On the Convergence of Chemical Reaction Optimization for Combinatorial Optimization

Albert Y.S. Lam, *Member, IEEE,* Victor O.K. Li, *Fellow, IEEE,* and Jin Xu

***Abstract*—A novel general-purpose optimization method, chemical reaction optimization (CRO), is a population-based metaheuristic inspired by the phenomenon of interactions between molecules in a chemical reaction process. CRO has demonstrated its competitive edge over existing methods in solving many real-world problems. However, all studies concerning CRO have been empirical in nature and no theoretical analysis has been conducted to study its convergence properties. In this paper, we present some convergence results for several generic versions of CRO, each of which adopts different combinations of elementary reactions. We investigate the limiting behavior of CRO. By modeling CRO as a finite absorbing Markov chain, we show that CRO converges to a global optimum solution with a probability arbitrarily close to one when time tends to infinity. Our results also show that the convergence of CRO is determined by both the elementary reactions and the total energy of the system. Moreover, we also study and discuss the finite time behavior of CRO.**

***Index Terms*—Chemical reaction optimization (CRO), convergence, convergence rate, finite absorbing Markov chain, first hitting time.**

I. Introduction

CHEMICAL reaction optimization (CRO), as introduced in [1], is a recently proposed metaheuristic for combinatorial optimization problems. CRO loosely mimics interactions of molecules in a chemical reaction, based on the principle that reactions yield products with the lowest energy on the potential energy surface (PES). CRO has demonstrated its splendid performance in solving many real-world optimization problems, e.g., spectrum allocation in cognitive radio systems [2], population transition in peer-to-peer live streaming [3], and task scheduling in grid computing [4]. Recently, it has also been applied to nonconvex continuous problems [5]. For more information about the state-of-the-art development of CRO, the interested reader may refer to [6].

Manuscript received April 19, 2012; revised August 23, 2012; accepted October 23, 2012. Date of publication November 16, 2012; date of current version September 27, 2013. This work was supported in part by the Strategic Research Theme of Information Technology of the University of Hong Kong. The work of A. Y. S. Lam was supported in part by the Croucher Foundation Research Fellowship. The work of V. O. K. Li was supported in part by a Visiting Professorship at the Department of Computer Engineering, King Saud University, Saudi Arabia.

A. Y.S. Lam is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (e-mail: albertlam@ieee.org).

V. O. K. Li and J. Xu are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam, Hong Kong (e-mail: vli@eee.hku.hk; xujin@eee.hku.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2012.2227973

In this paper, we focus on the combinatorial optimization problem whose solution space is discrete, usually large and finite. In general, a combinatorial optimization problem can be described as follows:

$$\min \quad \{C(x)|x \in \mathcal{X}\} \tag{1}$$

where $\mathcal{X} \subset \mathcal{R}^N$ is the set of feasible solutions in the N-dimensional real number space, $C(x) : \mathcal{X} \to \mathcal{Y} \subset \mathcal{R}$ is the cost function, and N is the dimension of the problem. Space $\mathcal{X}$ is constructed according to all the constraints imposed on the problem. Our goal is to find an optimal solution x^* in $\mathcal{X}$, which minimizes the cost function value such that $C(x^*) \leq C(x)$ for all $x \in \mathcal{X}$. Without loss of generality, we assume that $0 < C(x) < \infty$ for all $x \in \mathcal{X}$[1], which can easily be satisfied by adding a sufficiently large number to the objective function to construct a positive $C(x)$. In this way, $C(x)$ can be viewed as the potential energy (*PE*) of the molecules in CRO.

Many theoretical analyses have been conducted to study the convergence of other popular evolutionary algorithms (EAs), e.g., genetic algorithm (GA) [8]–[10], simulated annealing (SA) [11], ant colony optimization (ACO) [12]–[15], and threshold accepting (TA) [16]. By using the homogeneous Markov chain, [10] shows that the canonical GAs with proportional selection will never converge to the global optimum, while its elitist variants do. Reference [11] proves that given a sufficiently large number of iterations at each temperature, SA converges almost surely to the global optimum. Reference [15] gives a short convergence proof for a class of ACO indicating that the probability of finding an optimal solution tends to unity when the algorithm is allowed to run forever. Reference [16] suggests that TA will converge to the global optimum provided that the corresponding version of SA does. This is proved under the fact that SA usually belongs to the convex hull of TA. On the other hand, studies regarding the finite time behavior of EAs have attracted great attention. Rudolph [17] summarized some results on the limiting and finite time behavior of EAs. Reference [18] gives a survey on the time complexity of EAs for combinatorial optimization developed in the recent decade. Moreover, the convergence rate and the expected first hitting time of EAs are also investigated in [19] and [20], respectively. However, those EAs discussed above belong to GA or its variants. In other words, only the finite

[1]Any solution with infinite objective function value is considered as an infeasible solution [7]. The issue of nonnegativity has been discussed in details in [5]. Here we make a stronger assumption of precluding the zero objective function value.

time behavior of GA has been studied. To summarize, all these results explain why the corresponding algorithms work well in practice.

CRO has been applied to tackle various types of nondeterministic polynomial-time hard (NP-hard) problems. Simulation results show that CRO is very competitive in solving many real-world NP-hard problems. Although abundant experimental and empirical research has already been done on CRO, to the best of our knowledge, no theoretical work has been conducted to analyze the performance of CRO. It is important to study the convergence properties of CRO. These properties allow a better understanding of the algorithm and facilitate the design of the algorithmic operators. This paper is the first study of the limiting and finite time behavior properties of CRO.

Considering the characteristics of CRO's algorithmic design, Markov chains offer an appropriate model to analyze its convergence properties. In this paper, we model CRO as a finite absorbing Markov chain and analyze its convergence based on the selection of the elementary reactions adopted in the algorithm and the total energy (*TE*) of the system. In particular, we provide a necessary condition for CRO's convergence from the perspective of elementary reaction design. With the optimum reachability condition, we study the influence of the *TE* on convergence of several versions of CRO with different combinations of elementary reactions. A lower bound of *TE* that can guarantee CRO's convergence is derived. Moreover, we also explore the finite time behavior of CRO.

The remainder of this paper is organized as follows. In Section II, we revisit CRO and describe the corresponding terminologies that will appear in the later proofs. In Section III, we introduce some definitions and model CRO as a finite absorbing Markov chain. Various results of CRO's convergence are given in Section IV. In Section V, we study the finite time behavior of CRO. In Section VI, we discuss the significance and the limitation of the current convergence proofs. Section VII concludes this paper by summarizing our results and proposing potential future work.

II. Chemical Reaction Optimization

In this section, we will first give the basic principles of CRO, explaining the relationship between chemical reactions and optimization. Then, we discuss the energy exchanges and triggering criteria of the elementary reactions in CRO. Next, we describe the flow of the algorithm, and give a brief survey on the state of the art of CRO. Finally, we will explain the unique characteristics of CRO, pointing out the differences between CRO and other EAs.

A. Basic Principles

CRO introduced in [1] is designed based on the nature of chemical reactions, where molecules continuously change in an attempt to attain the lowest free energy. CRO loosely couples optimization with chemical reactions, and thus the details of chemical reactions, e.g., quantum and statistical mechanics, are not captured in the canonical design of CRO. The manipulated agents in CRO are molecules, each of which maintains a certain number of attributes, including a molecular structure, potential energy (*PE*), kinetic energy (*KE*), the number of hits, the minimum hit number, and the minimum value. These attributes are further illustrated as follows.

1) *Molecular structure*: It represents the feasible solution of the optimization problem currently attained by the molecule.
2) *PE*: It quantifies the molecular structure in terms of energy and is modeled as the cost function value of the optimization problem.
3) *KE*: It is associated with the degree of the molecule's activity. In CRO, it symbolizes the solution's ability of jumping out of local optima. The larger the value of *KE*, the more likely the molecular structure can have a rigorous change.
4) *Number of hits*: It counts the number of hits[2] experienced by the molecule.
5) *Minimum hit number*: We record the hit as the minimum hit number when the molecule possesses the current best solution. Thus, the value of (number of hits − minimum hit number) is the number of hits that the molecule has experienced without finding a better solution. This is also used as the criterion for decomposition (one of the four elementary reactions).
6) *Minimum value*: It is the cost function value of the solution generated at the time when the minimum hit number is updated. In other words, it is the minimum *PE* experienced by the molecule itself.

All quantities related to energy (e.g., *PE* and *KE*) should have nonnegative values.

Imagine that there is a closed container with a certain number of molecules. These molecules collide and undergo elementary reactions, which may modify their molecular structures and the attained energies. In other words, elementary reactions are operators, which update the solutions manipulated by the algorithm. Through a random sequence of elementary reactions, CRO explores the solution space and converges to the global minimum. We define four types of elementary reactions, namely, on-wall ineffective collision, decomposition, intermolecular ineffective collision, and synthesis, each of which has different characteristics. The two ineffective collisions (i.e., on-wall ineffective collision and intermolecular ineffective collision) correspond to the cases, in which molecules are modified to new molecular structures close to the original ones, thus enabling the molecules to search their immediate surroundings on the PES. Conversely, decomposition and synthesis tend to produce new molecular structures that may be very different from the original ones. They provide molecules with the ability to jump to other regions on the PES for better solutions after conducting a certain number of local searches. In this way, molecules explore different parts of the PES (solution space) and the elementary reactions gradually bring them toward the lowest energy state. With the four types of elementary reactions, the

[2]A hit of a molecule corresponds to an attempt at one of the elementary reactions, regardless of whether or not this reaction actually takes place. Note that an elementary reaction is only said to have taken place when the new solutions are accepted.

system tries to redistribute the energies (*PE* and *KE*) among the different molecules and central energy buffer (*buffer*).[3] As the process evolves, the molecules get reduced *PE*. In other words, the molecules possess molecular structures with lower and lower *PE* in each subsequent change. This phenomenon is the driving force of CRO to ensure convergence to lower energy state.

B. Elementary Reactions

The energy exchanges and reaction criteria for the four elementary reactions are explained as follows.

1) *On-Wall Ineffective Collision:* It happens when a molecule hits a wall of the container and bounces back. Its molecular structure does not change (from x to x') too rigorously[4] in this case since it tries to transform to a new solution in its neighborhood.[5] The change is allowed if

$$PE_x + KE_x \geq PE_{x'} \tag{2}$$

where PE_x and KE_x are the *PE* and *KE* of the original molecule with molecular structure x, respectively, while $PE_{x'}$ stands for the potential energy of the resultant molecule. If (2) holds, we calculate the new *KE* by

$$KE_{x'} = (PE_x + KE_x - PE_{x'}) \times q \tag{3}$$

where

$$q \in [KELossRate, 1) \tag{4}$$

and $(1-q)$ represents the fraction of *KE* transferred to buffer. *KELossRate* is a system parameter that limits the maximum percentage of *KE* transferred to *buffer* in each of the collisions. For simplicity, we assume that the kinetic energy is transferred to *buffer* with the constant rate μ $(0 < \mu < 1)$ instead of the general case given in (4) in the sequel. In this way, a certain portion of *KE* will be transferred from a molecule to *buffer*. Gradually, when more collisions of this kind take place, total energy of the molecules decrease and energy in *buffer* grows. The energy in *buffer* can be used to support decompositions.

