AutoDock Koto: A Gradient Boosting Differential Evolution for Molecular Docking

Junkai Ji, Jin Zhou, Zhangfan Yang, Qiuzhen Lin, *Member, IEEE*, Jianqiang Li, *Member, IEEE*, and Carlos A. Coello Coello, *Fellow, IEEE*

***Abstract*—Molecular docking plays a vital role in modern drug discovery, by supporting predictions of the binding modes and affinities of ligands at the binding site of target proteins. Several docking programs have been developed for both commercial and academic applications. Typically, a docking program's performance depends on the sampling algorithm used to generate the ligand's potential conformations and the scoring function applied to evaluate and rank these conformations. Evolutionary algorithms are widely used as sampling algorithms in docking programs. However, both the linkage problem and the dimensionality degenerate the search ability of evolutionary algorithms in the docking process. Therefore, a newly designed docking program named AutoDock Koto was developed in this study, which adopts a novel gradient boosting differential evolution algorithm to effectively address these issues. Experimental results show that compared with commonly used docking programs, AutoDock Koto yields dramatic improvements in docking performance based on an extensive dataset of 285 protein-ligand complexes. In addition, due to its strong docking ability, AutoDock Koto was used to identify potential drugs for COVID-19 based on a virtual screening of all approved drugs in our experiments. Sixteen drugs are found to possess low binding energy to the main target protease of SARS-CoV-2, and thus have the potential to treat COVID-19 as antiviral drugs. The source code of AutoDock Koto can be downloaded for free from https://github.com/codezhouj/Molecular_Docking.**

***Index Terms*—Drug Design; Molecular Docking; Differential Evolution; Virtual Screening; SARS-CoV-2.**

I. Introduction

Computer-aided drug design (CADD) techniques have become reliable and essential tools in modern drug design. The discovery and development of novel drugs are typically expensive and time-consuming, while CADD can significantly accelerate the screening pace and drastically improve hit discovery rates [1]. In fact, most pharmaceutical companies operate CADD departments, and many commercial drugs have been developed through the use of CADD, such as saquinavir, indinavir, captopril and tirofiban [2], [3].

In modern CADD, molecular docking is one of the most commonly used computational technologies to assess interactions between small-molecule ligands and macromolecular target proteins. As a structure-based drug design strategy, computational docking relies on knowledge of the three-dimensional (3D) structures of target proteins, which are mainly obtained from X-ray crystallography, NMR spectroscopy and homology modeling approaches [4]. However, the main obstacle of protein-ligand docking is that the 3D protein structures are unavailable or the reported structures are sometimes unreliable. Fortunately, an emerging machine learning approach called AlphaFold2 provides a very reliable way to produce 3D protein structures [5]. This greatly accelerates docking programs and improves the success rate of computational virtual high-throughput screening. In the past two decades, a large number of commercial and academic docking programs have been developed for pharmaceutical research, such as Glide [6], GOLD [7], DOCK [8], Surflex [9], AutoDock [10] and AutoDock Vina [11]. Due to their open source nature and ease of use, AutoDock Vina and DOCK are the most popular docking programs and have relatively high citation rates in the literature. Although docking programs are originally designed to investigate molecular recognition between proteins and ligands, they are also widely used in other tasks of drug discovery, including drug repositioning, reverse screening, multitarget ligand design and polypharmacology [12], [13].

In general, the success of a docking program is dependent on two essential components: an effective sampling algorithm and an accurate scoring function [14]. The sampling algorithm generates putative ligand conformations within the appropriate target binding site of a protein. The sampling algorithms can be divided into three main classes: shape-matching, systematic search and stochastic search algorithms [15]. The scoring function guides the sampling and ranks the generated conformations by estimating the binding affinity between the protein and ligand. Different scoring functions are based on different theoretical assumptions and follow diverse model construction approaches, which can be classified into four main categories: force-field-based, empirical, knowledge-based and machine-learning-based scoring functions [16]. Although docking programs have undergone continual method development for decades, the accuracy and effectiveness of

This work is supported in part by the National Key R&D Program of China under Grant 2020YFA0908700, in part by the National Natural Science Foundation of China (NSFC) under Grants 62106151, 61876110 and 62073225, in part by a Project of the Guangdong Basic and Applied Basic Research Fund (No. 2019A1515111139), in part by the Natural Science Foundation of Guangdong Province-Outstanding Youth Program under Grant 2019B151502018, and in part by Shenzhen Science and Technology Program Under Grant JCYJ20220531101411027. Carlos A. Coello Coello gratefully acknowledges support from CONACyT grant no. 2016-01-1920 (Investigacion en Fronteras de la Ciencia 2016). (Corresponding author: Qiuzhen Lin).

Junkai Ji and Jianqiang Li are with the National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China. (e-mail of Ji: jijunkai@szu.edu.cn).

Jin Zhou, Zhangfan Yang, and Qiuzhen Lin are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.(e-mail of Lin: qiuzhlin@szu.edu.cn)

Carlos A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), México, D.F. 07300, MÉXICO. He is also a Faculty of Excellence with School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey, N.L., Mexico.

their applications have not always been ensured, and related computational approaches for drug discovery are still in their infancy [17]. Since scoring functions are specifically designed for different requirements from users, it is difficult to determine which scoring function is suitable for particular target proteins [18]. Thus, more attention is given to sampling algorithms to improve the docking performance in this study.

Being stochastic algorithms, evolutionary algorithms (EAs) are widely used as sampling algorithms in docking programs. For instance, a genetic algorithm (GA) was adopted to search complex conformations in the GOLD program [7]. A modified particle swarm optimization (PSO) algorithm enhanced with an efficient local search strategy was used in SODOCK [19]. In a further paper, a Lamarckian genetic algorithm was employed in AutoDock, which combined a traditional GA with a simulated annealing algorithm [10]. PSOVina and GWOVina, two variants of Autodock Vina, utilize a chaos-embedded PSO and the so-called Gray wolf optimizer as the sampling algorithms, respectively [20], [21]. In [22], a differential evolution (DE) algorithm was found to achieve better docking performance than a GA and PSO in the framework of AutoDock. More recently, BRKGA-DOCK used a biased random keys GA and MSLDOCK employed a random drift PSO combined with a Solis and Wets local search method as their search strategies in [23] and [24], respectively. In addition to single-objective EAs, multi-objective evolutionary algorithms (MOEAs) have also been used to address molecular docking optimization problems. EADock first adopted an MOEA in a docking program, which formulates the total binding affinity and the solvation energy as two fitness functions [25]. In [26], a multi-objective particle swarm optimizer (MOPSO) was employed to optimize the intermolecular and intramolecular binding affinity simultaneously. Similarly, the NSGA-II was adopted to optimize the bonding and non-bonding terms of binding energy which are considered to be conflicting in [27]. Moreover, two variations of MOPSO were adopted to improve the optimization results for a specific multi-objective docking problem, in which the intermolecular binding energy and the distances between the predicted ligand conformations and the co-crystallized structures were formulated as the two main objectives [28].

Although various EAs have been applied to molecular docking problems, no significant improvement in accuracy can be achieved by simple modifications and hybridizations of EAs [29], [30]. Through an in-depth investigation, we found that the docking performance of EAs is affected by two main issues. The first is the linkage problem. Strong interactions exist among variables in the decision space of a docking problem. The crossover operator destroys the interactions and then influences the fitness of the population during the optimization process. The second issue is the dimensionality of the problem. The dimensionality of docking problems is high when the chosen ligand has many rotatable bonds, and such a high-dimensional problem greatly increases the difficulty for EAs to produce a good approximation of the optimum. Therefore, in this study, a novel docking program named AutoDock Koto (also termed Koto) was developed to improve the performance in protein-ligand docking problems, in which a gradient boosting differential evolution (GBDE) approach is adopted as the sampling algorithm. Specifically, the GBDE algorithm uses the parameter adaptation strategy, originally proposed in the L-SHADE algorithm [31], to control the parameters of DE, and thus alleviate the fitness degeneration triggered by the linkage problem. Additionally, it simultaneously incorporates an adaptive gradient descent algorithm, termed adaptive moment estimation (Adam) [32], to enhance the optimization ability in high-dimensional docking problems. Compared with several state-of-the-art docking programs, a comprehensive evaluation is executed to verify the effectiveness of Koto in our experiments. Undoubtedly, AutoDock Koto can be regarded as a reliable and efficient computational tool for molecular docking, according to our experimental results.

In addition, Koto was employed to virtually screen antiviral drugs for COVID-19 in this study due to its powerful docking performance. The COVID-19 pandemic triggered by a novel SARS-CoV-2 pathogen had infected over 240 million people and had caused approximately one million deaths by 20 October 2021. Developing novel drugs is time-consuming and usually requires several years for clinical approval [33], while adopting docking programs to re-purpose approved pharmaceutical drugs provides an alternative treatment process to rapidly reduce the mortality and morbidity of infectious agents [14]. This is because approved drugs with well-established pharmacological and safety profiles can be used to directly validate their effectiveness against SARS-CoV-2. Actually, several Food and Drug Administration (FDA)-approved antiviral drugs have been studied for their anti-COVID-19 activities in clinical trials, including remdesivir, umifenovir, darunavir, nitazoxanide and favipiravir [34]. However, compared with experimental high-throughput screening, an *in silico* screening approach based on docking programs is a faster and lower-cost strategy that can serve as an initial filter to evaluate thousands of compounds. Our screening results show that sixteen approved drugs have low binding energy to the main target protease of SARS-CoV-2 and have the potential for treating COVID-19 as antiviral drugs. Most of them have not yet been reported in the literature and need further validation of the therapeutic efficacy in vitro and in large clinical trials.

The remainder of this paper is organized as follows: in Section II, the problem definition of molecular docking is presented. Section III provides two issues of EAs and our corresponding strategies to address them. Next, the program design of Koto is presented in Section IV. A docking performance analysis is provided in Section V, and a Koto-based virtual screening for COVID-19 is provided in Section VI. Finally, our conclusions and some possible directions for future work are presented in Section VII.

