On the Use of Elitism to Improve Convergence of the Chemical Reaction Optimization Algorithm in Discrete Optimization Problems

Sinan Salman
College of Technological Innovation
Zayed University
Abu Dhabi, UAE
sinan.salman@zu.ac.ae

Abstract

This paper investigates the use of elitism in chemical reaction optimization (CRO) to address its convergence performance in difficult problems. We focus on problems with complex and highly discontinuous solution space. In such problems CRO's convergence performance tend to be sluggish as the algorithm repeatedly digress from the best-found solution characteristics in search for solutions in different areas in the problem's solution space. A complex road network design problem is used to demonstrate this issue and experiment with the impact of elitism on algorithm convergence. Elitism has been used successfully in evolutionary algorithms. Results show that its use in CRO improves algorithm convergence performance drastically. However, due to CRO's tendency to have diminishing population of molecules in such problems, the use of a larger list of elite solutions appears to be ineffective in improving the algorithm performance beyond the initial gains from introducing elitism. We investigate the reasons behind this observation and point to possible solutions.

Keywords
Discrete Optimization, Chemical Reaction Optimization, Elitism, Traffic Network Design

1. Introduction

Chemical Reaction Optimization (CRO) is a relatively new addition to the metaheuristics optimization literature. It was proposed by Lam and Li (2010). While many metaheuristics take inspiration from the behavior of biological systems or organisms (e.g. Genetic Algorithm, Swarm Optimization, Ant Colney Optimization), CRO, on the other hand, is inspired by chemical reactions in the physical world. Chemical reactions allow molecules to attain their optimal stable energy level. The algorithm mimics reactant molecules taking part in reactions to produce lower energy molecules, known as products. In the algorithm, candidate solutions are represented by molecules and neighborhood search operators by reactions.

This paper investigates the CRO algorithm performance in difficult optimization problems. It focuses on discrete optimization problems where small changes in the solution characteristics typically result in drastic changes in the objective function. Such problems are referred to here as high-discontinuity optimization problems. They present a unique optimization challenge as metaheuristics often struggle to achieve convergence on such problems. This is due to the complex and highly discontinuous problem solution space. Metaheuristics employ neighborhood search operators (e.g., mutation and crossover in Genetic Algorithms) to search the problem's solution space. However, in such difficult solution space terrain, neighborhood search operators tend to digress from best solutions found in previous iterations or stall in sub-optimal areas.

One solution that has been suggested in the literature to help algorithm convergence is the use of Elitism. It is the concept of maintaining a list of elite solutions and using members of this list in the search procedure in hope of utilizing their characteristics in finding better solutions. This approach has been applied to various metaheuristics in the optimization literature (Guo et al. 2014; Laumanns, Zitzler, and Thiele 2000; Shengxiang Yang 2008). This paper investigates the use of elitism in CRO to achieve better optimization convergence performance on the highly discontinuous network design problem (NDP). Examples of CRO applications in NDP literature can be found in

(Szeto, Wang, and Wong 2014; Wang and Szeto 2017), and an example implementation of elitism in CRO for a computer vision application can be found in (Duan and Gan 2015).

CRO applications in the literature are growing both in number and variety. This is due to its success in adapting to and handling a range of different optimization problems. In discrete optimization, CRO applications cover a wide variety of application areas such as 0-1 knapsack optimization (Truong, Li, and Xu 2013), Quadratic assignment problem (Xu, Lam, and Li 2010), flow-shop scheduling (Bargaoui, Belkahla Driss, and Ghédira 2017), Network coding (Bo Pan, Lam, and Li 2011), and wireless sensor networks clustering and routing (Srinivasa Rao and Banka 2017), to mention a few. CRO was also extended to handle multi-objective optimization problems (Bechikh, Chaabani, and Ben Said 2015; Wang and Szeto 2017). While it was originally proposed for solving discrete optimization problems, its applications now include continuous optimization problems as well (Lam, Li, and Yu 2012; Yu, Lam, and Li 2015). Both Islam et al. (2019) and Nayak et al. (2019) provide a review of CRO applications and list its different variants. For a comprehensive tutorial on CRO, readers are referred to (Lam and Li 2012).