2) *Decomposition:* Decomposition refers to the process that one molecule decomposes into two or more molecules (assume two here). Decomposition is a more rigorous process than the on-wall ineffective collision and leads to resultant molecules with very different molecular structures compared to that of the original molecule. Let x, x'_1, and x'_2 be the original and the two new molecular structures, respectively. Decomposition happens when the molecule itself has sufficient energy (*PE* and *KE*) to satisfy

$$PE_x + KE_x \geq PE_{x'_1} + PE_{x'_2}. \tag{5}$$

When (5) is not satisfied, energy in *buffer* can be used to support the reaction if it is sufficient.[6] The corresponding condition becomes

$$PE_x + KE_x + buffer \geq PE_{x'_1} + PE_{x'_2}. \tag{6}$$

Meanwhile, if (5) holds, we determine the new *KE*s by

$$KE_{x'_1} = temp_1 \times \eta \tag{7}$$

and

$$KE_{x'_2} = temp_1 \times (1-\eta) \tag{8}$$

where $temp_1 = (PE_x + KE_x - PE_{x'_1} - PE_{x'_2})$ and η is a random number generated from $[0, 1]$. Otherwise, if (6) is satisfied, the new *KE*s are computed by

$$KE_{x'_1} = (temp_1 + buffer) \times \eta_1 \times \eta_2 \tag{9}$$

and

$$KE_{x'_2} = (temp_1 + buffer - KE_{x'_1}) \times \eta_3 \times \eta_4 \tag{10}$$

where η_1, η_2, η_3, and η_4 are randomly generated from the interval $[0, 1]$. In the original implementation of CRO [1], decomposition is triggered by the decomposition criterion specified with a parameter α, which represents the maximum time that a selected molecule is allowed to stay in a stable state. In other words, when the condition

$$(\text{number of hits} - \text{minimum hit number}) > \alpha \tag{11}$$

is met, decomposition will happen to drive the molecule to search other regions of the PES.

3) *Inter-Molecular Ineffective Collision:* An inter-molecular ineffective collision describes the situation when two or more molecules collide with each other and then bounce away. This collision is similar to the on-wall ineffective collision as the molecules try to attain new solutions in their own neighborhoods. However, no *KE* is drawn to *buffer* and more molecules (also assume two here) are involved. Consider that x_1 and x_2 change to x'_1 and x'_2, respectively. The change will be accepted only if

$$PE_{x_1} + PE_{x_2} + KE_{x_1} + KE_{x_2} \geq PE_{x'_1} + PE_{x'_2}. \tag{12}$$

Their corresponding *KE*s are determined by

$$KE_{x'_1} = temp_2 \times \xi \tag{13}$$

and

$$KE_{x'_2} = temp_2 \times (1-\xi) \tag{14}$$

where $temp_2 = (PE_{x_1} + PE_{x_2} + KE_{x_1} + KE_{x_2} - PE_{x'_1} - PE_{x'_2})$ and ξ is a random number generated from the interval $[0, 1]$.

[3]In the closed container, according to the law of conservation of energy, the *TE* comprising all *PE*s and *KE*s of all molecules and *buffer* is constant.

[4]Rigorousness describes the degree of change in an elementary chemical reaction. The more rigorous the elementary reaction, the greater the changes in the molecular structures, energy, and other attributes of the involved molecules. The corresponding meaning in optimization is the degree of exploration.

[5]The precise meaning of the neighborhood for a solution will be explained in the next section.

[6]The use of *buffer* to support a decomposition here is a little bit different from that adopted in [1]. This minor change depends on how we implement the algorithm.

4) *Synthesis:* A synthesis depicts two or more molecules (also assume two) that collide and combine into one. Therefore, the resultant molecular structure would be quite different from those of the original molecules. Synthesis will happen only if

$$PE_{x_1} + PE_{x_2} + KE_{x_1} + KE_{x_2} \geq PE_{x'}. \tag{15}$$

If (15) holds, we calculate the KE of the new molecule by

$$KE_{x'} = PE_{x_1} + PE_{x_2} + KE_{x_1} + KE_{x_2} - PE_{x'}. \tag{16}$$

The resultant molecule is normally given with larger KE, i.e., greater ability to escape from a local minimum in subsequent elementary reactions involving this molecule. In the implementation of synthesis, the synthesis criterion is described with parameter β, which defines the least amount of KE a molecule should possess. When both reactant molecules have KE lower than β, that is

$$KE_{x_1} \leq \beta \text{ and } KE_{x_2} \leq \beta \tag{17}$$

synthesis will be triggered.

As a whole, Conditions (2), (5), (6), (12), and (15) enforce the conservation of energy.

C. Algorithm

Generally, CRO is composed of three stages: initialization, iterations, and the final stage. In initialization, we randomly generate a population of molecules, determine their PEs, and assign them with proper initial KEs. Other variables and control parameters (i.e., *buffer*, *KELossRate*, α, β, and *MoleColl*) are also configured at this stage. Then, CRO enters into the stage of iterations. In each iteration, we will first decide whether the collision is uni-molecular or inter-molecular. To do this, a random variable u (uniformly distributed in the interval of [0, 1]) is generated and compared with the system parameter *MoleColl*. If u is greater than *MoleColl*, the collision is considered to be uni-molecular. Otherwise, the collision is inter-molecular. For a uni-molecular reaction, one molecule is randomly selected from the population and we check the decomposition criterion, i.e., (11), against the molecule. If the decomposition criterion is satisfied, the molecule will undergo a decomposition; otherwise, an on-wall ineffective collision results. For an intermolecular collision, two molecules are randomly chosen and they are checked against the synthesis criterion, i.e., (17). If the synthesis criterion is met, the molecules will undergo a synthesis. Otherwise, we will have an inter-molecular ineffective collision. At the end of each iteration, the algorithm checks if any new solution with an objective function value superior to the best-so-far solution is found. If so, the solution will be kept in memory. This iteration stage repeats until any one of the stopping criteria is matched. It may be defined based on the maximum amount of CPU time used, the maximum number of function evaluations computed, or the maximum number of iterations performed without improvements, etc. In the final stage, the solution with the lowest cost function value (i.e., optimal solution) is outputted. The pseudocode of CRO is shown in Table I.

TABLE I
PSEUDOCODE OF THE CRO ALGORITHM

```
Stage 1:  1.1: Assign proper values to system parameters: MoleColl,
              KELossRate, population size (n), α, and β
          1.2: Generate a certain number (n) of solutions randomly,
              calculate their cost function values as the PEs, and
              initialize other attributes for each molecule
Stage 2:  While the stopping criterion not satisfied, do
          2.1: Generate u randomly from [0, 1]
              If u > MoleColl then
                Choose a molecule randomly from the population
                If number of hits − minimum hit number ≥ α then
                  Try decomposition
                Else
                  Try on-wall ineffective collision
              Else
                Select two molecules (ω1 and ω2) randomly from the
                population
                If KE_x1 ≤ β & KE_x2 ≤ β then
                  Try Synthesis
                Else
                  Try inter-molecular ineffective collision
          2.2: If the corresponding collision is triggered then
                Substitute the original molecule(s) with the new one(s)
                in the population, and update the buffer
              Else
                Keep the original molecule(s) in the population
          2.3: Check if there is any new solution with an objective
              function value superior to the best-so-far solution and
              update the best solution found
Stage 3:  Output the solution with the lowest cost function value
```

D. A Brief Survey

CRO is the first metaheuristic that mimics chemical reactions to construct optimization mechanisms. It was first proposed in [1], where its applicability in solving two well-known NP-hard problems in operations research, namely, the quadratic assignment problem and the resource-constrained project scheduling problem, and a practical channel assignment problem in wireless mesh networks is demonstrated. Since then, CRO has been applied broadly to problems in many different domains.

1) Communications and networking: population transition problem in peer-to-peer live streaming [3], cognitive radio spectrum allocation problem [2], [28], and network coding optimization problem [26].
2) Computing: grid scheduling problem [4].
3) Finance: stock portfolio selection problem [27].
4) Computational intelligence: artificial neural network training [25] and fuzzy rule learning problem [21].
5) Energy and environment: optimal power flow problem [22] and sensor deployment problem for air pollution monitoring [23].
6) Bioinformatics: short adjacent repeat identification problem [38].

In addition, different versions of CRO have been developed for different problem types. CRO is originally focused on combinatorial problems in [1]. We have also developed a continuous version for the continuous problems in [5] and the variants with different perturbation functions in [24]. In order to reduce the influence on performance due to a control parameter setting, we proposed an adaptive design in [5] and an adaptive framework in [29] to reduce the number of

parameters required. More information about the state of the art can be found in [30].

E. Characteristics

Similar to other metaheuristics, CRO shares many similarities with other EAs. They all are inspired by Nature. In terms of algorithmic structures, after initialization, they undergo a certain number of iterations until a stopping criterion has been satisfied. Each algorithm contains a number of control parameters and their best values depend on the problem being solved. The performance of the algorithms usually cannot be guaranteed and matching between algorithms and problems depends on the experience and preferences of researchers and practitioners [31].

However, CRO has some unique features that make it fundamentally different from other EAs.

1) The manipulated agents are molecules and the events for manipulating the solutions represented by the molecules are classified into four kinds of elementary reactions. In each iteration of the algorithm, only one elementary reaction will take place, depending on the conditions of the chosen molecules for that iteration.
2) CRO has a variable population size, which is adaptive to the problem being solved. When diversification is required, decompositions are triggered and more molecules are created to explore the solution space. When intensification is preferred, the algorithm initiates syntheses; merging of molecules results in higher probability for the resultant molecules to be selected for manipulation.
3) The underlying principle of CRO is the conservation of energy. The total amount of energy held by the molecules and the buffer remains constant. During the process, we redistribute the energies among the molecules and convert the energy from one form to another (among PE, KE, and buffer).
4) CRO is proposed as an optimization framework. Dedicated problem-specific heuristics can easily be incorporated into the elementary reactions.
5) CRO is easy to parallelize. Several CROs can be carried out simultaneously on the same problem and synchronization among the CROs is, in general, unnecessary.

A more thorough comparison of CRO with other algorithms can be found in [1].

III. Finite Absorbing Markov Chain Model

By referring to the composition of a molecule given in Section II-A, a molecule ω can be represented by an element in the set given by the Cartesian product $\mathcal{X} \times \mathcal{I}$, where $\mathcal{I}$ characterizes the set of additional information, including *KE*, the number of hits, the minimum hit number, and the minimum value. Thus, a population consisting of n molecules is an element of $(\mathcal{X} \times \mathcal{I})^n$. Before showing the definitions and the results, we first make some assumptions as follows.