II. Preliminaries

The original concept of the ligand-protein docking mechanism was derived from the "lock-and-key" theory introduced by Fischer, in which the ligand is docked into the receptor like a key and lock [35]. Both the ligand and protein are treated as rigid bodies in this theory. Then, the "induced-fit" theory proposed by Koshland takes a step further and

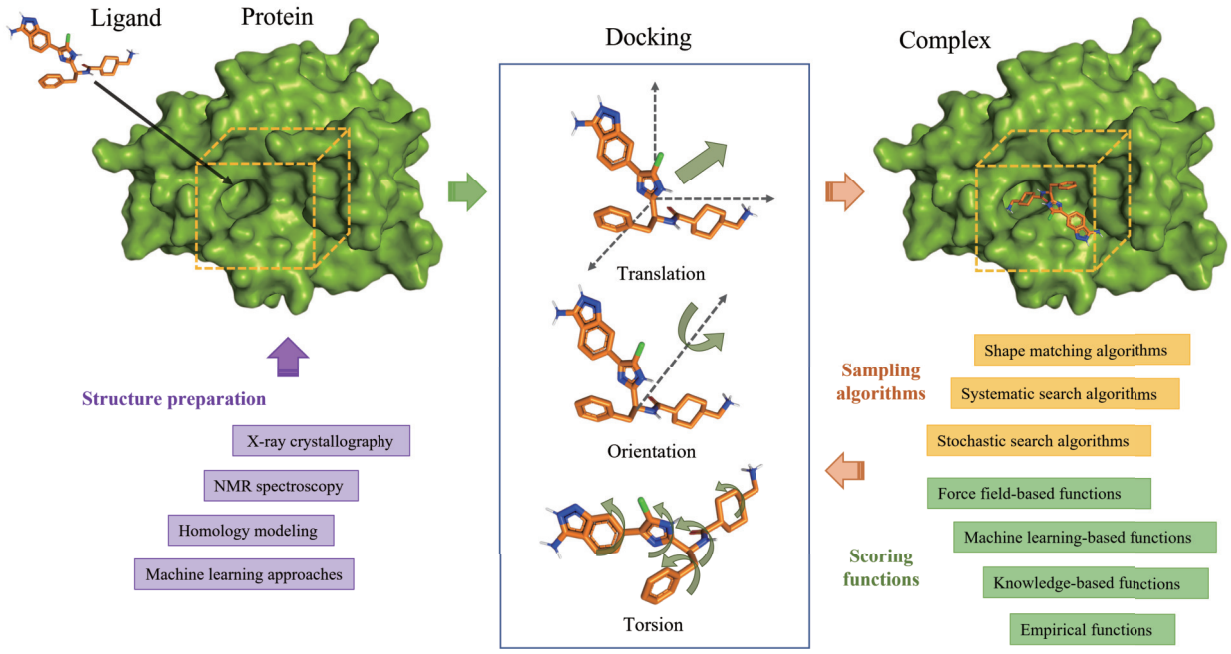


Fig. 1. Flowchart of docking a flexible ligand into a target protein.

suggests that the ligand and the active site of the protein are continually reshaped by the interaction between each other [36]. Treating both ligand and protein as flexible yields a more accurate prediction than that resulting from a rigid treatment. However, due to the limitations of computer resources and time consumption, the most popular method is to perform docking with a flexible ligand but a rigid receptor in practical applications [37], [38].

The main objective of docking is to optimize an ideal complex conformation that results in a minimum binding affinity between the ligand and the particular protein of interest. As presented in Fig. 1, the precondition of docking is that a target protein should be given, with a reliable 3D structure and a known binding site. When a ligand is docked into the protein, the structures of complex conformations are determined by the ligand's translation, orientations, and torsions. The former two correspondingly describe the position and rotation of the ligand as a rigid body, and the latter measures the rotatable bonds between the fragments in the ligand. Molecular docking is a typical non-deterministic polynomial-time (NP)-hard problem. Thus, it is too expensive to explore all the potential conformations. Only approximate solutions can be obtained by computational sampling algorithms. Scoring functions are used to score and rank the protein-ligand conformations generated by sampling algorithms, with an intertwined goal of accurately predicting the binding affinity and the correct binding mode of a complex. Usually, the conformation with the lowest predicted binding affinity is regarded as the binding mode for further biochemistry experiments and development. Moreover, designing sampling algorithms and scoring functions in docking programs must strike a balance between accuracy and speed, and the equipment requirement should be low, with the algorithms and functions working well on personal computers.

III. Motivation

The docking performance of EAs suffers from the linkage problem and of the curse of dimensionality. Further investigations concerning these two issues and our corresponding strategies to address them are presented in this section.

In biological systems, a pair of genes close on the same chromosome have a high probability of being inherited by the offspring together. Linkage represents the degree of association in the inheritance of the nonallelic genes. In EAs, linkage measures the interrelationships among the variables, and identifying these interactions is generally called linkage learning [39]. During the evolutionary process, the crossover operator generally destroys the linkages between pairs of variables and degenerates the fitness of individuals of the population. Consequently, linkage learning has been extensively investigated in the context of both discrete and continuous EAs [40]. These related studies are mainly based on the assumption that it is difficult to obtain *a priori* the linkage information in a specific problem [41], [42]. However, the linkage of variables is apparent in the decision space of docking problems when EAs are adopted as the sampling algorithms.

Specifically, the linkage problem of docking contains two components: the first is the interaction among the variables that correspond to the orientations of the ligand. In general, the spatial orientations and rotations of rigid bodies in 3D space can be described by a quaternion $\mathbf{q} = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, where a, b, c and d are real numbers, and $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are unit vectors pointing along the three Cartesian axes. Quaternions are widely used in the field of applied mathematics because they are compact and can avoid the problem of gimbal lock [43]. However, the quaternion is no longer a suitable choice when adopting EAs as the sampling methods in a docking program. The rotation of the ligand is determined by a unit

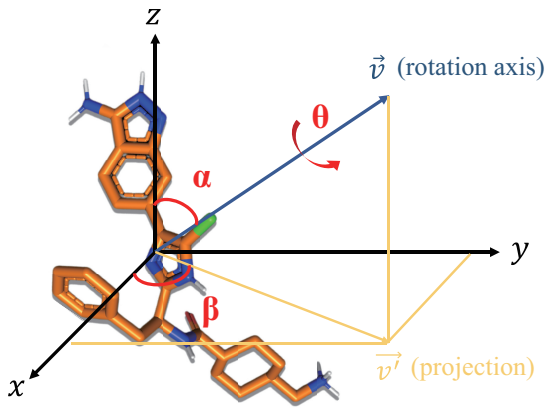


Fig. 2. Orientation of the ligand encoded by the three independent angles θ, α and β. Let the blue arrow $\vec{v}$ represent the rotation axis of a rigid ligand and the yellow arrow $\vec{v'}$ represent the projection of the rotation axis on the xy-plane; θ is defined as the rotation angle of the axis; α is the angle between $\vec{v}$ and the z-axis; and β is the angle between $\vec{v'}$ and the x-axis.

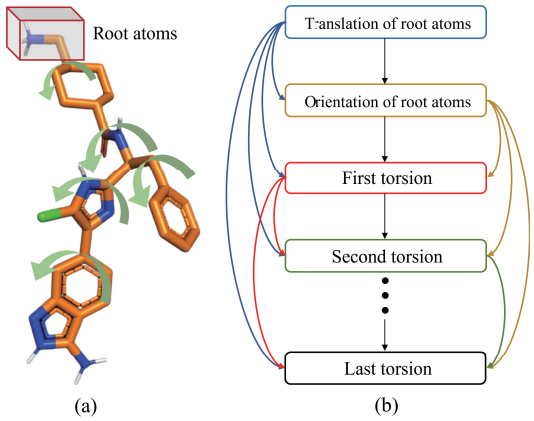


Fig. 3. Interactions among the variables of the translation, orientations and torsions of a flexible ligand. (a) For a flexible ligand, the atoms in the red rectangle are defined as the root atoms. The operations of translation and orientations are executed on the root atoms directly. Curved green arrows represent the torsions of rotatable bonds in a flexible ligand. (b) The relationships of the variables are presented in a hierarchical form. Curved arrows indicate that the upper variables can influence the lower ones.

quaternion $\mathbf{Uq} = \mathbf{q}/\|\mathbf{q}\|$, where $\|\mathbf{q}\|$ denotes the norm of the quaternion. This means that the four real-valued variables of $\mathbf{Uq}$ must obey one constraint, namely, $a^2 + b^2 + c^2 + d^2 = 1$. Although the ligand's rotation is encoded by the four variables, there actually exist three degrees of freedom, and these four variables interact with each other closely. Therefore, an alternative encoding scheme is adopted to avoid the linkage problem in this study, in which the orientations of ligands are represented by three independent angles. An illustration of the three angles is presented in Fig. 2. Let a unit vector $\vec{v}$ represent the rotation axis of a rigid body in three-dimensional Cartesian coordinates and θ represent the rotation angle of this axis. In addition, α represents the angle between $\vec{v}$ and the z-axis, and β represents the angle between the x-axis and the projection $\vec{v'}$ of the rotation axis on the xy-plane. The domains of the three angles are defined as follows: $\theta \in [0, 2\pi), \alpha \in [0, \pi), \beta \in [0, 2\pi)$. The conversion between quaternions and three angles is presented in the Supplementary file. It is worth noting that the angles α and β are similar to those of the spherical coordinate system.

The second component of the linkage problem is related to the interactions among the variables of translation, orientations and torsions, as clearly illustrated in Fig. 3. The variables of a flexible ligand are linked with each other in such a hierarchical form. One upper variable has strong interrelationships with all the other variables below it. For instance, the three variables of the translation affect the results of the orientation and torsions. Even a slight fluctuation of the torsion near root atoms may cause significant adjustments to all the variables of the other torsions. The scenario implies strong interactions among the variables in such a common encoding method, while adopting the crossover operator will destroy these interactions and render the docking problem intractable. Thus, more attention needs to be paid to alleviating the performance degeneration triggered by the crossover operator. It is unwise to remove the crossover operator from the framework of DE directly because the operator still plays an important role in the evolutionary process. Actually, the most straightforward but effective approach is to tune the crossover rate CR in an adaptive manner. For instance, at an early stage, choosing a large CR value can yield sufficient exploratory moves in the global search space. Later on, the fitness degeneration will become more serious along with the optimization process, considering highly correlated variables in the docking problem, especially in the final stage. This requires a small CR value to maintain the linkage among the variables. Fortunately, many efforts have been made to adaptively tune the CR value in the field of EAs, achieving a significant performance improvement, such as SaDE [44], JADE [45] and L-SHADE [31]. Therefore, the parameter adaptation strategy in L-SHADE is used to adjust CR and the scaling factor of differential mutation F in this study. Once the crossover operator affects the fitness of the population, the value of CR adaptively decreases to alleviate the fitness degeneration.

In addition, it is known that EAs suffer from the curse of dimensionality, namely, that their optimization performance deteriorates rapidly as the dimensionality of the problem increases [46]. Indeed, the search ability of EAs weakens in the docking problem when the torsion number of a ligand is large. Moreover, as mentioned above, the parameter adaptation commonly reduces the CR to a small value in the final optimization stage, which will further hamper the search ability of EAs. Accordingly, an adaptive gradient descent algorithm called Adam is employed to improve the optimization performance of the sampling algorithm in high-dimensional docking problems. Unlike other conventional gradient descent algorithms, Adam produces adaptive learning rates for each variable and uses an exponentially decaying average of past gradients to dampen oscillations and to speed up convergence. Adam can achieve robust performances and satisfactory optimization speed with small memory requirements. Thus, it is a suitable choice to search complex conformations in docking problems.

IV. AutoDock Koto

A. Conformation Representation

Molecular docking can be regarded as an optimization process that aims to find the most suitable conformation of the receptor and ligand according to the binding affinity. During the docking process, the receptor is rigid while the ligand remains flexible. The basis of a flexible ligand is a rigid root constituted by a fixed group of atoms. Then, the remaining atoms are connected to the rigid root by rotatable bonds. Thus, the degrees of freedom of a 3D conformation are determined by three components of the ligand molecule: the translation of a rigid root in Cartesian coordinate space $\{P_x, P_y, P_z\}$, the orientation of rigid root rotation represented by three angles $\{\theta, \alpha, \beta\}$, and the flexibilities describing the rotations of dihedral angles in the ligand and certain side-chains of amino acids in the receptor $\{R_1, R_2, ...R_t\}$, where t denotes the number of dihedral angles (torsions). Accordingly, the complex conformation can be represented by a vector X:

$$X = \{P_x, P_y, P_z, \theta, \alpha, \beta, R_1, R_2, ...R_t\}, \tag{1}$$

where its dimensionality is $D = 6 + t$.

