv.org/abs/1307.4186)
7. Eiben AE, Smith JE et al (2003) Introduction to evolutionary computing, vol 53. Springer, Berlin
8. Al-Salami NM (2009) Evolutionary algorithm definition. *Am J Eng Appl Sci* 2(4):789–795
9. Holland J, Goldberg D (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston
10. Kennedy J, Eberhart RC, Shi Y (2001) Swarm intelligence. Elsevier, Amsterdam. <https://doi.org/10.1016/B978-1-55860-595-4.X5000-1>
11. Dorigo M, Birattari M, Stützle T (2006) Ant colony optimization—artificial ants as a computational intelligence technique. *IEEE Comput Intell Mag* 1(4):28–39
12. Brooks SP, Morgan BJ (1995) Optimization using simulated annealing. *The Statistician* 44:241–257
13. Lam AY, Li VO (2010) Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans Evol Comput* 14(3):381–399
14. Xu J, Lam AY, Li VO (2010) Parallel chemical reaction optimization for the quadratic assignment problem. In: World congress in computer science, computer engineering, and applied computing, Worldcomp 2010
15. Xu J, Lam AY, Li VO (2011) Chemical reaction optimization for task scheduling in grid computing. *IEEE Trans Parallel Distrib Syst* 22(10):1624–1631
16. Lam AY, Li VO (2010) Chemical reaction optimization for cognitive radio spectrum allocation. In: Global telecommunications conference (GLOBECOM 2010), 2010 IEEE. IEEE, pp 1–5
17. Truong TK, Li K, Xu Y (2013) Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. *Appl Soft Comput* 13(4):1774–1780
18. Truong TK, Li K, Xu Y, Ouyang A, Tang X (2013) An artificial chemical reaction optimization algorithm for multiple-choice knapsack problem. In: Proceedings on the international conference on artificial intelligence (ICAI). The steering committee of the world congress in computer science, computer engineering and applied computing (WorldComp), p 1
19. Xu J, Lam AY, Li VO (2011) Stock portfolio selection using chemical reaction optimization. In: Proceedings of international conference on operations research and financial engineering (ICORFE 2011), pp 458–463
20. Ngambusabongsopa R, Li Z, Eldesouky E (2015) A hybrid mutation chemical reaction optimization algorithm for global numerical optimization. *Math Probl Eng* 2015, Article ID 375902. <https://doi.org/10.1155/2015/375902>
21. Islam MR, Asha ZT, Ahmed R (2015) Longest common subsequence using chemical reaction optimization. In: 2015 2nd international conference on electrical information and communication technology (EICT). IEEE, pp 29–33
22. Saifullah CK, Islam MR (2016) Solving shortest common supersequence problem using chemical reaction optimization. In: 2016 5th international conference on informatics, electronics and vision (ICIEV). IEEE, pp 50–55
23. Eldos T, Kanan A, Nazih W, Khatatbih A (2015) Adapting the chemical reaction optimization algorithm to the printed circuit board drilling problem. In: International conference on computer, computational and mathematical sciences. Zürich, Switzerland
24. Xu Y, Li K, He L, Truong TK (2013) A dag scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization. *J Parallel Distrib Comput* 73(9):1306–1322
25. Nayak J, Naik B, Behera HS, Abraham A (2017) Hybrid chemical reaction based metaheuristic with fuzzy c-means algorithm for optimal cluster analysis. *Expert Syst Appl* 79:282–295
26. Bensedira B, Layeb A, Bouzoubia S, Habbas Z (2016) CRO-CARP: a chemical reaction optimization for capacitated arc routing problem. In: 2016 8th international conference on modelling, identification and control (ICMIC). IEEE, pp 757–762
27. Sahu SR, Behera HS (2016) A hybrid CRO-based FLANN for financial stock market forecasting. *Int J Data Anal Tech Strateg* 8(3):261–279
28. Szeto W, Liu Y, Ho SC (2016) Chemical reaction optimization for solving a static bike repositioning problem. *Transp Res Part D Transp Environ* 47:104–135
29. Dam TL, Li K, Fournier-Viger P (2017) Chemical reaction optimization with unified tabu search for the vehicle routing problem. *Soft Comput* 21(21):6421–6433
30. Mahmud MR, Pritom RM, Islam MR (2017) Optimization of collaborative transportation scheduling in supply chain management with TPL using chemical reaction optimization. In: 2017 20th international conference of computer and information technology (ICCIT). IEEE, pp 1–6
31. Lam AY, Li VO (2012) Chemical reaction optimization: a tutorial. *Memet Comput* 4(1):3–17
32. Bechikh S, Chaabani A, Said LB (2015) An efficient chemical reaction optimization algorithm for multiobjective optimization. *IEEE Trans Cybern* 45(10):2051–2064
33. Chaabani A, Bechikh S, Said LB (2018) A new co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization. *Appl Intell* 48(9):2847–2872
34. Deb K (2014) Multi-objective optimization. In: Search methodologies. Springer, pp 403–449
35. Bechikh S, Said LB, Ghédira K (2011) Negotiating decision makers' reference points for group preference-based evolutionary multi-objective optimization. In: 2011 11th international conference on hybrid intelligent systems (HIS). IEEE, pp 377–382
36. Lam AY, Li VO, Xu J (2013) On the convergence of chemical reaction optimization for combinatorial optimization. *IEEE Trans Evol Comput* 17(5):605–620
37. Islam MR, Saifullah CK, Asha ZT, Ahamed R (2018) Chemical reaction optimization for solving longest common subsequence problem for multiple string. *Soft Comput*. <https://doi.org/10.1007/s00500-018-3200-3>
38. Kabir R, Islam R (2018) Chemical reaction optimization for RNA structure prediction. *Appl Intell* 49(2):352–375
39. Guggenheim EA (1985) Thermodynamics—an advanced treatment for chemists and physicists. Amsterdam, North-Holland, p 414
40. Yu JJQ, Lam AYS, Li VOK (2012) Real-coded chemical reaction optimization with different perturbation functions. In: 2012 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8
41. Dong H, He J, Huang H, Hou W (2007) Evolutionary programming using a mixed mutation strategy. *Inf Sci* 177(1):312–327
42. Nguyen TT, Li Z, Zhang S, Truong TK (2014) A hybrid algorithm based on particle swarm and chemical reaction optimization. *Expert Syst Appl* 41(5):2134–2143
43. Li H, Wang L, Hei X (2016) Decomposition-based chemical reaction optimization (CRO) and an extended CRO algorithms for multiobjective optimization. *J Comput Sci* 17:174–204
44. Nayak S, Misra B, Behera H (2017) Artificial chemical reaction optimization of neural networks for efficient prediction of stock market indices. *Ain Shams Eng J* 8:371–390

45. Saifullah CK, Islam MR (2016) Chemical reaction optimization for solving shortest common supersequence problem. *Comput Biol Chem* 64:82–93
46. Szeto WY, Wang Y, Wong SC (2014) The chemical reaction optimization approach to solving the environmentally sustainable network design problem. *Comput-Aided Civ Infrastruct Eng* 29(2):140–158
47. Yu JJQ, Lam AYS, Li VOK (2011) Evolutionary artificial neural network based on chemical reaction optimization. In: 2011 IEEE congress on evolutionary computation (CEC). IEEE, pp 2083–2090
48. Yu JJQ, Lam AYS, Li VOK (2014) Chemical reaction optimization for the set covering problem. In: 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 512–519

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.