A1) Total energy (TE) of the system is finite, that is

$$TE \triangleq \sum_{i=1}^{n} (PE_i + KE_i) + buffer < \infty. \tag{18}$$

A2) For a combinatorial problem A, we denote the minimum and maximum cost function values over all solutions in the solution space $\mathcal{X}$ by $C^A_{\min}$ and $C^A_{\max}$, respectively such that $C^A_{\min} \leq C^A(x) \leq C^A_{\max}, \forall x \in \mathcal{X}$. They also satisfy $C^A_{\min} > 0$ and $C^A_{\max} < \infty$.

A3) There are a finite number of possible values for *KE*. This can be justified as computers can only have finite numbers of representation of real numbers.

Note that A1) is always satisfied if A2) and A3) are as long as the initial value of *buffer* is finite, and this value is controlled by the user at initialization of the algorithm mentioned in Section II-C.

Intuitively, a state of the system is determined by the population of the molecules, each of which is an element of $\mathcal{X} \times \mathcal{I}$. As the size of the population changes from time to time due to decomposition and synthesis, simply constructing a state of the Markov chain with the size-variant population will make it extremely difficult to analyze. We have Lemma 1 and Definition 1 to fix the number of variables describing each state of the Markov chain.

Lemma 1: [7]The maximum number of molecules $n_{\max}$ is finite. ◇

Definition 1 (Pseudo-Molecule): A pseudo-molecule $\omega_\emptyset$ is an imaginary molecule with no attributes. In other words, a pseudo-molecule does not have a molecular structure, *PE*, *KE*, and other attributes. It is not actually involved in any elementary reaction, but participates virtually in decomposition and synthesis. ◇

The purpose of introducing the pseudo-molecule is to keep the population size of the system constant. Consider a decomposition that converts ω into ω'_1 and ω'_2. We can interpret it as $\omega+\omega_\emptyset \rightarrow \omega'_1+\omega'_2$. Similarly, consider a synthesis that combines ω_1 and ω_2 into ω'. It can be interpreted as $\omega_1+\omega_2 \rightarrow \omega'+\omega_\emptyset$. In this way, the decomposition and synthesis does not change the total number of molecules in the system. Note that the original meanings and implementations of decomposition and synthesis are still the same. We do not need to consider pseudo-molecules in on-wall ineffective collisions (e.g., $\omega \rightarrow \omega'$) and inter-molecular ineffective collisions (e.g., $\omega_1+\omega_2 \rightarrow \omega'_1+\omega'_2$) as they always keep the number of molecules unchanged.

After defining a state, we can construct the state space accordingly.

Definition 2 (State Space): Given a combinatorial problem A, a state of CRO can be represented by

$$S^A \triangleq \omega^A_1 \times \omega^A_2 \times \cdots \times \omega^A_{n_{\max}}$$

where $n_{\max}$ is the maximum population size given in (25) and $\omega_i \in \mathcal{X} \times \mathcal{I}$ for $1 \leq i \leq n_{\max}$. The whole state space of CRO is defined by

$$\Omega^A = \cup S^A. \qquad \diamond$$

[7]The proofs of all the Lemmas, Theorems, and Corollaries are included in the Appendices.

A state S^A captures the characteristics of CRO at a particular time instant for Problem A and Ω^A includes all possible states explored by CRO for Problem A. Although the number of molecules in the system may change due to decomposition and synthesis, the introduction of pseudo-molecule makes the dimensions of the state space unaltered. It is also worth mentioning that it is not necessary to involve *buffer* in S^A although *buffer* may change after an elementary reaction. Due to the conservation of energy, *TE* is kept constant in the whole course of the algorithm. Once the distributions of *PE* and *KE* over the $n_{\max}$ molecules have been specified by $\omega_1^A, \ldots, \omega_{n_{\max}}^A$, *buffer* can then be determined. Thus the definition of S^A has already captured *buffer*. We have Theorems 1 and 2 and Definition 3 to describe the properties of the state space of the Markov chain, which can be used to model CRO.

Theorem 1: The state space Ω^A is finite. ◇

Here, we consider that CRO evolves in discrete time and one state is chosen at each time slot. Let $S_t^A \in \Omega^A$, for $t = 0, 1, 2, \ldots$ be the state of the random process at time t. Then, we have the following result.

Theorem 2: The evolving process of CRO on solving Problem A can be modeled by a Markov chain $\{S_t^A\}_{t=0}^{+\infty}$, where $S_t^A \in \Omega^A$ for $t = 0, 1, \ldots$ ◇

Definition 3 (Optimal State Set of CRO): For a combinatorial problem A, $\Omega_{\text{opt}}^A \subseteq \Omega^A$ is called the optimal state set of CRO if for all S^A in Ω_{opt}^A, there exists x_{opt} in $\mathcal{X}_{\text{opt}}$, which is the set of optimal solutions, such that x_{opt} is an entry of the vector S^A. ◇

Note that there may be more than one optimal solution for Problem A. Definition 3 indicates that each optimal state of CRO consists of at least one optimal solution.

As seen from Step 2.3 of Table I, the algorithm records the solution with the best objective function value found. We formally define the best-so-far solution as follows.

Definition 4 (Best-So-Far Solution): For Problem A, the best-so-far solution at time t, $x_t^{A,bsf}$, is the current best solution found by the algorithm up to time t, where $t = 0, 1, 2, \cdots$. ◇

When CRO starts, $x_0^{A,bsf}$ is chosen from the initial solution with the lowest cost function value. In the subsequent time $t+1$, $x_{t+1}^{A,bsf}$ is updated by a newly generated solution at time $t+1$, provided that the cost function value of the new solution is lower than that of $x_t^{A,bsf}$. Otherwise, $x_{t+1}^{A,bsf}$ is equal to $x_t^{A,bsf}$. Incorporating the best-so-far solution with the Markov chain allows us to construct an absorbing Markov chain for CRO. We first define an absorbing Markov chain as follows.

Definition 5 (Absorbing Markov Chain): Let Ω be the set of absorbing states, where the transition probability from an absorbing state to itself is one. A Markov chain $\{S_t\}_{t=0}^{+\infty}$ is said to be absorbing if it satisfies

$$P\{S_{t+1} \notin \Omega | S_t \in \Omega\} = 0 \quad t = 0, 1, 2, \ldots \qquad \diamond$$

Indeed, we do usually record the best-so-far solution in each iteration in practice (see Table I). Thus, we can model CRO as an absorbing Markov chain by attaching an additional term to each state defined in Definition 2, which is

$$S_t^A \triangleq x_t^{A,bsf} \times \omega_1^A \times \omega_2^A \times \cdots \times \omega_n^A$$

where $x_t^{A,bsf} \in \mathcal{X}$. Hence, in this case, the state space $\Omega^{A,bsf}$ constituted by S_t^A remains finite since the number of different possible $x_t^{A,bsf}$ is finite. From now on, we will focus on $\Omega^{A,bsf}$ instead of Ω^A. To simplify the notation, we abbreviate Ω_A^{bsf} by Ω_A.

Before discussing the convergence proof of CRO, we first define convergence. Taking into account the probabilistic nature of the state transition behind CRO, we adopt the stochastic convergence defined in [32].

Definition 6 (Convergence): Given an absorbing Markov chain $\{S_t^A\}_{t=0}^{+\infty}$ on the problem A, CRO is said to converge to the global optimum in probability if it satisfies

$$\lim_{t\to\infty} P\{S_t^A \in \Omega_{\text{opt}}^A\} = 1. \qquad \diamond$$

This implies that CRO will produce a global optimal solution almost surely as time tends to infinity.

We set up the basic framework for the rest of this paper. We will utilize an absorbing Markov chain discussed here to prove the convergence of CRO in the next section. We summarize this section as follows. We model CRO as a finite absorbing Markov chain in Theorems 1 and 2, which rely on the construction of the maximum number of molecules in Lemma 1, pseudo-molecule in Definition 1, and the state space in Definition 2. We identify the goal of proving convergence as reaching a state in the optimal state set defined in Definition 3. According to the characteristics of CRO, we define the best-so-far solution in Definition 4, which is the key of the proof. We incorporate this key into the Markov chain model in Definition 5. Finally, we formally define convergence in terms of the Markov chain in Definition 6.

IV. Convergence Proof

In this section, we study CRO's convergence from two different aspects, i.e., the operator designs for the elementary reactions and the total energy of the system. They are the main factors affecting CRO's convergence. Note that the convergence considered here is based on Definition 6. Similar to [33], we present some definitions and lemmas that will be useful in the proofs before investigating these two factors.

Definition 7 (Non-Recessionary Sequence): Let $\{\sigma_t\}_{t=0}^{+\infty}$ be a sequence, where $0 \le \sigma_t \le 1$ for all $t \ge 0$. The sequence $\{\sigma_t\}$ is non-recessionary if it satisfies

$$\prod_{t=0}^{+\infty}(1-\sigma_t) = 0. \qquad \diamond$$

We try to relate the convergence of the algorithm to a non-recessionary sequence with Lemma 2.

Lemma 2: Given an absorbing Markov chain $\{S_t\}_{t=0}^{+\infty}$, there exists a non-recessionary sequence $\{\sigma_t\}_{t=0}^{+\infty}$ such that $P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} \ge \sigma_t$ for $t = 0, 1, 2, \ldots$ if and only if the Markov chain is said to be convergent. ◇

With this lemma, we can give a convergence result whenever the algorithm is associated with a Markov chain that constitutes a non-recesionary sequence.

A. A Short Intuitive Proof

Before we start the detailed convergence proof, we give the underlying intuitive ideas. Solutions explored by the algorithm are maintained by the molecules and the four types of elementary reactions transform the solutions explored. The on-wall and inter-molecular ineffective collisions try to exploit local regions looking for the local minima. In order to prevent the algorithm from getting stuck at some local minima, decompositions and syntheses jump the algorithm to other regions for further exploration. If the algorithm is allowed to run infinitely, the process continues and the algorithm can check every region. Thus, a global optimum can be located eventually. Since the best-so-far solution is updated from time to time, the global optimum found will not be missed when the algorithm terminates.

Assume that the operators are designed properly such that they are recurrent, i.e., the probability of transiting from solutions i to j, $p(i \to j) > 0$ for all $i, j \in \mathcal{X}$. Because of the independence of different iterations, after t iterations, the probability for a nonoptimum solution not reaching a global optimum is $(1-p)^t$. Hence, the probability for a nonoptimum solution reaching a global optimum is $1-(1-p)^t \to 1$ as $t \to \infty$. Therefore, CRO converges to a global optimum solution with probability one when time tends to infinity.