B. Scoring Function

The scoring function is the keystone of a protein-ligand docking program. The fast and effective scoring function of AutoDock Vina is also employed in our docking method, which takes advantage of both knowledge-based and empirical scoring functions. In this scoring function, the total free energy of a complex conformation contains intermolecular and intramolecular parts:

$$\Delta G_{total} = \Delta G_{inter} + \Delta G_{intra}. \tag{2}$$

where ΔG_{total} represents the total free energy. ΔG_{inter} is the intramolecular part that measures the free energy between the protein and ligand, and ΔG_{intra} is the intramolecular part that measures the free energy of the ligand itself. The scoring function is used to rank diverse complex conformations, and the sampling algorithm is adopted to search the global minimum of ΔG_{total}.

Both intermolecular and intramolecular parts are calculated by adding up five separate terms in a linear model, which are defined as follows:

$$\begin{aligned}\Delta G = &\sum_{i<j}\{w_1 \Delta H^{i,j}_{gauss_1} + w_2 \Delta H^{i,j}_{gauss_2} + w_3 \Delta H^{i,j}_{repulsion}\} \\ &+ w_4 \sum \Delta H_{hydrophobic} + w_5 \sum \Delta H_{hydrogen},\end{aligned} \tag{3}$$

where $H^{i,j}_{gauss_1}$, $H^{i,j}_{gauss_2}$ and $H^{i,j}_{repulsion}$ represent the steric interactions of all atom pairs in the protein and ligand. $H^{i,j}_{gauss_1}$ and $H^{i,j}_{gauss_2}$ integrate two Gaussian functions to describe the attraction of atom pairs, and $H^{i,j}_{repulsion}$ uses a quadratic function to reflect the repulsion of atom pairs. $H_{hydrophobic}$ denotes the hydrophobic effect between hydrophobic atom pairs. The desolvation process always accompanies the protein-ligand binding. Nonpolar groups tend to aggregate in the aqueous solution, which favors the binding process. $H_{hydrogen}$ records the interaction of hydrogen bonding. Such interaction occurs when two atoms get close enough and form a donor-acceptor pair. Hydrogen bonding can be regarded as the most important factor in the binding mode. In addition, $w_{1\sim5}$ are the weights allocated to different terms, which are tuned through a linear regression technique and set to -0.0356, -0.00516, 0.840, -0.0351 and -0.587, respectively.

The specific definitions of the five terms are presented below:

$$\Delta H^{i,j}_{gauss_1} = exp\{-(d_{i,j}/0.5)^2\}, \tag{4}$$

$$\Delta H^{i,j}_{gauss_2} = exp\{-((d_{i,j} - 3)/2)^2\}, \tag{5}$$

$$\Delta H^{i,j}_{repulsion} = \begin{cases} d^2_{i,j}, & \text{if } d_{i,j} < 0 \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where $d_{i,j} = r_{i,j} - R_i - R_j$, $r_{i,j}$ is the interatomic distance of atoms i and j, and R is the van der Waals radius of an atom.

$$\Delta H_{hydrophobic} = \begin{cases} 1, & \text{if } \hat{d}_{\hat{i},\hat{j}} < 0.5 \\ 1.5 - \hat{d}_{\hat{i},\hat{j}}, & \text{if } 0.5 \le \hat{d}_{\hat{i},\hat{j}} \le 1.5 \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where $\hat{d}_{\hat{i},\hat{j}} = \hat{r}_{\hat{i},\hat{j}} - R_{\hat{i}} - R_{\hat{j}}$, $\hat{r}_{\hat{i},\hat{j}}$ is the interatomic distance between hydrophobic atoms $\hat{i}$ and $\hat{j}$.

$$\Delta H_{hydrogen} = \begin{cases} 1, & \text{if } \bar{d}_{\bar{i},\bar{j}} < -0.7 \\ \bar{d}_{\bar{i},\bar{j}}/(-0.7), & \text{if } -0.7 \le \bar{d}_{\bar{i},\bar{j}} \le 0 \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $\bar{d}_{\bar{i},\bar{j}} = \bar{r}_{\bar{i},\bar{j}} - R_{\bar{i}} - R_{\bar{j}}$, $\bar{r}_{\bar{i},\bar{j}}$ is the length of hydrogen bonding. It measures the interatomic distance between the acceptor $\bar{i}$ and the donor $\bar{j}$ of a hydrogen bond.

The final predicted free energy of binding in each conformation can be calculated by the following formula:

$$\Delta G_{bind} = \frac{G_{total} - \Delta G^*_{intra}}{1 + w_6 N_{\text{rot}}}, \tag{9}$$

where ΔG^*_{intra} is the intramolecular part of the conformation that has the lowest total free energy among the final achieved conformations. N_{rot} represents the rotatable bond number of heavy atoms in the ligand, and w_6 denotes its weight and is set to 0.0585.

C. Gradient Boosting Differential Evolution

A newly designed sampling approach, namely the GBDE algorithm, is developed to search complex conformations in the program Koto. The GBDE algorithm uses an adaptive mechanism to automatically adjust the F and CR parameters and a linear reduction strategy to determine the population size, which were originally proposed in the L-SHADE algorithm [31]. Moreover, the GBDE adopts Adam, an adaptive gradient descent algorithm [32], to speed up convergence and improve the search ability in high-dimensional docking problems.

1) Evolutionary Operators: Similar to a conventional DE algorithm, L-SHADE has three main operators: mutation, crossover and selection. The population is represented as a solution set $\{X_i = (x_1, ..., x_D), i = 1, ..., N\}$, where D represents the dimensionality (i.e., the number of decision variables) of the target problem, and N denotes the population size.

In the initialization, each solution X_i is generated in a random manner. Then, a mutation operator with the "current-to-pbest/1" strategy is used to produce the mutation solutions V. The formula is as follows:

$$V_{i,g} = X_{i,g} + F_i \cdot (X_{\text{pbest},g} - X_{i,g}) + F_i \cdot (X_{r1,g} - X_{r2,g}), \quad (10)$$

where $F_i \in (0, 1]$ is the scaling factor, g represents the current iteration number, $X_{\text{pbest},g}$ is randomly selected from the top $100p\%$ current solutions with the probability $p \in (0, 1]$. $X_{r1,g}$ is randomly selected from the current population, while $X_{r2,g}$ is randomly selected from the current population and an external archive A preserves the solutions that do not survive in the selection operator. The archive has a fixed size $|A|$. Once it overflows, it discards redundant solutions in a random manner.

A binomial crossover operator is employed to produce trial solutions. The formula is described by:

$$U_{j,i,g} = \begin{cases} V_{j,i,g}, & \text{if } \text{rand}(0,1) \leq CR_i \text{ or } j = j_{\text{rand}}\,; \\ X_{j,i,g}, & \text{otherwise}\,; \end{cases} \quad (11)$$

where $CR_i \in [0, 1]$ represents the crossover rate, j represents the dimensional index of the vector, $\text{rand}(0, 1)$ denotes a randomly uniform distribution value from $(0, 1)$, and j_{rand} denotes a dimension index randomly selected from $\{1, ..., D\}$.

The greedy selection operator is used to determine whether $X_{i,g}$ or $U_{i,g}$ survives in the next generation. The formula is the following:

$$X_{i,g+1} = \begin{cases} U_{i,g}, & \text{if } f(U_{i,g}) < f(X_{i,g})\,; \\ X_{i,g}, & \text{otherwise}\,; \end{cases} \quad (12)$$

2) Parameter Adaptation Strategy: In the population, each solution has its control parameters F_i and CR_i. A historical memory with K entries is used to record the mean of control parameters M_F and M_{CR}. At each generation, F_i and CR_i are generated by randomly selecting a set of $M_{F,k}$ and $M_{CR,k}$ in the memory, where $k \in \{1, 2, ...K\}$. The formulas are presented as follows:

$$F_i = \text{randc}(M_{F,k}, 0.05), \quad (13)$$

and

$$CR_i = \text{randn}(M_{CR,k}, 0.1), \quad (14)$$

where $\text{randc}(\mu, \sigma^2)$ and $\text{randn}(\mu, \sigma^2)$ are randomly generated values from a Cauchy and a Normal distributions regarding the mean μ and the variance σ^2. The crossover rates are truncated to the interval [0, 1]. The sampling process will repeat if the random sample is outside.

At each generation, the archives S_F and S_{CR} will record the values of F_i and CR_i, which successfully produce a better trial vector in the selection operator. Then, a new set of values for M_F and M_{CR} are generated by calculating the weighted Lehmer mean of the control parameters in the archives. The formulas of M_F are described below:

$$M_F = \frac{\sum_{p=1}^{|S_F|} w_p \cdot S_{F,p}^2}{\sum_{q=1}^{|S_F|} w_q \cdot S_{F,q}}, \quad (15)$$

and

$$w_p = \frac{|f(U_{p,g}) - f(X_{p,g})|}{\sum_{q=1}^{|S_F|} |f(U_{q,g}) - f(X_{q,g})|}. \quad (16)$$

The value of M_{CR} is updated in the same manner. Next, the generated values of M_F and M_{CR} are stored in the historical memory. If the memory overflows, one of the entries will be randomly abandoned. At the beginning, all the entries are initially set to 0.5. During the evolutionary process, once $max(S_{CR}) = 0$, $M_{CR,k}$ will be set to 0 and will be no longer updated in the remaining generations. This strategy is capable of enhancing the search ability of GBDE for multimodal problems.

3) Population Size Strategy: The population size is set to N^{init} in the initialization, and it is then linearly reduced to N^{final} ($N^{final} < N^{init}$) at the end of the run. Thus, at each generation, the population size N_{g+1} can be calculated as follows:

$$N_{g+1} = \text{round}[(N^{final} - N^{init}) \cdot \frac{g}{G} + N^{init}], \quad (17)$$

where g represents the current iteration number, and G represents the maximum number of iterations. More details regarding these strategies can be found in [31].

4) Adaptive Moment Estimation Algorithm: Adam is a powerful first-order gradient optimization algorithm, which has been widely used to optimize neural networks. The algorithm adaptively determines individual learning rates for each parameter by estimating the first and second moments of gradients.

$$X_{t+1} = X_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t, \quad (18)$$

where η denotes the initialized learning rate. ϵ is set to 10^{-8}. $\hat{m}_t$ and $\hat{v}_t$ represent the bias-corrected first moment estimate and the bias-corrected second raw moment estimate, which are defined as follows:

$$\hat{m}_t = \frac{m_t}{1 - (\lambda_1)^t}, \quad (19)$$

and

$$\hat{v}_t = \frac{v_t}{1 - (\lambda_2)^t}, \quad (20)$$

where m_t denotes the biased first moment estimates and v_t denotes the biased second raw moment estimates. Both are separately updated by:

$$m_t = \lambda_1 \cdot m_{t-1} + (1 - \lambda_1) \cdot g_t, \quad (21)$$

and

$$v_t = \lambda_2 \cdot v_{t-1} + (1 - \lambda_2) \cdot g_t^2, \quad (22)$$

where λ_1 and λ_2 are the exponential averaging factors for the first and second moment estimates of g_t, which are set to 0.5 and 0.999, respectively. In addition, $g_t = \nabla_X f_t(X_{t-1})$, which represents the first-order derivative vector with respect to all

Algorithm 1: GBDE algorithm.