The rest of the paper is organized as follows. Section 2 provides a brief overview of CRO. Section 3 summarizes the NDP used to demonstrate the optimization performance of elitist CRO (ECRO) in a highly discontinuous problem. Section 4 describes the algorithm modifications needed to introduce elitism in CRO. Section 5 presents the numerical experiments and discuss the paper findings. Finally, we conclude and point to future work in Section 6.

2. Overview of CRO

This paper builds on the conical CRO algorithm proposed by (Lam and Li 2010). CRO includes four reaction types, these reactions function as search operators for the metaheuristic. The reactions include unimolecular and inter-molecular reactions memicing collisions in chemical reactions. The reactions achieve results analogous to those of crossover and mutation in Genetic Algorithms. The four reactions types are: on-wall ineffective collision, inter-molecular ineffective collision, decomposition, and synthesis. The following describes our implementation of the four reaction types for solving the NDP.

Both decomposition and on-wall ineffective collision are unimolecular reactions. A decomposition reaction decomposes a single molecule into two by splitting its solution encoding into two to create two product molecules. The remaining encoding bits are set with random values from the domain of possible bit values. Decomposition is considered a diversification agent in CRO as it helps discover new areas of the solution space.

An on-wall ineffective collision subtly modifies the solution by randomly selecting a single bit from its solution encoding and replacing its value with a randomly selected value from the domain of possible bit values. On-wall ineffective collision is the weakest of the four reactions and is used to search the immediate neighborhood of solution candidates. It can be compared to the mutation operator in Genetic Algorithms.

Synthesis and inter-molecular ineffective collision are both inter-molecular reactions. A synthesis reaction combines two reactant molecules into one. It does so by combining two portions from the reactant molecules' bits into a new product molecule. The point of splitting in the bits sequence is randomly selected. Synthesis is considered a concentration agent in CRO as it helps concentrate solution characteristics.

Finally, an inter-molecular ineffective collision works by swapping bits between molecules. A bit position is selected at random and bits after the position are swapped between the two molecules. Inter-molecular ineffective collision can be compared to the crossover operator in Genetic Algorithms.

Figure 1 illustrates the four reactions and how they operate on solution encodings. The solution encoding design is described in the following section. At this point it suffices to say that each bit describes a unique solution characteristic and its domain of values is the set {0, 1, 2}.

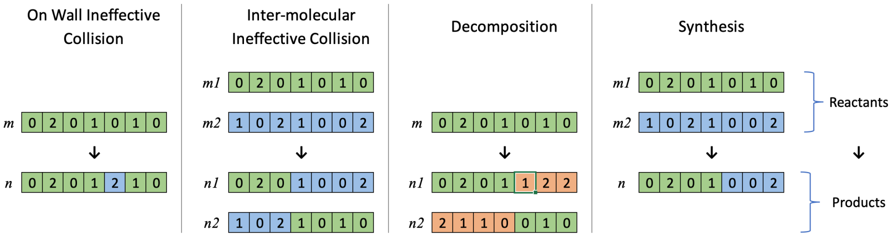

Figure 1. Illustration of CRO's four reactions. In the reactions, m, m1, and m2 are reactant molecules, and n, n1, and n2 are product molecules.

Conservation of energy is a core feature in CRO. Energy cannot be created nor destroyed in a chemical reaction, rather it is converted between molecules' potential and kinetic energies (*PE* and *KE*). In CRO, the solution's *PE* represents its objective function value while *KE* is used to allow it to escape local optima. Each reaction is governed by energy laws, and only when its input energy is greater than its output energy (i.e. $E_{Reaction} \geq 0$) the reaction is allowed to take place. This mechanism allows the heuristic to search for solutions with lower *PE* while still being able to explore the problem's solution space through the creation of occasionally higher *PE* solutions.