B. Sketch of the Detailed Proof

As mentioned in Section II-E, elementary reactions and energy manipulation are the key characteristics of CRO. Thus, we demonstrate the proof at two levels. We start with operator (elementary reaction) designs. First, we examine some examples of operators used in the literature and then model the operators together with the solution space as a solution graph. Next, we introduce optimum reachability, a property of a solution graph, and give the necessary condition for the convergence of CRO in the context of the optimum reachable solution graph. Note that the result is not restricted to the examples provided; it gives the baseline to design operators in order to drive convergence of the algorithm. In other words, this provides guidelines to design operators for good performance.

Next, we analyze the convergence in terms of energy. We study several variants of CRO with different subsets of elementary reactions. We show that CRO will converge only when all the four elementary reactions are included in the algorithm. The four elementary reactions can complement each other in the way they manipulate the energy so that the algorithm will continue to strive for better solutions but will not get stuck in local minima.

C. Operator Designs

In Section II, we introduce the four elementary reactions of CRO and present their triggering conditions. We also provide ideas on how to design operators for the elementary reactions. The general rule is that the on-wall ineffective collision and the inter-molecular ineffective collision are responsible for neighborhood search[8] while decomposition and synthesis mainly explore regions in the solution space farther away from the existing ones. We usually regard the two ineffective collisions as intensification or exploitation, whereas the other two refer to diversification or exploration.

However, there is no strict requirement for the definition of the neighborhood of a solution in CRO. Roughly speaking, those solutions with relatively slight changes to a specific solution constitute the neighborhood for that solution, but this is not always the case. Consider the pseudo-Boolean problem [34] as an example, where a solution is a binary vector. We may apply the one-bit change operator (e.g., adopted in [4]) to the on-wall ineffective collision, where x' is produced from the neighborhood of x. We randomly select one bit of the vector and then flip it, for example

$$\underbrace{[0\ 1\ 1\ 0\ \underline{1}\ 0]}_{x} \to \underbrace{[0\ 1\ 1\ 0\ \underline{0}\ 0]}_{x'}.$$

Thus, the solution [0 1 1 0 1 0] has six candidates in its neighborhood. In another case, we may also employ the bitwise change operator with probability $1/n$ (e.g., adopted in [20]) in the on-wall ineffective collision, where n is the length of the bit string of a solution. Each bit is flipped with the probability $1/n$, for example

$$\underbrace{[0\ \underline{1}\ 1\ 0\ \underline{1}\ 0]}_{x} \to \underbrace{[0\ \underline{0}\ 1\ 0\ \underline{0}\ 0]}_{x'}$$

with the flipping probability equal to $1/6$. In this case, there are $2^6 = 32$ candidates in the neighborhood for each solution. Besides the on-wall ineffective collision, these neighborhood search operators can also be used in the inter-molecular ineffective collisions.

As for decomposition and synthesis, examples of their operator designs adopted from [4] are shown as follows.

1) Decomposition (half-random change)

   $[\underline{0}\ \mathbf{1}\ \underline{1}\ \mathbf{0}\ \underline{1}\ \mathbf{0}] \to [\underline{0}\ 0\ \underline{1}\ 0\ \underline{1}\ 1] + [1\ \mathbf{1}\ 1\ \mathbf{0}\ 0\ \mathbf{0}]$

   where the first generated solution inherits the same value in the odd positions from the original solution and randomly choose value for the even positions while the second solution inherits the even positions. The noninherited positions of the resultant solutions are randomly generated.

2) Synthesis (half recombination)[9]

   $$[\underline{0\ 0\ 1}\ 0\ 1\ 1] + [1\ 1\ 1\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}] \to [\underline{0\ 0\ 1}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}]$$

   where the new solution has the same entries in the first half positions as those in the first original solution, and its remaining entries are inherited from the second original solution.

In fact, there are numerous ways to design operators for the four elementary reactions. Essentially, all operators define the one-hop candidates of solutions, which are defined as follows.

Definition 8 (One-Hop Candidate Set): Consider a time-variant operator $\Lambda_t : \mathcal{X} \to \mathcal{X}$, which can change with time t. $\Psi_{\Lambda_t}(x)$ denotes the one-hop candidate set of the solution

[8]For a given solution, neighborhood search produces a new solution in the neighborhood of that solution.

[9]This operator is also called one-position exchange in [4].

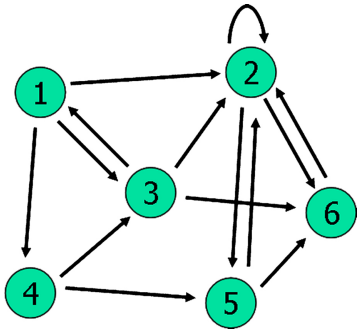


Fig. 1. Example of the static directed graph $G(V, E)$ with $V = 6$ under operator Λ.

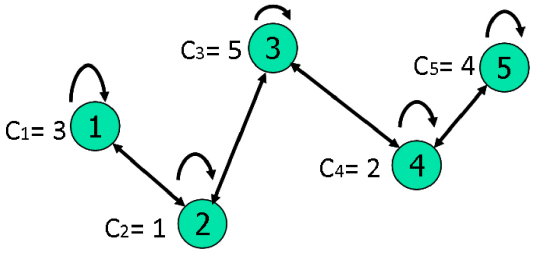


Fig. 2. Example of the solution graph $G(V, E)$ with $V = 5$.

$x \in \mathcal{X}$ at time t and it consists of all possible solutions that can be generated by using operator Λ_t on x only once. ◇

Despite the operators can be designed to vary with time, for simplicity, in this paper we assume that they are time invariant, i.e., being kept constant during the whole evolving process of CRO. This assumption is realistic as most operators are time-invariant in most applications of CRO, e.g., [1]–[5]. Accordingly, the one-hop candidate set $\Psi_\Lambda(x) \subseteq \mathcal{X}$ is fixed for each $x \in \mathcal{X}$. With this assumption, the tuple $(\mathcal{X}, (\Psi_\Lambda(x), x \in \mathcal{X}))$ can be associated with a static directed graph $G(V, E)$, where the vertex set V is $\mathcal{X}$ and the edge set is defined based on $\Psi_\Lambda(x) = \{i|(x, i) \in E\}$. We call $G(V, E)$ a solution graph. Fig. 1 gives a typical example of such a graph. In this example, it can be observed that $\Psi_\Lambda(x_1) = \{x_2, x_3, x_4\}$, $\Psi_\Lambda(x_2) = \{x_2, x_5, x_6\}$, $\Psi_\Lambda(x_3) = \{x_1, x_2, x_6\}$, $\Psi_\Lambda(x_4) = \{x_3, x_5\}$, $\Psi_\Lambda(x_5) = \{x_2, x_6\}$, and $\Psi_\Lambda(x_6) = \{x_2\}$. Note that a solution graph defines what solutions can be reached from a particular solution with a specific operator. Consider two solutions x and x'. An edge (x, x') means that the probability of moving to x' from x is larger than or equal to zero. The actual probability also depends on the energy of the molecule holding x. If no such edge exists, the probability is equal to zero.

We define a property of the solution graph related to optimality as follows.

Definition 9 (Optimum Reachability): A static directed graph $G(V, E)$ is said to be optimum reachable, if $\forall x \notin \mathcal{X} - \mathcal{X}_{\text{opt}}$, there exists an optimum solution $x_{\text{opt}} \in \mathcal{X}_{\text{opt}}$ such that we can find at least one directed path from x to x_{opt}, that is

$$x = v_0 \to v_1 \to v_2 \to \cdots \to v_l = x_{\text{opt}}$$

on $G(V, E)$, where l is the number of hops that x needs to get to x_{opt}. ◇

Let $G(V, E^{\text{owi}})$, $G(V, E^{\text{dec}})$, $G(V, E^{\text{imi}})$, and $G(V, E^{\text{syn}})$[10] be the corresponding solution graphs induced by on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis, respectively. Then, we have the following necessary condition for CRO's convergence.

Theorem 3: Let $G(V, E)$ be the solution graph for CRO, where $E = E^{\text{owi}} \cup E^{dec} \cup E^{imi} \cup E^{syn}$. If the initial solutions are randomly generated, the necessary condition for CRO to converge to an optimal solution is

G(V,E) is optimum reachable. ◇

This says that if the solution graph is not optimum reachable, then the algorithm will not converge to an optimal solution.

D. Total Energy of the System

Here, we study the impact of *TE*, defined in (18), on CRO's convergence. All the following results are accountable under the assumption that the solution graph $G(V, E)$ of CRO is optimum reachable, which is the necessary condition for CRO to converge. Here, we consider four versions of CRO composed of different combinations of elementary reactions:

1) on-wall ineffective collision only;
2) on-wall ineffective collision and decomposition;
3) on-wall ineffective collision and synthesis;
4) all four elementary reactions.

These versions can be realized by setting the parameter *MoleColl* and manipulating the decomposition or synthesis criteria. For example, setting *MoleColl* to zero will result in a CRO with on-wall ineffective collisions and decompositions only, i.e., Version II. If we further set α to ∞, we will not have any decomposition, i.e., Version I. We assume that a solution is also included in its own one-hop candidate in any elementary reaction, i.e., a solution is allowed to remain unchanged in a successful on-wall ineffective collision.[11]

We formally define local optima as follows.

Definition 10 (Local Optimum): Given an operator Λ, the solution x is called the local optimum if its cost is smaller than or equal to that of any of its one-hop neighbors, i.e., $C(x) \leq C(x'), \forall x' \in \Psi_\Lambda(x)$, where $\Psi_\Lambda(x)$ is the one-hop candidate set of the solution $x \in \mathcal{X}$.

With this definition, a global optimum is also a local optimum but the converse is not true. Consider the example given in Fig. 2, which is the solution graph of CRO solving a particular problem. There are five feasible solutions and their corresponding cost function values are $C_1 = 3$, $C_2 = 1$, $C_3 = 5$, $C_4 = 2$, and $C_5 = 4$. It can be observed that Solutions 2 and 4 are the local optima while Solution 2 is the only global optimum.

With Lemma 3, we show that CRO will not get stuck in a local optimum.

[10] A synthesis involves the interaction of two molecules, e.g., $x_1 + x_2 \to x'$. We define $\Psi(x_1)$ as all possible solutions produced when x_1 encounters any solution $x_2 \in \mathcal{X}$. Solution graph $G(V, E^{syn})$ is constructed based on this definition.

[11] In such a case, we still consider that an on-wall ineffective collision has taken place since *KE* and other attributes of that molecule may change.