Input: Initialized population size N^{init}, final population size N^{final}, external archive size $|A|$, maximum number of iterations G, and maximum running time (T);
Output: All the solutions in the final population;

$g = 0$, $A = \emptyset$, $S_F = \emptyset$ and $S_{CR} = \emptyset$;
$X_g \leftarrow$ Initialize the population randomly;
$f(X_g) \leftarrow$ Evaluate the fitness of the initialized population;
$M_F = 0.5$, $M_{CR} = 0.5$;
while $g < G$ **do**
  Generate the parameters F and CR using Eq. (13) and (14);
  $V_g \leftarrow$ Produce the mutated solutions using Eq. (10);
  $U_g \leftarrow$ Produce the trial solutions using Eq. (11);
  $f(U_g) \leftarrow$ Evaluate the fitness of trial solutions;
  while $t < T$ **do**
    Select the best solution from U_g, and optimize it using Eq. (18);
  end
  $U_{g+1} \leftarrow$ Produce the next generation of the population using Eq. (12);
  Update A, S_F and S_{CR}, and calculate M_F and M_{CR} using Eq. (15);
  $N_{g+1} \leftarrow$ Determine the current population size using Eq. (17);
  Discard the worst $N_{g+1} - N_g$ solutions from the population based on the fitness;
  $g \leftarrow g + 1$;
end
return Final population.

the design variables of the translation and orientation between protein and ligand as well as each rotatable bond.

For ease of understanding, the derivatives of the position, orientation and torsions can be regarded as the negative total force, the negative total torque and the torque projected on the rotation axis of each rotatable bond, respectively. Applying Adam to update each solution in the population is time-consuming because the DE algorithm produces a set of solutions at each generation, and calculating the gradients of the solutions is very expensive. Therefore, in the GBDE algorithm, only the best trial solution is further optimized by the Adam algorithm before the selection operation. The update will continue until it exceeds the maximum running times (T). The main procedure of the proposed GBDE Algorithm is presented in Algorithm 1.

V. Docking Performance Analysis

To evaluate the docking performance of Koto, seven commonly used docking programs are used as the competitors in the experiments, including Glide, GOLD, Dock (Version 6.9), rdock, LeDock, AutoDock (Version 4.2) and AutoDock Vina. Among them, Glide and GOLD are both commercial programs with high purchase prices which were developed by Schrödinger Inc. and Cambridge Crystallographic Data Centre, respectively [7], [47]. Glide uses an exhaustive search algorithm, and GOLD adopts the GA as the sampling algorithm. Dock is an anchor-and-grow-based docking program for academic use, in which a shape-fitting approach is used to generate low-energy ligand poses [8]. In addition, rdock is a fast open-source docking program maintained by the rDock Development Team, whose sampling algorithm combines a GA and the Monte Carlo method [48]. LeDock is an academic docking program based on a combination of EAs and the simulated annealing algorithm [49], and a comprehensive evaluation verifies that it can achieve very competitive docking performance [29]. AutoDock is a free docking program developed by The Scripps Research Institute [10], and Autodock Vina is its improved version in terms of higher accuracy and speed [11]. Each program uses default optimization parameters in the experiments. The hyperparameters of Koto are set as follows: the initialized population size $N^{init} = 8 \times D$, the final population size $N^{final} = 4$, the external archive size $|A| = 8 \times D$, the maximum number of iterations $G = 800$, the maximum running times $T = 20$ and the initialized learning rate $\eta = 0.01$, where D represents the dimensionality of the problem. For each protein-ligand complex, docking programs are executed 20 times, independently. All the experiments are conducted on a Linux 64-bit system with an Intel Core i5, 3.30 GHz, and 8 GB of memory, using Python and the C++ programming language.

TABLE I
Protein-ligand complexes with different rotatable bond numbers in CASF-2016.

Number of rotatable bonds	Number of complexes	Number of rotatable bonds	Number of complexes
0	6	11	6
1	7	12	3
2	39	13	6
3	31	14	4
4	31	15	5
5	42	16	2
6	24	17	2
7	21	19	1
8	22	20	1
9	16	29	1
10	14	36	1

A. Benchmark Database

The PDBbind database collects protein-ligand complexes including crystal structures and experimental binding affinity and is available online.[1] The latest version of the PDBbind core set, also named the comparative assessment of scoring functions benchmark (CASF-2016) [50], is used to evaluate the docking performance of AutoDock Koto in the experiments. CASF-2016 selects 285 high-resolution protein-ligand complexes from 57 clusters of protein structures and contains strong, medium, and weak binders whose binding affinity spans nearly ten orders of magnitude. For CASF-2016, the numbers of complexes with different rotatable bond numbers

[1] https://archive.ics.uci.edu/ml/index.php

TABLE II
SUCCESS RATES OF THE BEST AND TOP-SCORE POSES GENERATED BY EIGHT DOCKING PROGRAMS.

Programs	Best poses (%)	Top-score poses (%)	Time (h)
Glide	68.31	47.54	34.20
GOLD	85.56	65.49	26.80
rDock	76.84	57.89	**10.95**
LeDock	82.11	70.53	58.21
DOCK	74.06	60.53	122.89
AutoDock	70.07	54.58	242.53
AutoDock-Vina	74.74	64.56	27.54
AutoDock-Koto	**94.39**	**74.19**	34.34

have been summarized in Table I. In addition, the procedure of preparing the target proteins and ligands of complexes, executed by AutoDock Tools [10], can be described as follows:

- Remove water molecules from the crystal structures of protein-ligand complexes.
- Delete the metal ions and add hydrogen atoms to all the atoms of proteins and ligands.
- Assign partial charges and protonation states to each atom of the proteins and ligands.
- Save the files of the proteins and ligands in the PDBQT format.

B. Evaluation Indicators

The root mean square deviation (RMSD) measures the geometric similarity between the docked binding conformation and the crystallographic structure of a protein-ligand complex. The RMSD of the docked conformation, which has a ligand with S heavy atoms, can be calculated by:

$$RMSD = \sqrt{\frac{1}{S}\sum_{s=1}^{S}[(x_s - x_s^*)^2 + (y_s - y_s^*)^2 + (z_s - z_s^*)^2]}, \tag{23}$$

where (x_s, y_s, z_s) and (x_s^*, y_s^*, z_s^*) are the coordinates of the s^{th} heavy atom in the docked conformation and referenced structure, respectively. In general, the prediction of docking is considered to be successful when the RMSD is less than 2.0 Å [51]. A smaller RMSD value of a docked conformation indicates a more powerful docking performance of the program. Each program will produce a set of docked binding conformations in a run. The conformation with the lowest score in the set is chosen as a candidate conformation. Thus, a pool of candidate conformations can be obtained by a docking program within the predefined repetition times. The candidate conformation with the lowest binding affinity is termed the top-scored pose, and the conformation with the smallest RMSD value is termed the best pose. The success rates of the top-scored and best poses are used as two main indicators with which to evaluate the docking performance of each program.

C. Experimental Results

For each docking program, the success rates of the top score and best poses have been presented in Table II. Koto exhibits success rates of 74.19% and 94.39% for the top-score and best poses, which are higher than the success rates of all the other docking programs. LeDock has the second-highest success rate of 70.53% for the top-score poses, and GOLD has the second-highest success rate of 85.56% for the best poses. Even when compared with two commercial programs, namely Glide and GOLD, Koto can still provide better docking performance for 285 protein-ligand complexes. For a more intuitive understanding of the results, bar graphs illustrating the success rates of each program have been presented in Fig. 4. It is easy to observe that Koto has an obvious advantage over the other programs in solving docking problems.

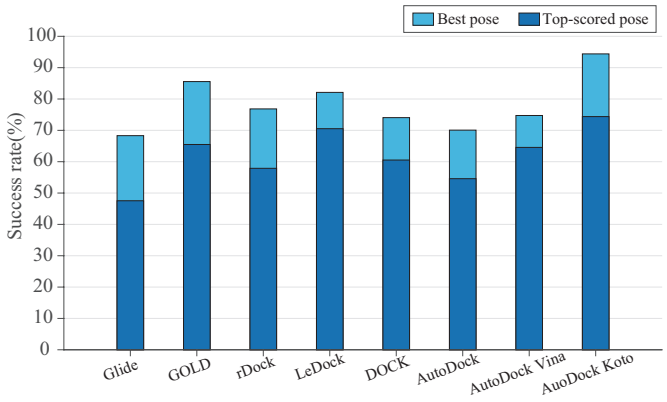


Fig. 4. Bar graphs illustrating the success rates of the best and top-score poses generated by eight docking programs.

Convergence curves of the top-scored conformation's fitness of Koto on six randomly selected complexes are presented in Fig. 5. Although the maximum number of iterations (G) is set to 800, Koto has almost finished the optimization process in the 300^{th} iteration. The maximum number of iterations of each docking program is set to ensure that the optimization algorithms cannot further improve the population's fitness. Such a setting can eliminate the sensitivity of the termination criterion to the docking performance. Regarding the commercial programs, Glide and GOLD, they do not even provide the option of changing the termination criterion to the subscribers. The computation time of each program is presented in Table II. The calculation time of Koto is more than that of rDock, GOLD and Autodock Vina and nearly identical to that of Glide. Compared with DOCK and Autodock, all of them can be regarded as fast docking programs.

As a more comprehensive and effective way to evaluate the sampling power of docking programs, the cumulative frequencies of the success rates with the number of top-score poses, which are taken from every single run, for all the eight docking programs are plotted in Fig. 6. We can observe that the success rates of each program are improved with an increased number of top-score poses. And Koto can consistently achieve a higher success rate than the other programs. In practical applications, for a specified protein-ligand complex, an advisable way is to produce a set of predicted conformations and analyze their structural stability and physicochemical properties, respectively. It is remarkable that the success rates of both DOCK and Glide cannot become higher when the number of top-score poses exceeds a certain

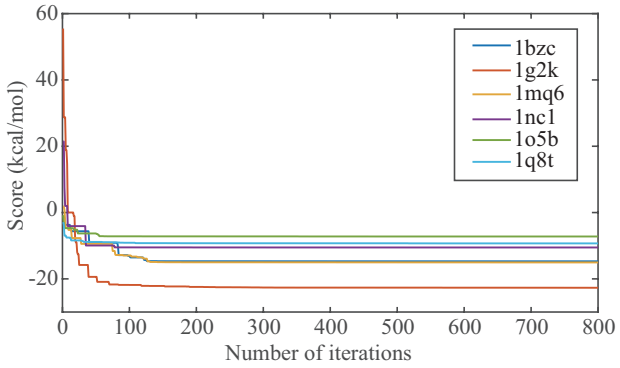


Fig. 5. Convergence curves of the top-scored conformation's fitness of Koto on six randomly selected complexes.