CRO also includes a central energy buffer, where excess energy from exothermic reactions is stored, and later used to enable endothermic reactions that may not otherwise be possible due to their high energy requirement. In CRO decomposition reactions are endothermic and on-wall ineffective collision reactions are exothermic. The remaining two reaction types are selfsufficient energy-wise. Equation 1 governs all reactions, with the exception of decomposition, which is allowed to use a limited amount of energy from the buffer, as shown in Equation 2. In Equation 2, δ_1 and δ_2 are polled from uniform random distribution with the range [0,1).

$$E_{Reaction} = \sum PE_{Reactants} + \sum KE_{Reactants} - \sum PE_{Products} \tag{1}$$

$$E_{Reaction} = \sum PE_{Reactants} + \sum KE_{Reactants} - \sum PE_{Products} + (\delta_1 \delta_2)\ Buffer \tag{2}$$

Algorithm 1 describes the main loop used in CRO and Table 1 (Section 5.1) lists the algorithm's parameters. For a more in-depth description of the algorithm logic and parameters the reader is referred to (Lam and Li 2012).

Algorithm 1: CRO main loop

```
Input: Parameters, InitialMolecules
Calculate PE for InitialMolecules
Set KE ← InitialKE for InitialMolecules
Set Buffer ← InitialBuffer
Set nEvaluations ← 0
Do while nEvaluations < MaxEvaluations
    if random(0,1) > MolColl or PopulationSize = 1
        Select a solution randomly
        if nHits > Alpha
            Perform Decomposition
        else:
            Perform OnWallIneffectiveCollision
    else:
        Select two solutions randomly
        if both molecules' KE < Beta
            Perform Synthesis
        else:
            Perform IntermolecularIneffectiveCollision
    nEvaluations ← nEvaluations + 1
```

3. A High-Discontinuity Optimization Problem

This section provides a summary of the NDP utilized to demonstrate the challenge posed by a complex and highly discontinuous solution space problem. This work uses the NDP proposed in (Salman and Alaswad 2018). The model optimizes traffic pattern design for a road network to reduce its congestion. Traffic direction conversion is used to modify the network, where roads can be converted from two-direction flow to one-direction flow. The challenging nature of the problem's solution space becomes evident when one considers the impact one or few road direction changes could have on the network as a whole. For instance, it is possible to convert a good network design to an extremely congested one or an invalid network design (disconnected or absorbing) with just one or few road conversions. This characteristic makes the investigated NDP a good candidate for demonstrating CRO performance in such problems. Its objective function is discrete and non-monotone with a very large solution space characterized by high degree of discontinuity.

The mathematical model is summarized in equations 3, 4 and 5 below. In the model D_{max} is the network's maximum road traffic density and is obtained from the inequality 4. V is the estimated total number of vehicles traversing the network at the time of data collection. π_i is the stationary distribution probability for road i, and is obtained from equation 5 (via Markov chain steady state theory). In Equation 5, $\boldsymbol{\pi}$ is the steady state probability vector, whose elements are π_i. L_i is road i length, and N_i is its number of lanes. P is the transition probability matrix. Different network designs can be implemented by modifying the P matrix based on the decision to convert roads or keep them flowing in two-directions. A detailed description of the NDP model is beyond the scope of this paper. Interested readers are referred to the original paper for a thorough discussion of the model. What we will focus on here is the solution representation and solution methodology.

$$\text{Minimize } D_{max} \tag{3}$$

Subject to:

$$\frac{V\,\pi_i}{L_i\,N_i} \leq D_{max} \tag{4}$$

$$\boldsymbol{\pi}P = \boldsymbol{\pi} \tag{5}$$

To represent solutions of this NDP, a solution encoding scheme is utilized where opposing direction roads are paired such that network changes are expressed as modifications to traffic direction of road pairs. That is, each bit in a solution encoding can take a value from the set {0, 1, 2}, indicating no change, reversal of the first road direction, or reversal of the second road direction, respectively. This scheme reduces optimization search space considerably compared to a scheme where each road gets its own binary bit. The use of this scheme to represent a simple 14-links road network is illustrated in Figure 1. The rest of the paper focuses on the solution methodology which utilizes the NDP model mentioned here for evaluating the objective function values of CRO generated network designs.