Lemma 3: Let $\mathcal{X}_{\text{local}}$ be the set of local optima, which also include all global optima. A molecule cannot always attach to the same solution x other than global or local optima, $x \notin \mathcal{X}_{\text{local}}$. In other words, the probability of the molecule getting stuck in a nonoptimum after t iterations equals zero when time t tends to infinity. ◇

We can also interpret Lemma 3 as follows: for any given $\epsilon_1 > 0$, there exists a corresponding natural integer $\delta_1 > 0$ such that the probability of a molecule staying continuously at a solution other than any optima for time $t > \delta_1$ is less than ϵ_1. Indeed, the description is true by letting

$$\delta_1 = \lceil \frac{\log \epsilon_1}{\log(1 - P^{\min}_{x \to x'})} \rceil.$$

Lemma 4: If only on-wall ineffective collisions are allowed, for every number $\epsilon_2 > 0$, there exists a corresponding $\delta_2 > 0$ (representing the number of successful collisions that have taken place) such that after a molecule ω has experienced on-wall ineffective collisions t times, where $t > \delta_2$, its *KE* at time t, denoted by $KE_\omega(t)$, is smaller than ϵ_2, i.e., $KE_\omega(t) < \epsilon_2$. ◇

Lemma 4 implies that a molecule's *KE* will be exhausted in finite time if it is involved in on-wall ineffective collisions only. From Lemmas 3 and 4, we can draw Conclusion 1.

Corollary 1: With on-wall ineffective collisions only, when time t goes to infinity, the molecule will finally reach and stay at a local optimum with probability one.

Therefore, we can show the following impossibility theorem for Version I.

Theorem 4 (for Version I): Regardless of *TE*, CRO with on-wall ineffective collisions only cannot be guaranteed to converge to the global optimum, with the exception that the solution graph $G(V, E)$ is complete. ◇

However, we can propose a variant of Version I, which can converge to the global minimum, as follows.

Corollary 2: There is a variant of CRO with on-wall ineffective collisions only that allows the molecules to regain energy from *buffer* when the following condition holds:

$$(\text{number of hits} - \text{minimum hit number}) \geq \vartheta \tag{19}$$

where ϑ is introduced here as a parameter of CRO. For a problem A, it will converge to the global optimum if the molecule's energy, i.e., $KE_\omega + PE_\omega$, is always larger than or equal to $C^A_{\max}$ after each energy regain, where $C^A_{\max}$ is the maximum cost function value. ◇

With the results obtained for Version I, we can show convergence results for Versions II, III, and IV in the following.

Theorem 5 (for Version II): Regardless of *TE*, CRO with on-wall ineffective collisions and decompositions only cannot be guaranteed to converge to the global optimum in probability, with the exception that $G(V, E^{\text{owi}})$ is complete. ◇

Theorem 6 (for Version III): Regardless of *TE*, CRO with on-wall ineffective collisions and syntheses only cannot be guaranteed to converge to the global optimum in probability, with the exception that $G^{\text{owi}}(V, E)$ is complete. ◇

Theorem 7 (for Version IV): Given a problem A, CRO with all four elementary reactions can be guaranteed to converge to the global optimum, as long as the total energy satisfies the condition $TE \geq 2C^A_{\max}$. ◇

The results in this subsection suggest that in general we should apply CRO with all the four elementary reactions (at least with the on-wall ineffective collision, decomposition, and synthesis) and make the total energy *TE* sufficiently large (e.g., larger than or equal to $2C^A_{\max}$) by controlling the initial *buffer* value and the initial population size so as to guarantee its convergence.

V. Finite Time Behavior of CRO

Many important optimization problems are NP-hard, e.g., the traveling salesman problem (TSP), the minimum graph coloring problem (GCP), the knapsack problem (KP) [35]. They are practical in the sense that many problems in science and engineering can be reduced from these NP-hard problems, e.g., VLSI design problems reduce to TSP [36], wireless sensor network scheduling problems to GCP [37], and DNA sequencing problems to KP [38]. Metaheuristic methods, such as CRO, are very useful in solving such problems since they can usually find the optimal or near-optimal solutions in a tolerable period of time but there is no guarantee. In this section, we study the finite time behavior of CRO. We first introduce the convergence rate and the first hitting time and then discuss how they can be utilized to evaluate the performance of the algorithm.

Definition 11 (Convergence Rate): Given an absorbing Markov chain $\{S_t\}_{t=0}^{\infty}$ and the optimal state space Ω_{opt}, the convergence rate at time t is defined by

$$\pi_t = P\{S_t \in \Omega_{\text{opt}}\} \quad t = 0, 1, 2, \ldots \qquad \diamond$$

The convergence rate at time t is given by the probability that CRO reaches the optimal state by that time. Intuitively, it shows how likely CRO can converge to the global optimal solution in probability at a specific time. With a similar definition of convergence rate, $(1 - \pi_t)$ is considered the convergence rate in [19] and [20]. We adopt Definition 11 to fit the characteristics of CRO better. We can reformulate π_t as follows:

$$\begin{aligned}\pi_t &= 1 - P\{S_t \notin \Omega_{\text{opt}}\} \\ &= 1 - P\{S_0 \notin \Omega_{\text{opt}}\} \prod_{i=0}^{t-1} P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\}\end{aligned}$$

where $P\{S_0 \notin \Omega_{\text{opt}}\}$ depends on the initialization stage of the algorithm. As usual, if we generate the initial set of molecules randomly in the solution space, we have $P\{S_0 \notin \Omega_{\text{opt}}\} = (1 - \frac{||X_{\text{opt}}||}{||X||})^n$. Therefore, the convergence rate at time t is mainly determined by the sequence $\{P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\}\}_{i=0}^{t-1}$. By inspecting the solution graph $G(V, E)$ of CRO, it can be observed that there are two factors affecting the formation of the sequence $\{P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\}\}_{i=0}^{t-1}$, namely, the solution graph corresponding to the elementary reaction realized, and the *KE* of the molecules. The former probably changes with time due to the random sequence of the elementary reactions realized while the latter influences the acceptance of new

TABLE II
PARAMETERS FOR VCRO_1, VCRO_2, VCRO_3, AND VCRO_4

	Parameter	Value
VCRO_1, VCRO_3	*KELossRate*	0.8
	α	3
	Initial *KE*	5−initial *PE*[a]
	Initial *buffer*	0
VCRO_2, VCRO_4	*KELossRate*	0.8
	α	10
	Initial *KE*	5−initial *PE*[a]
	Initial *buffer*	0

[a] In this way, the total energy of the system is equal to the maximum cost function value, i.e., $TE = 5$, which can guarantee CRO's convergence by Corollary 2.

solutions in the graph at different times. In particular, when the operators used in the four elementary reactions are defined, their solution graphs $G(V, E^{\text{owi}})$, $G(V, E^{\text{dec}})$, $G(V, E^{\text{imi}})$, and $G(V, E^{\text{syn}})$ can be determined. However, the graphs realized at time t are decided by all parameters, values of the molecular attributes (e.g., number of hits and minimum hit number), and some random variables (e.g., u given in Section II-C). Similarly, *KE* for each molecule also depends on the CRO parameters and variables. In short, the convergence rate of CRO is significantly impacted by the problem characteristics, the operators configured for elementary reactions, and the parameters configured in CRO.

Definition 12 (First Hitting Time): We introduce a nondecreasing sequence $\{Z_t\}_{t=0}^{\infty}$ such that $Z_t = \mathbf{1}(S_t \in \Omega_{\text{opt}})$, where $\mathbf{1}$ denotes the indicator function, i.e., Z_t equals one if an optimal solution is found in one of the iterations $0, 1, 2, \ldots, t$, or is equal to zero otherwise. The first hitting time T_f refers to the iteration when a global minimum is found by the algorithm the first time, i.e., $T_f = \min\{t \geq 1 : Z_t = 1\}$. ◇

Then, we have the expected first hitting time as

$$\begin{aligned} E[T_f] &= \sum_{t=1}^{\infty} tP\{S_t \in \Omega_{\text{opt}} \cap S_{t-1} \notin \Omega_{\text{opt}}\} \\ &= \sum_{t=1}^{\infty} tP\{S_t \in \Omega_{\text{opt}} | S_{t-1} \notin \Omega_{\text{opt}}\} P\{S_{t-1} \notin \Omega_{\text{opt}}\} \\ &= \sum_{t=1}^{\infty} t[P\{S_t \in \Omega_{\text{opt}}\} - P\{S_t \in \Omega_{\text{opt}} | S_{t-1} \in \Omega_{\text{opt}}\} \\ &\quad \times P\{S_{t-1} \in \Omega_{\text{opt}}\}] \\ &= \sum_{t=1}^{\infty} t[P\{S_t \in \Omega_{\text{opt}}\} - P\{S_{t-1} \in \Omega_{\text{opt}}\}] \\ &= \sum_{t=1}^{\infty} t(\pi_t - \pi_{t-1}). \end{aligned} \tag{20}$$

We can see that the convergence rate and first hitting time are related. Once again, the first hitting time or its expected value are related to the characteristics of the problem, the operators, and all the parameters configured in CRO. On the other hand, by the Markov inequality [39] and Chebyshev's inequality [40], we also have

$$P\{T_f < t\} \geq 1 - \frac{E[T_f]}{t}$$

and

$$P\{|T_f - E[T_f]| < t\} \geq 1 - \frac{V[T_f]}{t^2}$$

where $V[T_f]$ is the variance of T_f and $t > 0$ can be interpreted as the time when the algorithm terminates (see stopping criteria). These inequalities can be used as bounds for the efficiency of an algorithm. Although it is hard to determine $E[T_f]$ and $V[T_f]$ exactly, we may estimate these values by repeated trials, which gives us a way to predict the performance of CRO for a particular problem. For instance, consider the example in Fig. 2, in which we employ the variant of CRO described in Corollary 2. For comparison, we consider two different sets of parameters listed in Table II and the resultant algorithms are called VCRO_1 and VCRO_2. The algorithm is set to run until it reaches an optimum and we record the numbers of iterations required as the first hitting time. After repeating the algorithm for a sufficiently large number of runs (say, 10 000 times), we obtain estimations of the expected value and variance for the first hitting time of VCRO_1 and VCRO_2 as $E_1[T_f] = 10.06$ and $V_1[T_f] = 170.47$, and $E_2[T_f] = 16.77$ and $V_2[T_f] = 388.15$, respectively. Accordingly, for VCRO_1, we have

$$P\{T_f < t\} \geq 1 - \frac{10.06}{t} \tag{21}$$

and

$$P\{10.02 - t < T_f < t + 10.02\} \geq 1 - \frac{170.47}{t^2}. \tag{22}$$

Similarly, for VCRO_2, we have

$$P\{T_f < t\} \geq 1 - \frac{16.77}{t} \tag{23}$$

and

$$P\{16.77 - t < T_f < t + 16.77\} \geq 1 - \frac{388.15}{t^2}. \tag{24}$$

From the above inequalities, we can easily draw some conclusions, e.g., the optimal solution can be reached in 69 (calculated based on 22) iterations using VCRO_1 or 105 (calculated based on 24) iterations using VCRO_2, with a probability larger than 95%. To enrich the results, we introduce another variant of CRO, which is similar to the previous one by replacing the neighborhood search operator with a random search operator (i.e., randomly pick a solution in the search space when the operator is called). We employ the same sets of parameter values of VCRO_1 and VCRO_2 and name the corresponding algorithms VCRO_3 and VCRO_4. We also plot the probability lower bounds (21) and (23) in Fig. 3 to see how the performances of VCRO_1, VCRO_2, VCRO_3, and VCRO_4 change with time. Such kinds of results are valuable when we decide the stopping criterion in terms of iterations. It can also be observed from Fig. 3 that VCRO_1 (VCRO_3) converges faster than VCRO_2 (VCRO_4). This means that the first set of parameter values in Table II outperforms the second set. Moreover, VCRO_3 and VCRO_4 converge faster than VCRO_1 and VCRO_2, and thus, the random search operator is better at solving the problem than the neighborhood search operator.