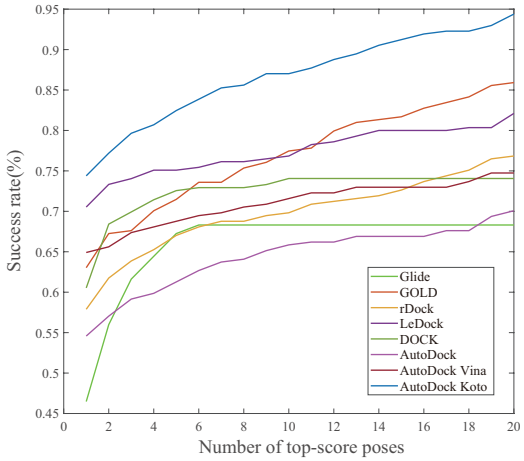


Fig. 6. Cumulative frequency of the success rates for eight docking programs.

threshold. That suggests that the predicted conformations of the two programs are quite similar for different runs.

In addition, the number of rotatable bonds in the ligands is directly related to their flexibility and approximately determines the optimization difficulty of the docking problems. This number has a critical influence on the conformation sampling ability of docking programs. For both the best and top-score poses, the heat maps of docking programs' success rates with different numbers of rotatable bonds are illustrated in Fig. 7 and Fig. 8, respectively. We find that the success rates of most docking programs significantly drop when the number of rotatable bonds increases. In Fig. 7, only the GOLD can successfully dock the largest ligands with 29 rotatable bonds. But, Koto can properly perform to predict the other complex conformations. Similarly, except for the instances with 29 and 36 rotatable bonds, the proposed method can maintain satisfactory docking accuracy in the best pose term, according to Fig. 8. Only GOLD and Glide can perform relatively better than the other comparison programs for the largest ligands with 29 and 36 rotatable bonds. Compared with GOLD, Koto can achieve superior performances on the uncomplicated docking problems, while it performs relatively worse on the complicated docking problems with a larger number of rotatable bonds. Fortunately, the rotatable bonds of most ligands are less than 10. Koto obtains an obvious advantage in practical applications.

Koto adopts the Adam algorithm to optimize one of the solutions in the current population during each iteration. With the aid of gradient information, Adam can effectively improve the fitness of the solution. Since the population size of the GBDE algorithm decreases with the number of iterations, this improvement has a more significant influence on the population, especially at the end of the optimization phase. Using Adam to optimize all the solutions is time-consuming and results in premature convergence. Thus, the strategy adopted in our docking program can significantly speed up convergence and improve the optimization performance of the sampling algorithm in high-dimensional docking problems. Remarkably, as shown in Table I, the number of complexes that possess a large number of rotatable bonds is relatively small in CASF-2016. In fact, over 90.0% of the existing drugs approved by the FDA have fewer than 10 rotatable bonds. Therefore, more attention should be given to the complexes with a small number of rotatable bonds. From Fig. 7 and Fig. 8, it can be seen that, even for the rigid ligands with no rotatable bonds, not all the docking programs can perform well except for Koto and GOLD. However, Koto is capable of providing more precise predictions of ligand conformations than other programs for most of these complexes.

To visually compare the docking results of eight programs, the predicted complex conformations of four typical instances are shown in Fig. 9. For each instance, the binding site of a protein is framed by an orange cuboid, and the referenced structures and docked conformations of ligands are marked in red and blue, respectively. Only Koto, LeDock and Glide can successfully dock the ligand into the binding pocket of the protein for a simple instance of 1o0h. The surfaces of proteins in 2wn9 and 2r9w are more rugged, indicating that there are more local minima in the landscape of the search space. Koto can also provide precise predictions of ligand poses. In addition, the ligand has 15 rotatable bonds in 3bv9, which suggests that predicting this complex is a high-dimensional optimization problem. Only Koto, Glide and GOLD can generate accurate docked conformations, and the conformation predicted by Koto is more similar to the reference structure of the ligand than those predicted by the other programs, according to the data in Fig. 7 and Fig. 8. In summary, Koto can produce satisfactory and robust solutions to solve protein-ligand docking problems. This benefits from the strong optimization ability of the GBDE algorithm on such multimode and high-dimensional docking problems, compared with the sampling algorithms in the other docking programs.

D. Discussion

The gradient descent algorithm plays a vital role during the optimization process of Autodock Koto. To analyze the sensitivity to the maximum number of iterations of the gradient descent algorithm (T), a comparison of Koto with different numbers of iterations is presented in Table III. It can be seen that employing the gradient descent algorithm can significantly improve the docking performance of Koto. The computation of gradient descent occupies 22.5% of the calculation time in

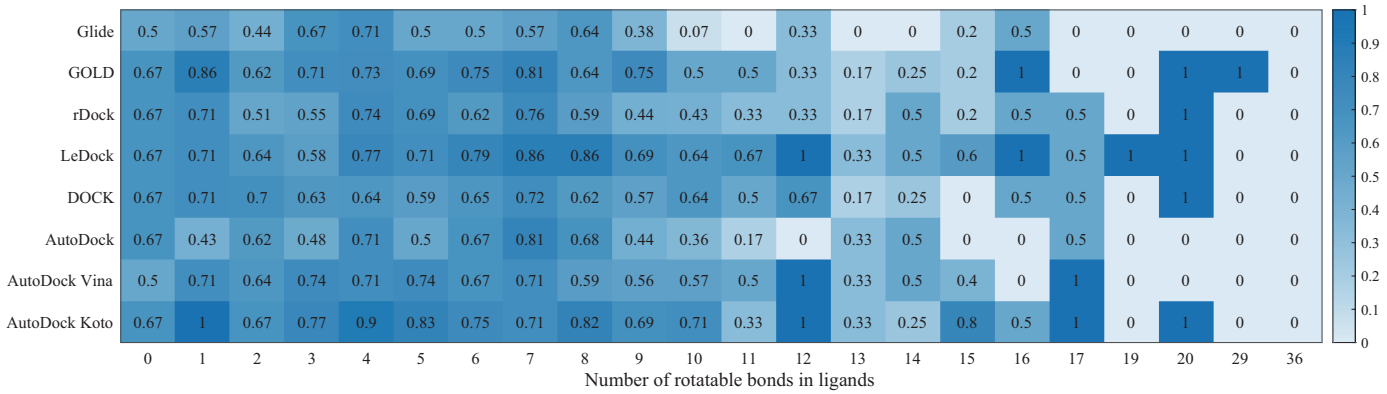


Fig. 7. Heat map of the success rates of the top-score poses generated by eight docking programs for the ligands with different numbers of rotatable bonds.

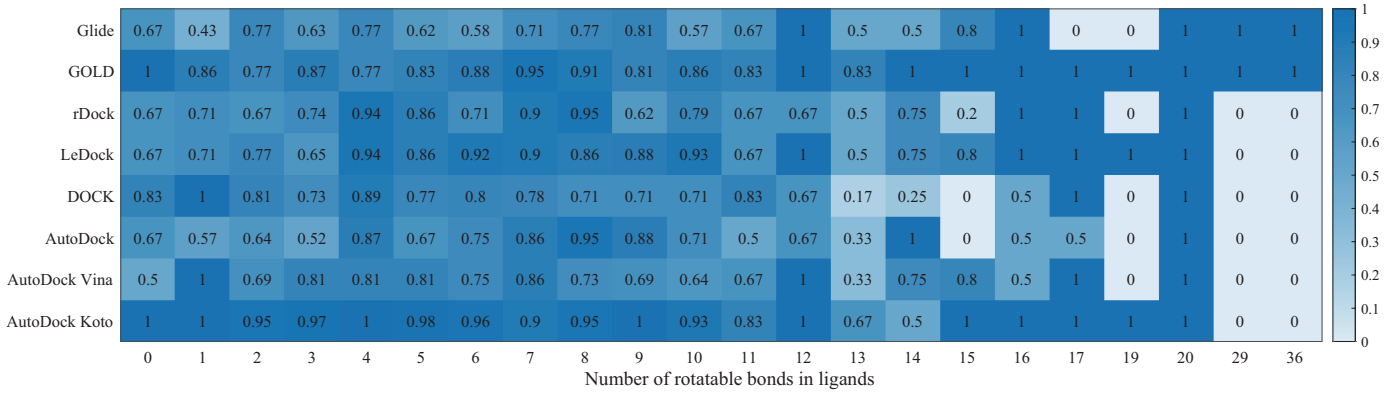


Fig. 8. Heat map of the success rates of the best poses generated by eight docking programs for the ligands with different numbers of rotatable bonds.

TABLE III
COMPARISON OF AUTODOCK KOTO WITH DIFFERENT RUNNING TIMES T.

Parameters	Best poses (%)	Top-score poses (%)	Time (h)
T=0	90.53	73.68	26.60
T=20	94.39	74.39	34.34
T=40	93.33	72.63	41.50
T=60	92.28	70.88	48.40
T=80	91.23	69.82	54.13

TABLE IV
ESTIMATION OF THE RELATIVE IMPORTANCE OF THE MAIN COMPONENTS IN AUTODOCK KOTO.

Models	Best poses (%)	Top-score poses (%)	Time (h)
Baseline	94.39	74.39	34.34
Half iterations	92.63	73.33	22.40
No gradient descent	90.53	73.68	26.60
No PS reduction	91.93	70.88	65.17
No CR adaptation	91.93	70.53	38.04
Variation	90.88	72.98	37.95

Koto when T is set to 20. Furthermore, setting a larger T does not further improve the program's accuracy but significantly increases the computational time.

In addition, the relative importance of the main components in Koto is also estimated by training and evaluating several ablation models. **Baseline**: Baseline model, as described above in the paper. Other ablation models should be understood relative to this baseline model. **Half iterations**: The maximum number of iterations of the sampling algorithm is halved, i.e., $G = 400$. **No gradient descent**: The component of the gradient descent algorithm is removed, and its running times (T) are set to 0. **No PS reduction**: The population size strategy is discarded. Instead, a fixed population size (N) is used, which is set to 100 in our experiments. **No CR adaptation**: The parameter adaptation strategy is not applied to adjust the crossover rate (CR), which is initially set to 0.1 and remains unchanged in each generation. **Variation**: The gradient descent algorithm is only employed to refine all the solutions in the last generation. The maximum running time (T) is set to 600.

The results of ablation models are presented in Table IV. It can be observed that the halved maximum number of iterations leads to a slight degeneration of docking performance but results in a drastic reduction of running time. In addition, the accuracy of the top-score and best poses will decrease when any main component is abandoned in the program. This implies that parameter adaptation, gradient descent and population reduction all contribute to the high performance of Koto. The results of the variation are worse than Koto's, suggesting that applying the gradient descent algorithm to refine the best solution at each generation can yield a better balance between exploration and exploitation during the optimization process.