4. The Use of Elitism in CRO

As mentioned in section 2, this work builds on the conical CRO. It does so without introducing significant changes other than adapting the algorithm's four reaction types to the NDP at hand. Nonetheless, elitism is introduced into the algorithm to investigate its impact on convergence. Elitism is used frequently in evolutionary optimization to keep best found solutions in the population in hope subsequent solutions will borrow and benefit from their characteristics.

We introduce elitism in CRO via catalysts. The algorithm maintains a fixed-size ordered list of best solutions found in the optimization at any time, this list is referred to as catalysts. As new best solutions are identified they are inducted into the list replacing the lowest ranked ones.

When a reaction succeeds in meeting its energy law requirement and its products are about to replace the reactants in the population, the elitisim mechanisim ensures that no catalyst molecules are discarded. If the reactants include catalyst molecules, instead of allowing product molecules to replace them, the product molecules replace the lowest ranked solution in the population. This allows both product and catalyst reactant molecules to coexist in the population post-reaction. The discarded low ranked molecules are considered unwanted reaction byproducts.

Algorithms 2 and 3 detail the catalysts implementation in CRO. In Algorithm 2, *nCatalysts* represents the size of the catalysts list. Note that the logic presented in Algorithm 3 is a simplification of the logic required in an elitist CRO. For presentation purposes, Algorithm 3 does not address cases when multiple reactants or multiple products are involved. Nevertheless, this can be addressed via a straightforward extension of the presented logic.

Algorithm 2: Managing the Catalysts list

```
Input: Catalysts, nCatalysts, NewMolecule
if sizeof(Catalysts) < nCatalysts
    add NewMolecule to Catalysts
    Catalysts ← sort(Catalysts)
else
    l ← lowest rank Catalyst
    if NewMolecule.PE < l.PE
        replace l with NewMolecule in Catalysts
        Catalysts ← sort(Catalysts)
```

Algorithm 3: Elitism logic in CRO

```
Input: Catalysts, ReactantMolecule, ProductMolecule, Population
if ReactantMolecule not in Catalysts
    remove ReactantMolecule from population
else
    l ← lowest rank Molecule in population (highest PE)
    remove l from population
add ProductMolecule to population
```

5. Numerical Experiments

5.1 Scenario and Parameter Tuning

This analysis utilizes the same road network data and assumption described in (Salman and Alaswad 2018) to demonstrate CRO's performance in solving the NDP. Figure 2 shows results of the parameter tuning process for the CRO algorithm. In the plots, *PE* values are reported as fractions of the observed D_{max} value of the unmodified road network. The tuning procedure sequentially tunes each parameter over its range. Each parameter is tuned using average *PE* values calculated from three identical optimization runs (replications), each capped at 2,000 evaluations. The parameter value resulting in the best average *PE* (lowest) is selected before moving to the next parameter. Table 1 shows the parameters and their tuned values. Due to the high *PE* of the randomly initialized starting solutions, parameters *InitialKE* and *InitialBuffer* are set to zero as they have little impact on CRO's performance in the investigated NDP instance.

Table 1. CRO Parameters

Parameter	Value	Description
PopSize	50	Initial number of molecules at the start of the optimization.
MolColl	0.3	Inter-molecular collision probability.
KELossRate	0.5	*KE* loss rate lower limit; when performing an on-wall ineffective collision, the product molecule keeps a portion of the reaction energy determined by Uniform(*KELossRate*,1).
Alpha	25	Decomposition criterion; the number of unsuccessful reactions attempts (*nHits*) before a molecule is decomposed to stop local search and diversify.
Beta	0.5	Synthesis criterion; perform synthesis if all involved molecules have $KE \leq Beta$.