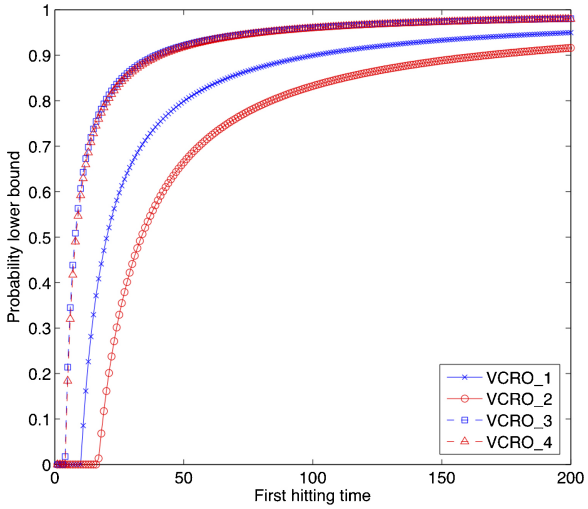


Fig. 3. Probability lower bound for the example.

The above method for evaluating the performance of an algorithm is not only restricted to a particular problem instance. However, once the performance for a problem instance has been determined, the results can be inferred to other instances of the same problem and similar problems.

VI. Discussion

In this section, we will discuss the results obtained, some issues raised in the convergence proofs, and the work on the finite time behavior of CRO.

A. What Do the Convergence Proofs Mean?

Theorem 1 proves that the state space of CRO is finite, and Definition 5 suggests that it may be appropriate to model CRO as an absorbing Markov chain. Thus, CRO can be modeled by a finite absorbing Markov chain, which gives us a proper framework for tackling the convergence proofs. Lemma 2 tells us that CRO's convergence can be proved by finding a non-recessionary sequence. Theorem 3, which studies the impact of operators, gives a necessary condition for the convergence. That is, the solution graph of CRO should be optimum reachable. From the perspective of the total energy of the system, Theorems 4–7 analyze the convergences of several versions of CRO, composed of different combinations of elementary reactions. We show that CRO requires at least three elementary reactions (i.e., on-wall ineffective collision, decomposition, and synthesis) to converge to the global optimum and the least amount of total energy required to guarantee convergence is $2C^A_{\max}$. This amount of energy will enable CRO to have infinite number of decompositions and syntheses, which ensure that CRO will not get stuck at local optima. Corollary 2 demonstrates that a variant of CRO, which consists of on-wall ineffective collisions only but allows recycling of energy from *buffer* back to the molecules can also be guaranteed to converge to the global optimum. As a whole, the convergence proofs offer some guidance on how to design CRO to guarantee its convergence.

B. What Does the Work on the Finite Time Behavior of CRO Mean?

This paper considers the convergence rate and the first hitting time, which are two major metrics to assess the performance of CRO in finite time. However, we find that their calculations depend on the problem types, the operators designed for each reaction, and the parameter set in CRO. Consequently, before conducting further theoretical studies on convergence rate and first hitting time, we need to identify certain types of problems and specific operators.

C. Limitations

To the best of our knowledge, this work is the first analytical study on the convergence of CRO. However, there are still some limitations, which present opportunities for future search.

1) *Inefficient Search:* Although we give the conditions of convergence guarantee, the efficiency of CRO may be low when the algorithm behaves like a random search to traverse the whole PES, rather than focusing on some promising regions where global optima are more likely to reside. We have not discussed how to design the operators and to set the parameter values for CRO to enhance its efficiency for a given problem.

2) *Long Searching Time:* In the current proofs, time is allowed to go to infinity to ensure the convergence to global optimum. Nonetheless, since there is no guarantee for the first hitting time, the actual time for convergence may be very long. In other words, we need more analysis on the convergence rate and the first hitting time.

3) *Combinatorial Problems:* This paper is restricted to combinatorial problems, which may be modeled as a finite Markov chain. However, many optimization problems have continuous solution space. In this case, there are infinite number of states and the methods used here are no longer appropriate.

VII. Conclusion and Future Work

In this paper, we analyzed CRO and provided some convergence properties of CRO under the framework of Markov process. First, we showed that CRO can be modeled as a finite absorbing Markov chain. Then, we analyzed its convergence from two aspects, including the operators and the total energy of the system. The corresponding results were: 1) a necessary condition for CRO's convergence is that the solution graph must be optimum reachable; 2) canonical CRO can only be guaranteed to converge to the global optimum when it is composed of at least three types of elementary reactions (i.e., on-wall ineffective collision, decomposition, and synthesis) and it has at least $2C^A_{\max}$ total system energy; and 3) a variant of CRO where the molecule is allowed to automatically absorb energy from the central energy buffer can converge with the on-wall ineffective collisions only. These results provided us with some insight into the characteristics of CRO, which is helpful to future designs. Finally, we discussed the significance of the proofs and the limitations of this paper.

In the future, further analytical research on CRO will be carried out in the following three directions.

1) More studies will be conducted to study the finite time behavior of CRO. In particular, we will investigate the effect of the system parameters used in CRO on its convergence rate and on the first hitting time.
2) We will explore the impact of different operator designs on the efficiency of CRO.
3) Theoretical analysis on the convergence of CRO for continuous problems will also be studied.

Appendix A
Proof Lemma 1

By A1), *TE* is finite. By the conservation of energy, *TE* of the system is a constant in the whole process of searching. The maximum number of molecules in the population at any time is bounded above by

$$n_{\max} = \lfloor \frac{TE}{C^A_{\min}} \rfloor < \infty \tag{25}$$

where $C^A_{\min} > 0$ by A2).

Appendix B
Proof of Theorem 1

For the combinatorial optimization problem A, we have a finite number of feasible solutions, which implies that $\mathcal{X}$ is finite. This also implies that the number of possible cost function values, i.e., *PE*, is finite. By A3), the number of possible values of *KE* is also finite. Moreover, both the number of hits and the minimum hit number are bounded by α, which is a predefined integer parameter. Following the finiteness of *PE*, there are a finite number of possible minimum values. Therefore, the set $\mathcal{X} \times \mathcal{I}$ for a molecule is finite. Thus, with Lemma 1, the state space Ω^A is finite.

Appendix C
Proof of Theorem 2

According to the pseudocode of CRO given in Table I, it can be observed that the system state S^A_{t+1} at time $t+1$ is only dependent on the state S^A_t at time t. Namely, we have

$$P\{S^A_{t+1} \in \Omega^A | S^A_0, S^A_1, \cdots, S^A_t\} = P\{S^A_{t+1} \in \Omega^A | S^A_t\} \tag{26}$$

where $P\{\cdot|\cdot\}$ is the transition probability. Equation (26) is the Markov property, and thus, $\{S^A_t\}_{t=0}^{+\infty}$ is a Markov chain with state space Ω^A.

Appendix D
Proof of Lemma 2

Only if ($\Rightarrow$) part: We show that if the transition probability sequence from a transient state to an absorbing state constitutes a non-recessionary sequence, then the Markov chain and thus the algorithm converges. For $t = 0, 1, 2, \ldots,$ $P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} \geq \sigma_t$ implies $P\{S_{t+1} \notin \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} \leq 1 - \sigma_t$. Define

$$\tilde{P}(t) \triangleq \prod_{i=0}^{t} P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\}. \tag{27}$$

Then, we have

$$\begin{aligned}\lim_{t\to\infty} \tilde{P}(t) &= \prod_{i=0}^{+\infty} P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\} \\ &\leq \prod_{i=0}^{+\infty} (1 - \sigma_i) = 0.\end{aligned}$$

Obviously, $\tilde{P}(t) \geq 0$ for all t. Thus $0 \leq \lim_{t\to\infty} \tilde{P}(t) \leq 0$ gives $\lim_{t\to\infty} \tilde{P}(t) = 0$. Since $\{S_t\}_{t=0}^{+\infty}$ is an absorbing Markov chain, immediately we have

$$P\{S_{t+1} \notin \Omega_{\text{opt}} | S_t \in \Omega_{\text{opt}}\} = 0$$

and

$$P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \in \Omega_{\text{opt}}\} = 1.$$

Therefore

$$\begin{aligned}&P\{S_{t+1} \notin \Omega_{\text{opt}}\} \\ &= P\{S_{t+1} \notin \Omega_{\text{opt}} | S_t \in \Omega_{\text{opt}}\} P\{S_t \in \Omega_{\text{opt}}\} \\ &\quad + P\{S_{t+1} \notin \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} P\{S_t \notin \Omega_{\text{opt}}\} \\ &= P\{S_{t+1} \notin \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} P\{S_t \notin \Omega_{\text{opt}}\} \\ &= P\{S_0 \notin \Omega_{\text{opt}}\} \prod_{i=0}^{t} P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\} \\ &= P\{S_0 \notin \Omega_{\text{opt}}\} \tilde{P}(t).\end{aligned} \tag{28}$$

As $\lim_{t\to\infty} \tilde{P}(t) = 0$, we have $\lim_{t\to\infty} P\{S_t \notin \Omega_{\text{opt}}\} = 0$. In other words

$$\lim_{t\to\infty} P\{S_t \in \Omega_{\text{opt}}\} = 1 - \lim_{t\to\infty} P\{S_t \notin \Omega_{\text{opt}}\} = 1.$$

Therefore, the algorithm will eventually reach the optimal state as long as the time allowed to evolve is sufficiently long.