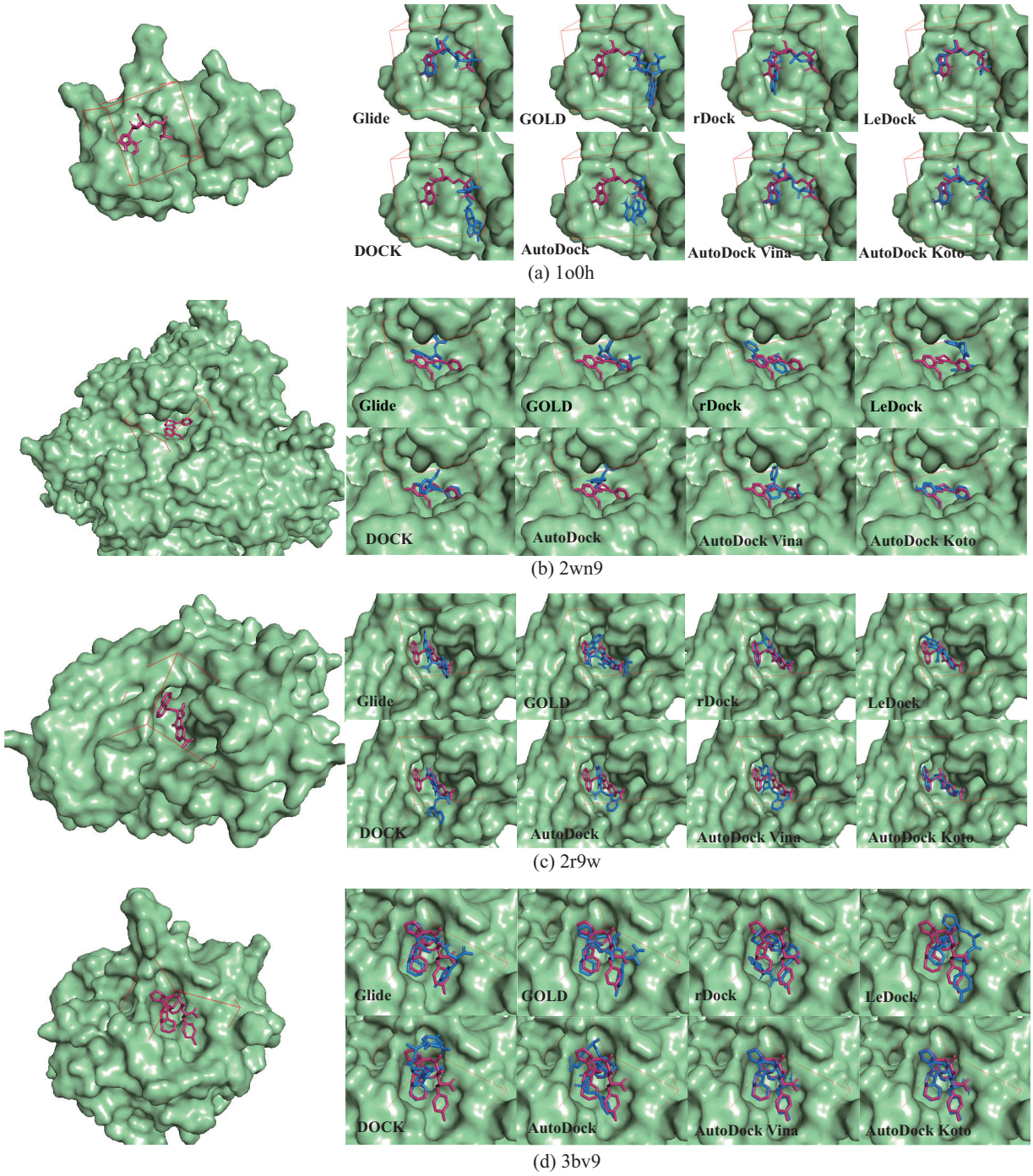


Fig. 9. Visualization of complex conformations predicted by eight docking programs for four typical instances. Surfaces of target proteins are shown in green, and binding sites are framed by orange cuboids. The referenced structures and docked conformations of ligands are separately marked in red and blue.

VI. Virtual Screening for COVID-19

A. Target Identification

SARS-CoV-2 is a novel coronavirus originally found in bats, which has adapted to infect humans. The genomic and protein sequences for SARS-CoV-2 have been accessible in the NIH gene data bank [52]. Sequence analysis suggests that SARS-CoV-2 shares 89% sequence similarity with other SARSCoVs [53]. The similarity enabled the rapid target identification of SARS-CoV-2 for drug design, which can profit from an earlier study on drug developments of SARS-CoV and MERS-CoV [54]. The main protease of SARS-CoV-2 M^{pro} is a pivotal coronavirus enzyme for mediating viral replication and transcription because it is in charge of the cleavage of polyproteins at cleavage sites. Thus, M^{pro} can be identified as an attractive drug target [55]. Furthermore, M^{pro} has a high degree of specificity and has not yet matched that of human proteases, which implies that it can be used to design inhibitors of the viral protease without inhibiting essential human protease activities [56]. In clinical trials, lopinavir and

TABLE V
APPROVED DRUGS WITH THE LOWEST BINDING ENERGIES FROM THE SUPERDRUG2 DATABASE.

	Ligand index	Binding energy (kcal/mol)	Drug name	Reported
1	SD001727	-10.0980	Nilotinib	[65], [66]
2	SD000929	-9.7324	Ergotamine	No
3	SD003345	-9.6293	Tirilazad	[67]
4	SD000790	-9.5128	Dihydroergotamine	No
5	SD002828	-9.4736	Guamecycline	No
6	SD006009	-9.4709	Glecaprevir	No
7	SD003930	-9.3926	Radotinib	No
8	SD000789	-9.2436	Dihydroergocristine	No
9	SD000783	-9.1955	Digitoxin	[68]
10	SD000521	-9.1343	Chlorhexidine	[69], [70]
11	SD000044	-9.1165	Acetyldigitoxin	No
12	SD000784	-9.0922	Digoxin	[71]
13	SD001561	-9.0399	Metildigoxin	No
14	SD000045	-9.0266	Acetyldigoxin	No
15	SD001313	-9.0227	Irinotecan	No
16	SD001676	-9.0010	Nandrolone cyclotate	No

ritonavir have been assessed as potential protease inhibitors of M^{pro} for the treatment of COVID-19 [57], [58]. However, no clinically available antiviral drugs have been developed for SARS-CoV-2 to date. Therefore, Koto is adopted to identify more inhibitors of M^{pro} by virtually screening other approved drugs in this study.

B. Protein and Ligand Preparation

The 3D structures of the target protein SARS-CoV-2 M^{pro} have been accessible in the Protein Data Bank (PDB) [59]. The structure of SARS-CoV-2 M^{pro} with a covalent inhibitor N3, marked 6LU7 (2.16Å resolution) [60], [61], is used for virtual screening of the approved drugs in our experiments. The 3D structures of ligands in approved drugs are obtained from SuperDRUG2, which is a database of approved and marketed drugs.[2] The latest version of SuperDRUG2 collected over 4600 active pharmaceutical ingredients [62], and the first conformer of ligands in the list of each drug is utilized. Charges are added to both the target protein and each ligand, and the structures are then converted into the pdbqt format by Obabel [63]. The visualization of the generated compounds is implemented by PyMOL [64].

C. Experimental Results

Considering that there exist large differences among the sizes of ligands in SuperDRUG2, the sizes of the grid boxes were set to 15Å, 18.5Å and 22.5Å for each dimension of the coordinate system, respectively. The grid boxes are located in the center of the region of the active site. The number of repetition times of Koto is set to 20 for each ligand. Since the binding affinity is always considered as a priority for evaluating the best candidate in virtual screening, the criterion that the binding energy (ΔG_{bind}) is less than or equal to -9 kcal/mol is used to evaluate the efficiency of the interaction between the ligands and the target structures. According to the criterion, a total of 16 approved drugs in SuperDrug2 screened by Koto, which have the potential to bind to SARS-CoV-2 M^{pro}, have been listed in Table V. Computationally determined binding modes of all the ligands in these drugs predicted by Koto are presented in Fig. S. (1∼4) of the Supplementary file.

Nilotinib has the best binding affinity predicted by Koto, which achieves a binding energy of -10.0980 kcal/mol. A slightly higher binding energy can also be predicted by the other docking programs, such as Vina and SMINA [72]. Nilotinib is an approved antagonist of tyrosine kinase to treat chronic myelogenous leukemia [73]. Additionally, its antiviral efficacy has also been validated in animal models [74]. Consequently, the efficacy of nilotinib against SARS-CoV-2 in vitro has been demonstrated in [65], [66]. The results are in accordance with the prediction of Koto that nilotinib achieved the best binder of M^{pro}.

Ergotamine is predicted to achieve the second-best binding energy of -9.7324 kcal/mol by Koto. A pi-alkyl and a pi-pi T-shaped hydrophobic interaction exist between the ligand of ergotamine and the catalytic residues of M^{pro}. Ergotamine is an approved non-antiviral drug and used to treat acute migraine-type headache [75]. It is known that ergotamine acts on extracellular G-protein coupled receptors. It still does not have the ability to engage intracellular targets. Accordingly, the efficacy of Ergotamine against SARS-CoV-2 has not previously been validated either in vitro or in vivo. In addition, dihydroergotamine has a similar structure to that of ergotamine, which has been widely used in the treatment of migraine [76]. Similarly, it also achieves the fourth-lowest binding energy of -9.5128 kcal/mol by Koto. Two hydrogen bonds and hydrophobic interactions are exhibited between the ligand of dihydroergotamine and residues of M^{pro}.

The binding ability of tirilazad, used to treat acute ischemic stroke [77], ranks third among all the approved drugs, with a predicted binding energy of -9.6293 kcal/mol by Koto. Tirilazad binds with M^{pro} by forming four H-bond interactions with the amino acids of M^{pro}. The same predicted results can also be found in [67]. Guamecycline, a tetracycline derivative for the treatment of acute pneumopathies, has a good binder with M^{pro} with a predicted binding energy of -9.4736 kcal/mol. Although guamecycline has not been suggested to treat COVID-19, it has been found to have significant interactions with other viral target proteins, such as Dengue, Japanese encephalitis and Ebola [78]. Accordingly, guamecycline is also a potential antiviral drug for SARS-CoV-2.

In addition, the antiviral effects of digitoxin have been verified in cotton rat lungs and can effectively reduce host-driven cytokine storms caused by SARS-CoV-2 [68]. Similarly, the antiviral activity of digoxin against SARS-CoV-2 infection has also been assessed in [71]. As an antimicrobial agent, chlorhexidine has an anti-bacterial effect and has been shown to be effective against SARS-CoV-2 in clinical trials [79], [80]. In addition to the above approved drugs, glecaprevir, radotinib, dihydroergocristine, acetyldigitoxin, metildigoxin, acetyldigoxin, irinotecan and nandrolone cyclotate also achieve good binding affinities as shown in Table V. The repurposing of approved drugs with high ΔG_{bind} presented in the experiment are preferred candidates as therapies of COVID-19. Thus, it is worthwhile to further investigate their therapeutic efficacy in vitro and in clinical trials.

[2]http://bioinf.charite.de/superdrug

VII. Conclusions

This study introduced a protein-ligand docking program called Autodock Koto, which adopts the novel GBDE algorithm as its sampling approach. Experimental results showed that GBDE can effectively alleviate the performance degeneration caused by the linkage problem and the curse of dimensionality during the optimization phase. Compared with either commercial or academic docking programs, Koto yields dramatic improvements in the success rates of generating crystal-like complex conformations. Moreover, Koto has also been applied to virtually screen antiviral drugs for COVID-19, considering its powerful docking performance. Sixteen approved drugs were found to have strong binders with the main protease of SARS-CoV-2. These findings in the experiment can serve as a guide for pharmaceutical experts and clinicians, which may help them to further validate the therapeutic efficacy of COVID-19 in vitro and large clinical trials.