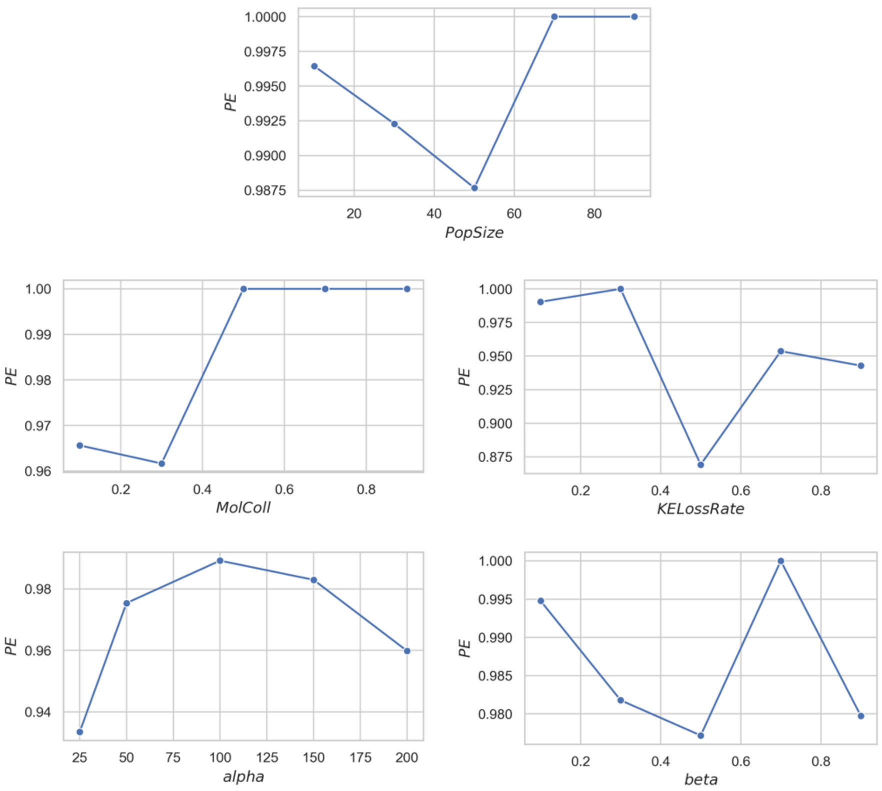

Figure 2. ECRO Parameters Calibration

5.2 Analysis of Results

With the algorithm and scenario setup, we now experiment with different sizes of the catalysts list (*nCatalysts*). For that purpose, the following list sizes are chosen: 0, 2, 4, 8, 16, 24, and 32 molecules. Each experiment is executed 5 times and the averages of these replications are reported here. Figure 3 shows the average CRO convergence performance for each of the *nCatalysts* values as a function of number of evaluations executed. The figure shows a drastic performance difference between running the CRO with and without catalysts. This confirms the observations previously discussed that complex and highly discontinuous objective functions present a challenging solution space for the conical CRO algorithm.

Figure 4 shows the impact of catalysts list size on the CRO algorithm performance. Just by introducing catalysts into the algorithm its performance improves from a best solution *PE* average of 0.967 for the conical CRO to a range of 0.694-0.733 for ECRO. The results show that the size of the catalyst list is less important compared to having elitism logic used in CRO.