If ($\Leftarrow$) part: We try to show that a convergent Markov chain will result in a non-recessionary sequence. If Markov chain is convergent, by Definition 6, we have

$$\lim_{t\to\infty} P\{S_t \in \Omega_{\text{opt}}\} = 1$$

which is equivalent to

$$\lim_{t\to\infty} P\{S_t \notin \Omega_{\text{opt}}\} = 1 - \lim_{t\to\infty} P\{S_t \in \Omega_{\text{opt}}\} = 0.$$

Then by (28), we have

$$\begin{aligned}\lim_{t\to\infty} \tilde{P}(t) &= \lim_{t\to\infty} \frac{P\{S_t \notin \Omega_{\text{opt}}\}}{P\{S_0 \notin \Omega_{\text{opt}}\}} \\ &= \frac{1}{P\{S_0 \notin \Omega_{\text{opt}}\}} \lim_{t\to\infty} P\{S_t \notin \Omega_{\text{opt}}\} = 0.\end{aligned}$$

According to the definition of $\tilde{P}(t)$, i.e., (27), we get

$$\prod_{i=0}^{+\infty} P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\} = 0.$$

Let $\sigma_i = 1 - P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\}$, i.e., $1 - \sigma_i = P\{S_{i+1} \notin \Omega_{\text{opt}} | S_i \notin \Omega_{\text{opt}}\}$, for $i = 0, 1, 2, \ldots$, then we have

$$\prod_{i=0}^{+\infty} (1 - \sigma_i) = 0.$$

By Definition 7, $\{\sigma_t\}_{t=0}^{+\infty}$ is a non-recessionary sequence. Meanwhile, we also have

$$\begin{aligned} P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} &= 1 - P\{S_{t+1} \notin \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} \\ &= \sigma_t \geq \sigma_t, t = 0, 1, 2, \ldots \end{aligned}$$

Appendix E
Proof of Theorem 3

This can be proved by contraposition. If the solution graph of CRO is not optimum reachable, there exists a solution $\tilde{x} \in \mathcal{X} - \mathcal{X}_{\text{opt}}$ such that the vertex of $\tilde{x}$ on the graph can never reach any one of the optimal solutions. Meanwhile, as the initial solutions of CRO are generated randomly, the system may result in the initial set of molecules, each of which corresponds to the same $\tilde{x}$, with probability $\frac{1}{||\mathcal{X}||^n} > 0$, where $||\mathcal{X}||$ represents the cardinality of $\mathcal{X}$ and n is the initial population size. When this happens, $P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} = 0$ for $t = 0, 1, 2, \ldots$. According to Lemma 2, σ_t equals zero for $t = 0, 1, 2, \ldots$. Thus, there does not exist a non-recessionary sequence satisfying the condition given in Definition 7. Hence, we have

$$\lim_{t \to \infty} P\{S_t \in \Omega_{\text{opt}}\} \leq 1 - \frac{1}{||\mathcal{X}||^n} < 1.$$

Therefore, CRO will not converge to the optimal solution almost surely if its solution graph $G(V, E)$ is not optimum reachable. In other words, the solution graph $G(V, E)$ being optimum reachable is the necessary condition for CRO convergence.

Appendix F
Proof of Lemma 3

Suppose that a molecule currently carries a nonoptimum solution x. By Definition 10, there always exists at least one $x' \in \Psi_\Lambda(x)$ such that $C(x') \leq C(x)$. Let $P_{x \to x'}(t)$ be the probability of a molecule moving from x to x' on the PES at time t. Since the energy condition of the elementary reaction corresponding to Λ, e.g., (2) always holds, $P_{x \to x'}(t)$ is always larger than zero for $t = 0, 1, 2, \ldots$, irrespective of the *KE* value of the molecule. Let $P_{x \to x'}^{\min} = \min\{P_{x \to x'}(t) : t = 0, 1, 2, \cdots\}$; obviously $P_{x \to x'}^{\min} > 0$. Thus, the probability for the molecule staying at x at time t is less than one, i.e., $P_{x \to x}(t) \leq 1 - P_{x \to x'}^{\min} < 1$. Therefore, we have

$$\lim_{t \to \infty} [P_{x \to x}(0) P_{x \to x}(1) \cdots P_{x \to x}(t)] \leq \lim_{t \to \infty} (1 - P_{x \to x'}^{\min})^t = 0$$

which indicates that the probability for a molecule getting stuck at a nonlocal optimum is negligible.

Appendix G
Proof of Lemma 4

$KE_\omega(t)$ represents the *KE* of molecule ω at time t and the initial *KE* of ω is denoted by $KE_\omega(0)$. The maximum amount *KE* that ω can have is $KE_\omega(0) + (C_{\max}^A - C_{\min}^A)$. Consider the δ_2 times of successful collisions which have happened at time t', where $t' \geq \delta_2$ because some collisions may fail, i.e., not satisfying (2). $KE_\omega(t')$ is upper bounded by $(KE_\omega(0) + C_{\max}^A - C_{\min}^A) \times \mu^{\delta_2}$, where μ is the rate of *KE* transferred to *buffer* described in Section II-B1. Trivially, we have $KE_\omega(t) \leq KE_\omega(t')$ for all $t \geq t'$. We can set

$$\delta_2 = \lceil \log \frac{\epsilon_2}{KE_\omega(0) + C_{\max}^A - C_{\min}^A} / \log \mu \rceil.$$

Note that $\delta_2 \leq t'$ only accounts for the successful collisions. By Lemma 3, t' is finite and so is δ_2.

Appendix H
Proof of Theorem 4

With on-wall ineffective collisions only, molecules will not collide or interact with each other based on the definition of the on-wall collision given in Section II-B1. So, each molecule explores the solution space independently and each can be considered a separated system (since energy can only be transferred in one direction from molecules to *buffer*). Without loss of generality, we assume that there is only one molecule ω in the system. According to Lemma 4, its kinetic energy KE_ω will approach zero in finite time. Since this theorem is an impossibility result, it is sufficient to prove it by giving a counterexample.

Consider the example given in Fig. 2 again. Let $\hat{\mathcal{X}} = \{3, 4, 5\}$. Consider the case that the molecule ω always shuffles around Solution 4 (i.e., move within $\hat{\mathcal{X}}$), such that it never visits Solutions 1 and 2. This means that ω always gets stuck in the local optimum Solution 4. By Lemma 4, there exists a time $t' < \infty$ such that when $t > t'$, we have $KE_\omega(t) < 3$. If ω is at Solution 4 at time t', its *KE* can never help ω jump out of the local optimum hereafter. Mathematically, we have

$$P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} = 0, \ t = t', t' + 1, t' + 2, \ldots$$

In this case, if ω does not visit Solutions 1 and 2 before t', it will never be able to determine the global optimum Solution 2.

Since the initial solution attached to ω is generated randomly, we have probability 3/5 that ω falls in $\hat{\mathcal{X}}$. Let $x(t)$ be the solution carried by ω at time $t \geq 0$. There exist many sequences $[x(0), x(1), \ldots, x(t')]$ such that $x(t) \in \hat{\mathcal{X}}, 0 \leq t \leq t' - 1$ and $x(t') = 4$. One such sequence is $[3, 3, \ldots, 3, 4]$: ω picks Solution 3 initially. It stays at Solution 3 until $t' - 1$ and reaches Solution 4 at time t'. Without loss of generality, assume that the collision always produces a solution in the neighborhood evenly. Since Solution 3 has three solution candidates in its neighborhood (i.e., Solutions 2, 3, and 4), the probability of ω reaching a solution in its neighborhood in one collision is $\frac{1}{3}$. The probability of generating such a sequence is $\frac{1}{5} \times (\frac{1}{3})^{t'}$.

As there are many sequences resulting in $x(t') = 4$, the probability of the molecule shuffling around Solution 4 until time t', denoted by $\hat{P}$, is bounded by

$$0 < \frac{1}{5} \times (\frac{1}{3})^{t'} < \hat{P} < \frac{3}{5}.$$

As $\hat{P}$ corresponds to the probability of some cases that will never result in the global optimum, we have

$$\lim_{t\to\infty} P\{S_t \in \Omega_{\text{opt}}\} \leq 1 - \hat{P} < 1.$$

This implies CRO Version I is not convergent in probability.

However, there is an exception when the solution graph $G(V, E)$ is complete.[12] This means that for each solution all other solutions are its one-hop candidates. In other words, the following inequality:

$$P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} > 0,\ t = 0, 1, 2, \cdots$$

will always hold regardless of the energy due to Lemma 3. This can be seen as the global minimum x^* is in the neighborhood of all other solutions with cost function values higher than $C(x^*)$ and moving from a solution with higher cost to another solution with lower cost in its neighborhood is always possible. Let $\sigma_t = P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\}$ for $t = 0, 1, 2, \ldots$, then the sequence $\{\sigma_t\}_{t=0}^{+\infty}$ is non-recessionary. According to Lemma 2, CRO will converge to the global optimum in probability.

Appendix I
Proof of Corollary 2

As the solution graph $G(V, E)$ is optimum reachable, for any solution $x_i \in \mathcal{X} - \mathcal{X}_{\text{opt}}$ there exists at least one path from x_i to x_{opt}: $x_i = v_0 \to v_1 \to v_2 \to \cdots \to v_l = x_{\text{opt}}$. Meanwhile, in each energy regain, the condition $KE_\omega + PE_\omega \geq C^A_{\max}$ will allow the molecule to escape from any local optimum. In other words, the probability of moving from v_j to v_{j+1} on the above path within ϑ steps, denoted by $P(v_j \to v_{j+1})$, is always greater than zero, for $j = 0, 1, \ldots, l(x_i)-1$.[13] Thus, the probability for the molecule moving from x_i to the optimum solution in $l\vartheta$ iterations, denoted by $P(x_i)$, is larger than zero, i.e., $P(x_i) = P(v_0 \to v_1)P(v_1 \to v_2)\cdots P(v_{l-1} \to v_l) > 0$. Define $T(x_i) = l\vartheta < \infty$ to be the expected time for molecule moving from x_i to x_{opt} and let $T^* = \max\{T(x_i) : x_i \in \mathcal{X}-\mathcal{X}_{\text{opt}}\}$. Then at any time $t = 0, 1, \ldots$, there exists at least a $t' \in \{t+1, t+2, \cdots, t+T^*\}$ such that

$$P\{S_{t'+1} \in \Omega_{\text{opt}} | S_{t'} \notin \Omega_{\text{opt}}\} > 0.$$

In other words, within any interval of T^*, there will be a positive value of $P\{S_{t'+1} \in \Omega_{\text{opt}} | S_{t'} \notin \Omega_{\text{opt}}\}$. Without loss of generality, assume that $t_i \in \{iT^*+1, iT^*+2, \cdots, (i+1)T^*\}$, $i = 0, 1, 2, \ldots$ be the time taking aforementioned positive values. Thus, we can construct a non-recessionary sequence $\{\sigma_t\}_{t=0}^{+\infty}$ by setting $\sigma_{t_i} = P\{S_{t_i+1} \in \Omega_{\text{opt}} | S_{t_i} \notin \Omega_{\text{opt}}\}$, $i = 0, 1, 2, \cdots$ and $\sigma_t = 0$ otherwise. In this way, we always have $P\{S_{t+1} \in \Omega_{\text{opt}} | S_t \notin \Omega_{\text{opt}}\} \geq \sigma_t$ for $t = 0, 1, 2, \cdots$. By Lemma 2, this variant of CRO is convergent.