We believe that the development of the Koto program will give the community a valuable freely available tool for identifying chemically promising compounds in drug discovery and design pipelines. In addition, an in-depth analysis reveals that scoring functions remain challenging for selecting the correct docked conformations and that their performance varies greatly among different target proteins. Therefore, there exists plenty of room on scoring functions for further improvement of docking programs. Our future work will focus on developing accurate as well as time-saving and cost-effective scoring functions through deep learning architectures, which may bring docking applications to a new stage.

References

[1] F. Ban, K. Dalal, H. Li, E. LeBlanc, P. S. Rennie, and A. Cherkasov, "Best practices of computer-aided drug discovery: lessons learned from the development of a preclinical candidate for prostate cancer with a new mechanism of action," *Journal of Chemical Information and Modeling*, vol. 57, no. 5, pp. 1018–1028, 2017.

[2] T. T. Talele, S. A. Khedkar, and A. C. Rigby, "Successful applications of computer aided drug discovery: moving drugs from concept to the clinic," *Current Topics in Medicinal Chemistry*, vol. 10, no. 1, pp. 127–141, 2010.

[3] D. J. Newman and G. M. Cragg, "Natural products as sources of new drugs over the nearly four decades from 01/1981 to 09/2019," *Journal of Natural Products*, vol. 83, no. 3, pp. 770–803, 2020.

[4] L. Chen, J. K Morrow, H. T Tran, S. S Phatak, L. Du-Cuny, and S. Zhang, "From laptop to benchtop to bedside: structure-based drug design on protein targets," *Current Pharmaceutical Design*, vol. 18, no. 9, pp. 1217–1239, 2012.

[5] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, pp. 1–11, 2021.

[6] R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry *et al.*, "Glide: a new approach for rapid, accurate docking and scoring. 1. method and assessment of docking accuracy," *Journal of Medicinal Chemistry*, vol. 47, no. 7, pp. 1739–1749, 2004.

[7] G. Jones, P. Willett, R. C. Glen, A. R. Leach, and R. Taylor, "Development and validation of a genetic algorithm for flexible docking," *Journal of Molecular Biology*, vol. 267, no. 3, pp. 727–748, 1997.

[8] W. J. Allen, T. E. Balius, S. Mukherjee, S. R. Brozell, D. T. Moustakas, P. T. Lang, D. A. Case, I. D. Kuntz, and R. C. Rizzo, "Dock 6: Impact of new features and current docking performance," *Journal of Computational Chemistry*, vol. 36, no. 15, pp. 1132–1156, 2015.

[9] A. N. Jain, "Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine," *Journal of Medicinal Chemistry*, vol. 46, no. 4, pp. 499–511, 2003.

[10] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson, "Autodock4 and autodocktools4: Automated docking with selective receptor flexibility," *Journal of Computational Chemistry*, vol. 30, no. 16, pp. 2785–2791, 2009.

[11] O. Trott and A. J. Olson, "Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading," *Journal of Computational Chemistry*, vol. 31, no. 2, pp. 455–461, 2010.

[12] J. Lyu, S. Wang, T. E. Balius, I. Singh, A. Levit, Y. S. Moroz, M. J. O'Meara, T. Che, E. Algaa, K. Tolmachova *et al.*, "Ultra-large library docking for discovering new chemotypes," *Nature*, vol. 566, no. 7743, pp. 224–229, 2019.

[13] R. R. Ramsay, M. R. Popovic-Nikolic, K. Nikolic, E. Uliassi, and M. L. Bolognesi, "A perspective on multi-target drug discovery and design for complex diseases," *Clinical and Translational Medicine*, vol. 7, no. 1, pp. 1–14, 2018.

[14] S. Pushpakom, F. Iorio, P. A. Eyers, K. J. Escott, S. Hopper, A. Wells, A. Doig, T. Guilliams, J. Latimer, C. McNamee *et al.*, "Drug repurposing: progress, challenges and recommendations," *Nature Reviews Drug Discovery*, vol. 18, no. 1, pp. 41–58, 2019.

[15] S.-Y. Huang and X. Zou, "Advances and challenges in protein-ligand docking," *International Journal of Molecular Sciences*, vol. 11, no. 8, pp. 3016–3034, 2010.

[16] J. Li, A. Fu, and L. Zhang, "An overview of scoring functions used for protein–ligand interactions in molecular docking," *Interdisciplinary Sciences: Computational Life Sciences*, vol. 11, no. 2, pp. 320–328, 2019.

[17] M. Gupta, R. Sharma, and A. Kumar, "Docking techniques in pharmacology: How much promising?" *Computational Biology and Chemistry*, vol. 76, pp. 210–217, 2018.

[18] Y.-C. Chen, "Beware of docking!" *Trends in Pharmacological Sciences*, vol. 36, no. 2, pp. 78–95, 2015.

[19] H.-M. Chen, B.-F. Liu, H.-L. Huang, S.-F. Hwang, and S.-Y. Ho, "Sodock: Swarm optimization for highly flexible protein–ligand docking," *Journal of Computational Chemistry*, vol. 28, no. 2, pp. 612–623, 2007.

[20] H. K. Tai, S. A. Jusoh, and S. W. Siu, "Chaos-embedded particle swarm optimization approach for protein-ligand docking and virtual screening," *Journal of Cheminformatics*, vol. 10, no. 1, pp. 1–13, 2018.

[21] K. M. Wong, H. K. Tai, and S. W. Siu, "Gwovina: A grey wolf optimization approach to rigid and flexible receptor docking," *Chemical Biology & Drug Design*, vol. 97, no. 1, pp. 97–110, 2021.

[22] E. López-Camacho, M. J. G. Godoy, J. Garcia-Nieto, A. J. Nebro, and J. F. Aldana-Montes, "Solving molecular flexible docking problems with metaheuristics: A comparative study," *Applied Soft Computing*, vol. 28, pp. 379–393, 2015.

[23] P. F. Leonhart, E. Spieler, R. Ligabue-Braun, and M. Dorn, "A biased random key genetic algorithm for the protein–ligand docking problem," *Soft Computing*, vol. 23, no. 12, pp. 4155–4176, 2019.

[24] C. Li, J. Sun, and V. Palade, "Msldock: Multi-swarm optimization for flexible ligand docking and virtual screening," *Journal of Chemical Information and Modeling*, vol. 61, no. 3, pp. 1500–1515, 2021.

[25] A. Grosdidier, V. Zoete, and O. Michielin, "Eadock: docking of small molecules into protein active sites with a multiobjective evolutionary optimization," *Proteins: Structure, Function, and Bioinformatics*, vol. 67, no. 4, pp. 1010–1025, 2007.

[26] S. Janson, D. Merkle, and M. Middendorf, "Molecular docking with multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 8, no. 1, pp. 666–675, 2008.

[27] A. Sandoval-Perez, D. Becerra, D. Vanegas, D. Restrepo-Montoya, and F. Nino, "A multi-objective optimization energy approach to predict the ligand conformation in a docking process," in *European Conference on Genetic Programming*. Springer, 2013, pp. 181–192.

[28] J. García-Nieto, E. López-Camacho, M. J. García-Godoy, A. J. Nebro, and J. F. Aldana-Montes, "Multi-objective ligand-protein docking with particle swarm optimizers," *Swarm and Evolutionary Computation*, vol. 44, pp. 439–452, 2019.

[29] Z. Wang, H. Sun, X. Yao, D. Li, L. Xu, Y. Li, S. Tian, and T. Hou, "Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power," *Physical Chemistry Chemical Physics*, vol. 18, no. 18, pp. 12 964–12 975, 2016.

[30] N. S. Pagadala, K. Syed, and J. Tuszynski, "Software for molecular docking: a review," *Biophysical Reviews*, vol. 9, no. 2, pp. 91–102, 2017.

[31] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 1658–1665.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] K. Stromgaard, P. Krogsgaard-Larsen, and U. Madsen, *Textbook of drug design and discovery*. CRC press, 2009.

[34] D. E. Gordon, G. M. Jang, M. Bouhaddou, J. Xu, K. Obernier, K. M. White, M. J. O'Meara, V. V. Rezelj, J. Z. Guo, D. L. Swaney *et al.*, "A sars-cov-2 protein interaction map reveals targets for drug repurposing," *Nature*, vol. 583, no. 7816, pp. 459–468, 2020.

[35] E. Fischer, "Einfluss der configuration auf die wirkung der enzyme," *Berichte Der Deutschen Chemischen Gesellschaft*, vol. 27, no. 3, pp. 2985–2993, 1894.

[36] G. G. Hammes, "Multiple conformational changes in enzyme catalysis," *Biochemistry*, vol. 41, no. 26, pp. 8221–8228, 2002.

[37] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui, "Molecular docking: a powerful approach for structure-based drug discovery," *Current Computer-aided Drug Design*, vol. 7, no. 2, pp. 146–157, 2011.

[38] J. J. Irwin and B. K. Shoichet, "Docking screens for novel ligands conferring new biology: Miniperspective," *Journal of Medicinal Chemistry*, vol. 59, no. 9, pp. 4103–4120, 2016.

[39] Y.-p. Chen, T.-L. Yu, K. Sastry, and D. E. Goldberg, "A survey of linkage learning techniques in genetic and evolutionary algorithms," *IlliGAL Report*, vol. 2007014, 2007.

[40] T.-L. Yu, D. E. Goldberg, K. Sastry, C. F. Lima, and M. Pelikan, "Dependency structure matrix, genetic algorithms, and effective recombination," *Evolutionary Computation*, vol. 17, no. 4, pp. 595–626, 2009.

[41] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2013.

[42] J. Wang, W. Zhang, and J. Zhang, "Cooperative differential evolution with multiple populations for multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2848–2861, 2015.

[43] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, 1985, pp. 245–254.

[44] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2008.

[45] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[46] Y. Mei, M. N. Omidvar, X. Li, and X. Yao, "A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 42, no. 2, pp. 1–24, 2016.

[47] R. A. Friesner, R. B. Murphy, M. P. Repasky, L. L. Frye, J. R. Greenwood, T. A. Halgren, P. C. Sanschagrin, and D. T. Mainz, "Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein- ligand complexes," *Journal of Medicinal Chemistry*, vol. 49, no. 21, pp. 6177–6196, 2006.

[48] S. Ruiz-Carmona, D. Alvarez-Garcia, N. Foloppe, A. B. Garmendia-Doval, S. Juhos, P. Schmidtke, X. Barril, R. E. Hubbard, and S. D. Morley, "rdock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids," *PLoS Computational Biology*, vol. 10, no. 4, p. e1003571, 2014.

[49] H. Zhao and A. Caflisch, "Discovery of zap70 inhibitors by high-throughput docking into a conformation of its kinase domain generated by molecular dynamics," *Bioorganic & Medicinal Chemistry Letters*, vol. 23, no. 20, pp. 5721–5726, 2013.

[50] M. Su, Q. Yang, Y. Du, G. Feng, Z. Liu, Y. Li, and R. Wang, "Comparative assessment of scoring functions: the casf-2016 update," *Journal of Chemical Information and Modeling*, vol. 59, no. 2, pp. 895–913, 2018.