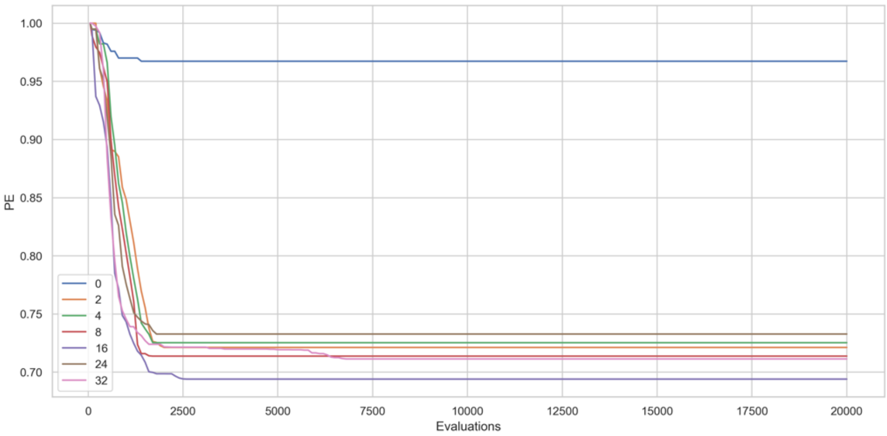

Figure 3. Effect of number of catalysts on CRO convergence

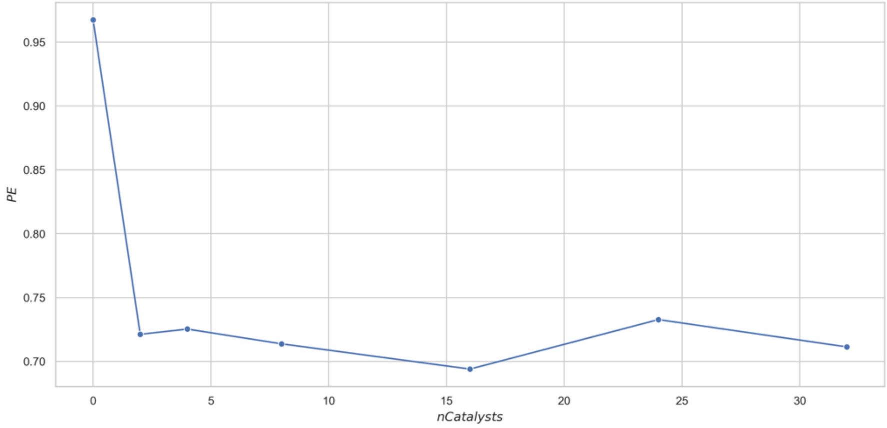

Figure 4. Effect of number of catalysts on CRO's final best solution

Figure 5 provides a closer look at the algorithm mechanics. For this illustration a *nCatalysts* value of 2 is chosen. The colored areas show the average number of population molecules by type, depending on the reaction that produced the molecules. Analyzing these results highlights two observations. The immediate observation is that the total number of molecules in the population quickly diminishes to an average of around 5 molecules in the first 2000 evaluations. It then continues to diminish at a slower rate in the next 10,000 evaluations to an average of around 2 molecules. This observation explains the weak differentiation between results of the *nCatalysts* values in Figures 3 and 4. That is, increasing the size of the catalysts list is ineffective when the number of the molecules in the population is less than *nCatalysts*. The same diminishing number of molecules behavior is notices in all experiments with different *nCatalysts* values. The second observation is that in general inter-molecular collisions and on-wall ineffective collisions are the dominant reactions in the algorithm's run history.

To understand the previous observations, we further detail the optimization history and track each reaction cumulative success rate, as seen in Figure 6. The plot shows that in the first 2000 evaluations synthesis is much more effective in finding good solutions than decomposition, and it is that imbalance that results in the diminishing of the total number of molecules in the population. When the number of molecules reduce to 1 synthesis is no longer executed, increasing the chances for inter-molecular reactions. Similarly, decomposition is less likely to occur than the other reactions as it requires a molecule to go through a number of reactions (*Alpha*) without success before it is decomposed. This also increases the chances for on wall ineffective collisions.