[12]For example, the bitwise change operator described in Section IV-C is a representative operator that makes the solution graph complete.

[13]The number of steps l to reach an optimum depends on the starting point x_i.

Appendix J
Proof of Theorem 5

Note that the total energy *TE* has a finite value by Assumption A1). By Lemma 1, the maximum number of molecules is $n_{\max}$. Since each decomposition produces at least one more molecule, with on-wall ineffective collisions and decompositions only, the number of times that the decomposition happens is finite and bounded above by $n_{\max} - 1$. Evidently, there exists $t' < +\infty$ such that when $t > t'$, there are no more decompositions. In the meantime, the probability of having a nonoptimal state at time $t' + 1$ is positive, i.e., $P\{S_{t'+1} \notin \Omega_{\text{opt}}\} > 0$. After time t', only on-wall ineffective collisions can take place. The remainder of the proof follows Theorem 4.

Appendix K
Proof of Theorem 6

The proof is similar to that of Theorem 5. Each synthesis always reduces the number of molecules by at least one. After a finite time t', there is only one molecule left, and no more synthesis. In this way, the algorithm degenerates to Version I which only allows on-wall ineffective collisions.

Appendix L
Proof of Theorem 7

By Lemma 4, a molecule's *KE* will be exhausted in finite time due to on-wall ineffective collisions. Moreover, by Lemma 1, the maximum number of molecules is bounded above. Thus, once there is more than one molecule in the system, a synthesis will happen in finite time almost surely since it is triggered when the involved molecules possess *KE* less than or equal to β due to the on-wall ineffective collisions. If there is no decomposition, syntheses will happen until only one molecule is left. Consider the worst case with only one molecule in the system. We can guarantee that a decomposition can happen when $TE \geq 2C^A_{\max}$. This can be seen by (5) and each of the two resultant molecules has *PE* at most $C^A_{\max}$. $2C^A_{\max}$ is the sufficient amount of energy to support a decomposition and it will surely happen in finite time. As a result, synthesis and decomposition will be continuously triggered in a roughly alternative manner. Furthermore, the molecule can obtain sufficient *KE* to avoid getting stuck in a local optimum, which may be encountered in on-wall ineffective collisions or inter-molecular ineffective collisions, via random combinations of decompositions and syntheses with probability larger than zero in finite time. This situation is similar to the variant of CRO discussed in Corollary 2. Thus the rest of the proof follows Corollary 2.

Acknowledgment

The authors would like to thank B. Z. Jin for his assistance in the preliminary study of this paper.

REFERENCES

[1] A. Y. S. Lam and V. O. K. Li, "Chemical-reaction-inspired metaheuristic for optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 381–399, Jun. 2010.

[2] A. Y. S. Lam and V. O. K. Li, "Chemical reaction optimization for cognitive radio spectrum allocation," in *Proc. IEEE Global Commun. Conf.*, Dec. 2010, pp. 1–5.

[3] A. Y. S. Lam, J. Xu, and V. O. K. Li, "Chemical reaction optimization for population transition in peer-to-peer live streaming," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[4] J. Xu, A. Y. S. Lam, and V. O. K. Li, "Chemical reaction optimization for task scheduling in grid computing," *IEEE Trans. Parallel Distributed Syst.*, vol. 22, no. 10, pp. 1624–1631, Oct. 2011.

[5] A. Y. S. Lam, V. O. K. Li, and J. J. Q. Yu, "Real-coded chemical reaction optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 339–353, Jun. 2012.

[6] A. Y. S. Lam and V. O. K. Li, "Chemical reaction optimization: A tutorial," *Memetic Comput.*, vol. 4, no. 1, pp. 3–17, Mar. 2012.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, NY: Cambridge Univ. Press, 2004.

[8] Y. Leung, Y. Gao, and Z. Xu, "Degree of population diversity: A perspective on premature convergence in genetic algorithms and its Markov chain analysis," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1165–1176, Sep. 1997.

[9] A. E. Eiben, E. H. L. Aarts, and K. M. van Hee, "Global convergence of genetic algorithms: A Markov chain analysis," in *Parallel Problem Solving from Nature*, H. P. Schwefel and R. Manner, Eds. Berlin. Germany: Springer-Verlag, 1991, pp. 4–12.

[10] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Trans. Neural Netw.*, vol. 5, no. 1, pp. 96–101, Jan. 1994.

[11] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311–329, May 1988.

[12] W. J. Gutjahr, "A generalized convergence result for the graph-based ant system metaheuristic," *J. Probability Eng. Inform. Sci.*, vol. 17, no. 4, pp. 545–569, Oct. 2003.

[13] W. J. Gutjahr, "ACO algorithms with guaranteed convergence to the optimal solution," *Inform. Process. Lett.*, vol. 82, no. 3, pp. 145–153, 2002.

[14] H. Huang, C. Wu, and Z. Hao, "A pheromone-rate-based analysis on the convergence time of ACO algorithm," *IEEE Trans. Syst., Man, Cybern. B*, vol. 39, no. 4, pp. 910–923, Aug. 2009.

[15] T. Stutzle and M. Dorigo, "A short convergence proof for a class of ant colony optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 358–365, Jun. 2010.

[16] I. Althofer and K. U. Koschnick, "On the convergence of 'threshold accepting'," *Appl. Math. Optimization*, vol. 24, no. 1, pp. 183–195, 1991.

[17] G. Rudolph, "Finite Markov chain results in evolutionary computation: A tour d'horizon," *Fundam. Inform.*, vol. 35, nos. 1–4, pp. 67–89, Aug. 1998.

[18] P. S. Oliveto, J. He, and X. Yao, "Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results," *Int. J. Autom. Comput.*, vol. 4, no. 3, pp. 281–293, Jul. 2007.

[19] J. He and X. Yu, "Conditions for the convergence of evolutionary algorithms," *J. Syst. Architecture*, vol. 47, no. 7, pp. 601–612, Jul. 2001.

[20] Y. Yu and Z. Zhou, "A new approach to estimating the expected first hitting time of evolutionary algorithms," *Artif. Intell.*, vol. 172, no. 15, pp. 1809–1832, Oct. 2008.

[21] A. Y. S. Lam, V. O. K. Li, and Z. Wei, "Chemical reaction optimization for the fuzzy rule learning problem," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[22] Y. Sun, A. Y. S. Lam, V. O. K. Li, J. Xu, and J. J. Q. Yu, "Chemical reaction optimization for the optimal power flow problem," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[23] J. J. Q. Yu, V. O. K. Li, and A. Y. S. Lam, "Sensor deployment for air pollution monitoring using public transportation system," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–7.

[24] J. J. Q. Yu, A. Y. S. Lam, and V. O. K. Li, "Real-coded chemical reaction optimization with different perturbation functions," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[25] J. J. Q. Yu, A. Y. S. Lam, and V. O. K. Li, "Evolutionary artificial neural network based on chemical reaction optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2011, pp. 2083–2090.

[26] B. Pan, A. Y. S. Lam, and V. O. K. Li, "Network coding optimization based on chemical reaction optimization," in *Proc. IEEE Global Commun. Conf.*, Dec. 2011, pp. 1–5.

[27] J. Xu, A. Y. S. Lam, and V. O. K. Li, "Stock portfolio selection using chemical reaction optimization," in *Proc. Int. Conf. Oper. Res. Financial Eng.*, Jun. 2011, pp. 458–463.

[28] A. Y. S. Lam, V. O. K. Li, and J. J. Q. Yu, "Power-controlled cognitive radio spectrum allocation with chemical reaction optimization," *IEEE Trans. Wireless Commun.*, submitted for publication.

[29] J. J. Q. Yu, A. Y. S. Lam, and V. O. K. Li, "An adaptive scheme for chemical reaction optimization," *IEEE Trans. Evol. Comput.*, submitted for publication.

[30] *Chemical Reaction Optimization* (2012, Apr. 19) [Online]. Available: http://cro.eee.hku.hk

[31] A. Y. S. Lam and V. O. K. Li, "Nature of search algorithms," Dept. EEE, Univ. Hong Kong, Pokfulam, Hong Kong, Tech. Rep. TR-2009-001, Jan. 2009.

[32] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*. Bristol, U.K.: IOP Publishing, 1997.

[33] H. Huang, Z. Y. Lin, Z. F. Hao, Y. S. Zhang, and X. Q. Li, "Convergence analysis and comparison of evolutionary algorithms based on relation model," *Chin. J. Comput.*, vol. 34, no. 5, pp. 801–811, May 2011.

[34] E. Boros and P. L. Hammer, "Pseudo-Boolean optimization," *J. Discrete Appl. Math.*, vol. 123, nos. 1–3, pp. 155–225, Nov. 2002.

[35] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

[36] A. Linhares, "Synthesizing a predatory search strategy for VLSI layouts," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 147–152, Jul. 1999.

[37] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 4, pp. 985–997, May 2010.

[38] J. Xu, A. Y. S. Lam, V. O. K. Li, Q. Li, and X. Fan, "Short adjacent repeat identification based on chemical reaction optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.

[39] M. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[40] W. Feller, *An Introduction to Probability Theory and Its Applications,* vol. 1. New York: Wiley, 1968.

Albert Y.S. Lam (S'03–M'10) received the B.Eng. degree (First Class Hons.) in information engineering and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2005 and 2010, respectively.

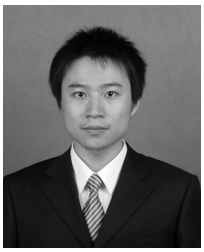
He is currently a Research Assistant Professor with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong. His current research interests include optimization theory and algorithms, evolutionary computation, smart grids, wireless and mobile networking, and Internet protocols and applications.

Dr. Lam is a Croucher Research Fellow. He is the founding Chair of the Social Media Subcommittee of the IEEE Computational Intelligence Society (CIS).

Victor O.K. Li (S'80–M'81–SM'86–F'92) received the SB, SM, EE, and ScD degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1977, 1979, 1980, and 1981, respectively.

He is currently a Chair Professor of information engineering and the Head of the Department of Electrical and Electronic Engineering, University of Hong Kong (HKU), Pokfulam, Hong Kong. His current research interests include optimization techniques, and core technologies and applications of information and communication technologies.

Dr. Li has received numerous awards, including the PRC Ministry of Education Changjiang Chair Professorship at Tsinghua University, the U.K. Royal Academy of Engineering Senior Visiting Fellowship in Communications, the Croucher Foundation Senior Research Fellowship, and the Order of the Bronze Bauhinia Star, Government of the Hong Kong Special Administrative Region, China. He is the Registered Professional Engineer and a Fellow of the IAE and the HKIE.



Jin Xu received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2008 and the Ph.D. degree in electrical and electronic engineering from the University of Hong Kong, Pokfulam, Hong Kong, in 2012.

His current research interests include evolutionary algorithms and their applications, parallel and distributed computing, bioinformatics, and portfolio management.