[51] T. Gaillard, "Evaluation of autodock and autodock vina on the casf-2013 benchmark," *Journal of Chemical Information and Modeling*, vol. 58, no. 8, pp. 1697–1706, 2018.

[52] F. Wu, S. Zhao, B. Yu, Y.-M. Chen, W. Wang, Z.-G. Song, Y. Hu, Z.-W. Tao, J.-H. Tian, Y.-Y. Pei *et al.*, "A new coronavirus associated with human respiratory disease in china," *Nature*, vol. 579, no. 7798, pp. 265–269, 2020.

[53] Y. Wan, J. Shang, R. Graham, R. S. Baric, and F. Li, "Receptor recognition by the novel coronavirus from wuhan: an analysis based on decade-long structural studies of sars coronavirus," *Journal of Virology*, vol. 94, no. 7, pp. e00 127–20, 2020.

[54] K. Anand, J. Ziebuhr, P. Wadhwani, J. R. Mesters, and R. Hilgenfeld, "Coronavirus main proteinase (3clpro) structure: basis for design of anti-sars drugs," *Science*, vol. 300, no. 5626, pp. 1763–1767, 2003.

[55] T. R. Tong, "Severe acute respiratory syndrome coronavirus (sars-cov)," *Perspectives in Medical Virology*, vol. 16, pp. 43–95, 2006.

[56] L. Zhang, D. Lin, X. Sun, U. Curth, C. Drosten, L. Sauerhering, S. Becker, K. Rox, and R. Hilgenfeld, "Crystal structure of sars-cov-2 main protease provides a basis for design of improved α-ketoamide inhibitors," *Science*, vol. 368, no. 6489, pp. 409–412, 2020.

[57] Z. Wang, X. Chen, Y. Lu, F. Chen, and W. Zhang, "Clinical characteristics and therapeutic procedure for four cases with 2019 novel coronavirus pneumonia receiving combined chinese and western medicine treatment," *Bioscience Trends*, 2020.

[58] W.-j. Guan, Z.-y. Ni, Y. Hu, W.-h. Liang, C.-q. Ou, J.-x. He, L. Liu, H. Shan, C.-l. Lei, D. S. Hui *et al.*, "Clinical characteristics of coronavirus disease 2019 in china," *New England Journal of Medicine*, vol. 382, no. 18, pp. 1708–1720, 2020.

[59] S. K. Burley, H. M. Berman, G. J. Kleywegt, J. L. Markley, H. Nakamura, and S. Velankar, "Protein data bank (pdb): the single global macromolecular structure archive," *Protein Crystallography*, pp. 627–641, 2017.

[60] K. Y. Wang, F. Liu, R. Jiang, X. Yang, T. You, X. Liu, C. Q. Xiao, Z. Shi, H. Jiang, Z. Rao *et al.*, "Structure of mpro from covid-19 virus and discovery of its inhibitors," *Nature*, 2020.

[61] X. Liu, B. Zhang, Z. Jin, H. Yang, and Z. Rao, "The crytal structure of 2019-ncov main protease in complex with an inhibitor n3," *RCSB Protein Data Bank*, vol. 10, 2020.

[62] V. B. Siramshetty, O. A. Eckert, B.-O. Gohlke, A. Goede, Q. Chen, P. Devarakonda, S. Preissner, and R. Preissner, "Superdrug2: a one stop resource for approved/marketed drugs," *Nucleic Acids Research*, vol. 46, no. D1, pp. D1137–D1143, 2018.

[63] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison, "Open babel: An open chemical toolbox," *Journal of Cheminformatics*, vol. 3, no. 1, pp. 1–14, 2011.

[64] W. L. DeLano *et al.*, "Pymol: An open-source molecular graphics tool," *CCP4 Newsletter on Protein Crystallography*, vol. 40, no. 1, pp. 82–92, 2002.

[65] S. Banerjee, S. Yadav, S. Banerjee, S. O. Fakayode, J. Parvathareddy, W. Reichard, S. Surendranathan, F. Mahmud, R. Whatcott, J. Thammathong *et al.*, "Drug repurposing to identify nilotinib as a potential sars-cov-2 main protease inhibitor: Insights from a computational and in vitro study," *Journal of Chemical Information and Modeling*, 2021.

[66] V. Cagno, G. Magliocco, C. Tapparel, and Y. Daali, "The tyrosine kinase inhibitor nilotinib inhibits sars-cov-2 in vitro," *Basic & Clinical Pharmacology & Toxicology*, vol. 128, no. 4, pp. 621–624, 2021.

[67] S. Chtita, A. Belhassan, A. Aouidate, S. Belaidi, M. Bouachrine, and T. Lakhlifi, "Discovery of potent sars-cov-2 inhibitors from approved antiviral drugs via docking and virtual screening," *Combinatorial Chemistry & High Throughput Screening*, vol. 24, no. 3, pp. 441–454, 2021.

[68] B. S. Pollard, J. C. Blanco, and J. R. Pollard, "Classical drug digitoxin inhibits influenza cytokine storm, with implications for covid-19 therapy," *In Vivo*, vol. 34, no. 6, pp. 3723–3730, 2020.

[69] C. Wu, Y. Liu, Y. Yang, P. Zhang, W. Zhong, Y. Wang, Q. Wang, Y. Xu, M. Li, X. Li *et al.*, "Analysis of therapeutic targets for sars-cov-2 and discovery of potential drugs by computational methods," *Acta Pharmaceutica Sinica B*, vol. 10, no. 5, pp. 766–788, 2020.

[70] G. Galindez, J. Matschinske, T. D. Rose, S. Sadegh, M. Salgado-Albarrán, J. Späth, J. Baumbach, and J. K. Pauling, "Lessons from the covid-19 pandemic for advancing computational drug repurposing strategies," *Nature Computational Science*, vol. 1, no. 1, pp. 33–41, 2021.

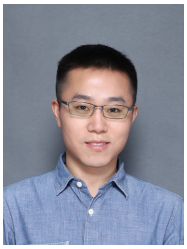
[71] J. Cho, Y. J. Lee, J. H. Kim, S. il Kim, S. S. Kim, B.-S. Choi, and J.-H. Choi, "Antiviral activity of digoxin and ouabain against sars-cov-2 infection and its implication for covid-19," *Scientific Reports*, vol. 10, no. 1, pp. 1–8, 2020.

[72] T. Sekhar, "Virtual screening based prediction of potential drugs for covid-19," *Combinatorial Chemistry & High Throughput Screening*, vol. 23, 2020.

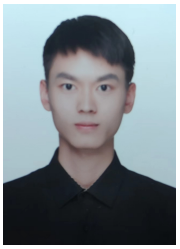
[73] E. Jabbour, J. Cortes, and H. Kantarjian, "Nilotinib for the treatment of chronic myeloid leukemia: An evidence-based review," *Core Evidence*, vol. 4, p. 207, 2009.

[74] H. K. Ananthula, S. Parker, E. Touchette, R. M. Buller, G. Patel, D. Kalman, J. S. Salzer, N. Gallardo-Romero, V. Olson, I. K. Damon *et al.*, "Preclinical pharmacokinetic evaluation to facilitate repurposing of tyrosine kinase inhibitors nilotinib and imatinib as antiviral agents," *BMC Pharmacology and Toxicology*, vol. 19, no. 1, pp. 1–11, 2018.

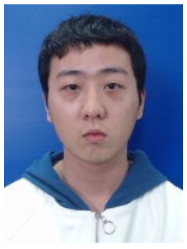
[75] P. Tfelt-Hansen, P. Saxena, C. Dahlöf, J. Pascual, M. Lainez, P. Henry, H.-C. Diener, J. Schoenen, M. Ferrari, and P. Goadsby, "Ergotamine in the acute treatment of migraine: a review and european consensus," *Brain*, vol. 123, no. 1, pp. 9–18, 2000.
[76] S. D. Silberstein and D. C. McCrory, "Ergotamine and dihydroergotamine: history, pharmacology, and efficacy," *Headache: The Journal of Head and Face Pain*, vol. 43, no. 2, pp. 144–166, 2003.
[77] T. I. S. Committee *et al.*, "Tirilazad for acute ischaemic stroke," *Cochrane Database of Systematic Reviews*, no. 4, 2001.
[78] N. Kumar, H. Sarma, and G. N. Sastry, "Repurposing of approved drug molecules for viral infectious diseases: a molecular modelling approach," *Journal of Biomolecular Structure and Dynamics*, pp. 1–17, 2021.
[79] J. G. Yoon, J. Yoon, J. Y. Song, S.-Y. Yoon, C. S. Lim, H. Seong, J. Y. Noh, H. J. Cheong, and W. J. Kim, "Clinical significance of a high sars-cov-2 viral load in the saliva," *Journal of Korean Medical Science*, vol. 35, no. 20, 2020.
[80] Y. H. Huang and J. T. Huang, "Use of chlorhexidine to eradicate oropharyngeal sars-cov-2 in covid-19 patients," *Journal of Medical Virology*, vol. 93, no. 7, pp. 4370–4373, 2021.



Junkai Ji received the B.S. degree from Hefei University of technology, Anhui, China, in 2013, and an M.S. degree and a D.E. degree from University of Toyama, Toyama, Japan, in 2016 and 2018, respectively. In 2019, he joined Shenzhen University, Shenzhen, China, where he is currently an Assistant Professor in the National Engineering Laboratory for Big Data System Computing Technology. His current research interests include neural networks, evolutionary computation and computer-aided drug design.



Jin Zhou received the B.S. degree from Jiangxi University of Science and Technology, Nanchang, China in 2020. He is currently pursuing his M.S. degree in College of Computer Science and Software engineering, Shenzhen University. His research focuses on protein-ligand docking and protein-protein docking.



Zhangfan Yang received the B.S. degree from Hubei Polytechnic University, HuangShi, China in 2019. He is currently pursuing his M.S degree in College of Computer Science and Software Engineering, Shenzhen University. His research focuses on evolutionary computation and deep learning methods for computer-aided drug design.

Qiuzhen Lin (Member IEEE) received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from De-partment of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. He has published over 100 research papers since 2008. His research interests include artificial immune system, multi-objective optimization, and dynamic system. Dr. Lin is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Jianqiang Li (Member, IEEE) received the B.S. and Ph.D. degrees in automation from the South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He led five projects of the National Natural Science Foundation and four projects of the Natural Science Foundation of Guangdong Province, China. His current research interests include robotics, embedded systems, and Internet of Things.

Carlos A. Coello Coello (Fellow, IEEE) received the Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996. He is a Professor (CINVESTAV-3F Researcher) with the Department of Computer Science of CINVESTAV-IPN, Mexico City, Mexico. He has authored and coauthored over 450 technical papers and book chapters. He has also coauthored the book Evolutionary Algorithms for Solving Multi-Objective Problems (Second Edition, Springer, 2007). His publi-cations currently report over 57 600 citations in Google Scholar (his H-index is 95). His research interests include evolutionary multiobjec-tive optimization and constraint-handling techniques for evolutionary algorithms.

Dr. Coello Coello was a recipient of the 2007 National Research Award from the Mexican Academy of Sciences in the area of Exact Sciences, the 2013 IEEE Kiyo Tomiyasu Award, and the 2012 National Medal of Science and Arts in the area of Physical, Mathematical and Natural Sciences. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He is a member of the Association for Computing Machinery and the Mexican Academy of Science.