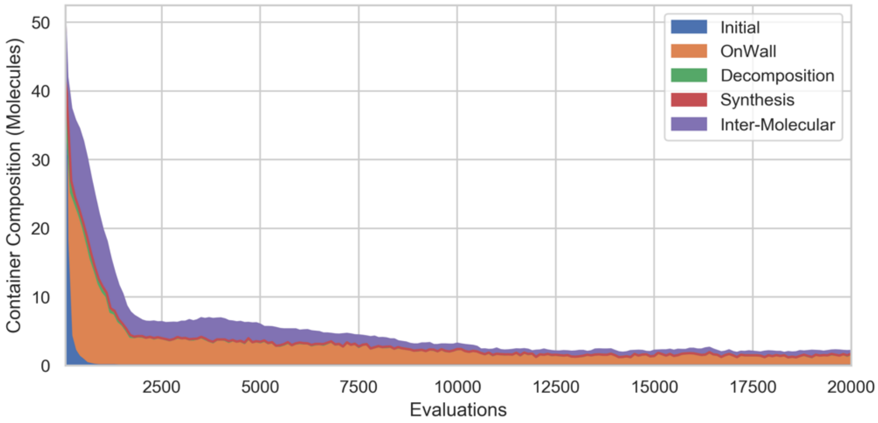

Figure 5. ECRO container composition

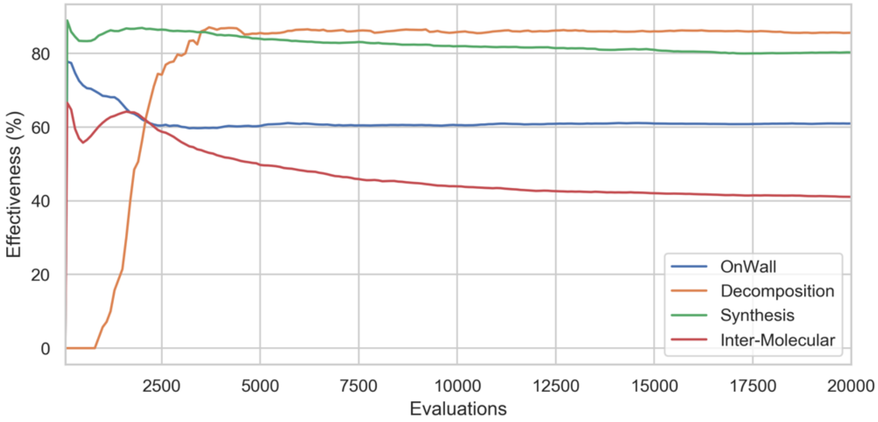

Figure 6. Cumulative reaction effectiveness

6. Conclusion

This paper investigates the effect of elitism on CRO when optimizing a discrete objective function with a complex and highly discontinuous problem solution space. Elitism is incorporated into CRO via designating molecules with best *PE* as catalyst molecules. Results show that just by introducing elitism into the CRO algorithm its convergence performance improves drastically (26% improvement). However, CRO convergence performance appears to be unaffected by the number of catalysts used in the algorithm. While surprising at first, this observation is explained by the fast diminishing number of molecules in the population. When the number of molecules is less than the number of catalysts, the impact of the latter becomes mute. Further analysis of the CRO algorithm points to an imbalance in the deployment of the different reactions' types used in CRO in this NDP, which leads to the diminishing number of population molecules.

Future directions for this work could include investigating population size preserving or adaptive CRO variants to help improve the algorithm's performance further. The impact of the catalysts list size could be different under such algorithm variants, requiring a similar reevaluation to the one presented here.

References

Bargaoui, Hafewa, Olfa Belkahla Driss, and Khaled Ghédira. 2017. "A Novel Chemical Reaction Optimization for the Distributed Permutation Flowshop Scheduling Problem with Makespan Criterion." *Computers & Industrial Engineering* 111 (September): 239–50. https://doi.org/10.1016/j.cie.2017.07.020.

Bechikh, Slim, Abir Chaabani, and Lamjed Ben Said. 2015. "An Efficient Chemical Reaction Optimization Algorithm for Multiobjective Optimization." *IEEE Transactions on Cybernetics* 45 (10): 2051–64. https://doi.org/10.1109/TCYB.2014.2363878.

Bo Pan, A. Y. S. Lam, and V. O. K. Li. 2011. "Network Coding Optimization Based on Chemical Reaction Optimization." In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 1–5. Houston, TX, USA: IEEE. https://doi.org/10/fzz86n.

Duan, Haibin, and Lu Gan. 2015. "Elitist Chemical Reaction Optimization for Contour-Based Target Recognition in Aerial Images." *IEEE Transactions on Geoscience and Remote Sensing* 53 (5): 2845–59. https://doi.org/10.1109/TGRS.2014.2365749.

Guo, Weian, Wuzhao Li, Qun Zhang, Lei Wang, Qidi Wu, and Hongliang Ren. 2014. "Biogeography-Based Particle Swarm Optimization with Fuzzy Elitism and Its Applications to Constrained Engineering Problems." *Engineering Optimization* 46 (11): 1465–84. https://doi.org/10.1080/0305215X.2013.854349.

Islam, Md. Rafiqul, C. M. Khaled Saifullah, and Md. Riaz Mahmud. 2019. "Chemical Reaction Optimization: Survey on Variants." *Evolutionary Intelligence* 12 (3): 395–420. https://doi.org/10/ggbtz2.

Lam, Albert Y S, and Victor O K Li. 2010. "Chemical-Reaction-Inspired Metaheuristic for Optimization." *IEEE Transactions on Evolutionary Computation* 14 (3): 381–99. https://doi.org/10.1109/TEVC.2009.2033580.

Lam, Albert Y. S., and Victor O. K. Li. 2012. "Chemical Reaction Optimization: A Tutorial: (Invited Paper)." *Memetic Computing* 4 (1): 3–17. https://doi.org/10/ggbmwk.

Lam, Albert Y. S., Victor O. K. Li, and James J. Q. Yu. 2012. "Real-Coded Chemical Reaction Optimization." *IEEE Transactions on Evolutionary Computation* 16 (3): 339–53. https://doi.org/10/fm8zqt.

Laumanns, Marco, Eckart Zitzler, and Lothar Thiele. 2000. "A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism." In *In Congress on Evolutionary Computation (CEC 2000*, 46–53. IEEE Press.

Nayak, Janmenjoy, Sura Paparao, Bighnaraj Naik, N. Seetayya, P. Pradeep, H. S. Behera, and Danilo Pelusi. 2019. "Chemical Reaction Optimization: A Survey with Application and Challenges." In *Soft Computing in Data Analytics*, edited by Janmenjoy Nayak, Ajith Abraham, B. Murali Krishna, G. T. Chandra Sekhar, and Asit Kumar Das, 758:507–24. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-13-0514-6_50.

Salman, Sinan, and Suzan Alaswad. 2018. "Alleviating Road Network Congestion: Traffic Pattern Optimization Using Markov Chain Traffic Assignment." *Computers & Operations Research* 99 (November): 191–205. https://doi.org/10.1016/j.cor.2018.06.015.

Shengxiang Yang. 2008. "Genetic Algorithms with Memory- and Elitism-Based Immigrants in Dynamic Environments." *Evolutionary Computation* 16 (3): 385–416. https://doi.org/10.1162/evco.2008.16.3.385.

Srinivasa Rao, P. C., and Haider Banka. 2017. "Novel Chemical Reaction Optimization Based Unequal Clustering and Routing Algorithms for Wireless Sensor Networks." *Wireless Networks* 23 (3): 759–78. https://doi.org/10.1007/s11276-015-1148-0.

Szeto, W. Y., Yi Wang, and S. C. Wong. 2014. "The Chemical Reaction Optimization Approach to Solving the Environmentally Sustainable Network Design Problem." *Computer-Aided Civil and Infrastructure Engineering* 29 (2): 140–58. https://doi.org/10.1111/mice.12033.

Truong, Tung Khac, Kenli Li, and Yuming Xu. 2013. "Chemical Reaction Optimization with Greedy Strategy for the 0–1 Knapsack Problem." *Applied Soft Computing* 13 (4): 1774–80. https://doi.org/10.1016/j.asoc.2012.11.048.

Wang, Yi, and W. Y. Szeto. 2017. "Multiobjective Environmentally Sustainable Road Network Design Using Pareto Optimization." *Computer-Aided Civil and Infrastructure Engineering* 32 (11): 964–87.

Xu, Jin, Albert Y S Lam, and Victor O K Li. 2010. "Parallel Chemical Reaction Optimization for the Quadratic Assignment Problem," 8.

Yu, James J. Q., Albert Y. S. Lam, and Victor O. K. Li. 2015. "Adaptive Chemical Reaction Optimization for Global Numerical Optimization." In *2015 IEEE Congress on Evolutionary Computation (CEC)*, 3192–99. Sendai, Japan: IEEE. https://doi.org/10/ggbmwq.

Biography / Biographies

Sinan Salman is an assistant professor of Information Systems and Technology Management at Zayed University. Dr. Salman received his PhD and MS in Industrial Engineering from University of Arkansas and Oklahoma State University, respectively, and BSc in Mechanical Engineering from Jordan University of Science and Technology. Prior to joining Zayed University, he worked in the US in the field of Supply Chain Management for 13 years. His professional career included logistics consulting, supply chain and transportation management, and optimization software development. His areas of interest include operations research and its applications in supply chain, logistics, and healthcare systems.