Self-adaptive differential evolution algorithm incorporating local search for protein-ligand docking

Hwan Won Chung\textsuperscript{1}, Seung Joo Cho\textsuperscript{2}, Kwang-Ryeol Lee\textsuperscript{1} and Kyu-Hwan Lee\textsuperscript{1}

\textsuperscript{1} Center for Computational Science, Korea Institute of Science and Technology, South Korea
\textsuperscript{2} Department of Cellular and Molecular Medicine, University of Chosun, South Korea
E-mail: hwchung@kist.re.kr

Abstract. Differential Evolution (DE) algorithm is powerful in optimization problems over several real parameters. DE depends on strategies to generate new trial solutions and the associated parameter values for searching performance. In self-adaptive DE, the automatic learning about previous evolution was used to determine the best mutation strategy and its parameter settings. By combining the self-adaptive DE and Hooke Jeeves local search, we developed a new docking method named SADock (Strategy Adaptation Dock) with the help of AutoDock4 scoring function. As the accuracy and performance of SADock was evaluated in self-docking using the Astex diverse set, the introduced SADock showed better success ratio (89\%) than the success ratio (60\%) of the Lamarckian genetic algorithm (LGA) of AutoDock4. The self-adapting scheme enabled our new docking method to converge fast and to be robust through the various docking problems.

1. Introduction
During the process to discover a new drug, the molecular docking is used frequently to find new promising hit molecules and also to optimize the binding affinities of lead compounds. The molecular docking consists of two components \cite{1}. The first is a scoring function to define and measure the interactions which a ligand experiences when it approaches to a receptor. The second is a searching method to find the numerous poses that the ligand will take to fit into the active site of the receptor. Many searching methods are based on stochastic optimizations such as genetic algorithm, particle swarm optimization, and DE, etc\cite{1, 2}.

DE is an evolutionary strategy that develops a population of real solution vectors using operators such as mutation, crossover, and selection\cite{3}. The performance of DE is significantly impacted by three major parameters: the number of solution vectors in the population (NP), mutation scaling (F), and crossover rate (CR)\cite{3}.

The present paper is focused on applying the self-adapting differential evolution \cite{4} to the molecular docking problem using AutoDock4 scoring \cite{1, 5} and estimating new searching method. Subsequently, we will describe the basic DE algorithm, its enhancement by self-adapting parameters, and explanation about the molecular docking problem. After introducing docking experiment, benchmark docking result and its analysis will be explained. At last, some concluding remarks will be shown.
2. Methods

2.1. Basic Differential Evolution
DE evolves a population using evolutionary operators and a scoring function to find optimal solutions for the given problem. The population consists of NP target vectors \( x_{i,G} \), \( i = 1, \ldots, NP \) with \( G \) generation number and \( NP \) population size.

After initialization of the population, a target vector is mutated to a mutation vector \( u_{i,G} \) with the difference vector of two target vectors as 

\[ u_{i,G} = x_{r1,G} + F(x_{r2,G} - x_{r3,G}), \quad r1 \neq r2 \neq r3 \]

where \( F \) is a mutation scaling factor and \( r1, r2, r3 \) are random indexes. Next, the mutant vector \( u_{i,G} \) will undergo a recombination with the current target vector \( x_{i,G} \) to generate a trial vector \( v_{i,G} \) in the crossover operator as follows: if \( rand < CR \) or \( i = rnd_j \) then \( v_{i,G} = u_{i,G} \) else \( v_{i,G} = x_{i,G} \). Here, the \( CR \) is a crossover rate, the \( rand \) is a random number in \( 0, 1 \), and the \( rnd_j \) is one index of the random target vector. The selection operator of DE is greedy; DE will choose the trial vector only if the fitness value of the trial vector is better than that of the current target vector.

2.2. Self-adaptive Differential Evolution
According to A.K. Qin[4], DE can increase performance using the adaptive change of its mutation strategies and the parameter values. SADock was implemented to include such a mutation strategy pool consisting of four mutation methods with diverse behaviors so that each strategy could show its strength at the different stage of the evolution. The probability to choose each strategy was updated from the previous learning of successful trial solutions.

In SADock, the scaling parameter \( F \) was generated from a normal distribution \( N(0.5, 0.3) \) and the crossover parameter \( CR \) was also generated from another normal distribution \( N(CR_{m_k}, 0.1) \). With the first normal distribution, random \( F \) values were sampled and assigned to the target vectors in the current population. Then, \( N(CR_{m_k}, 0.1) \) was used to create random \( CR \) values and these values were assigned to the target vectors according to their strategy types. The initial value of the \( CR_{m_k} \) for the kth strategy is assigned to 0.5. Qin also holds that \( CR \) value is critical for optimal performance and different strategies may need different \( CR \) values[4]. Therefore, the \( CR_m \) values were also adapted by the previous learning experiences.

The \( NP \) was a user-defined parameter because it depends on the difficulty of a given problem.

For the more detailed explanation about the self-adaptive DE, see [4].

2.3. Representation and Scoring Function
The molecular docking searches for optimal binding poses with the lowest energy of a ligand in the problem space defined by three movements: translation, orientation, and conformation[1]. At first, the ligand will define its position by three coordinates of \( t_x, t_y, t_z \). Next, the ligand can change its orientation using a rotation axis and a rotation angle. The rotation axis is unit vector of \( n_x, n_y, n_z \in \{0, 1\} \) and the ligand will rotate as much as the rotation angle \( \alpha \in \{-\pi, \pi\} \) around the rotation axis. Moreover, the ligand can take conformations corresponding to the combination of all these torsion angles \( (T) \) with the angle \( \in \{-\pi, \pi\} \). So, one pose will have \( 7 + T \) number of total parameters to be optimized.

Autodock4 scoring function was used to calculate empirical energy to estimate a docked conformation[1]. This scoring function evaluates the binding energy between the ligand and the receptor using the intermolecular energy, the internal energy of the ligand, and the desolvation energy: 

\[ E_{total}(X) = E_{inter} + E_{internal} + E_{desolation} \]

where \( E_{inter} \) consists of van der Waals force, hydrogen bonding, and electrostatic potential and \( E_{internal} \) also comprises these three forces. The last term models a charge-based desolvation and hydrophobic effect[6, 5].
Table 1. Molecular docking result of four algorithms in terms of success number and ratio.

| # torsional range \(^a\) | # complex | LGA | SADock(m1) | SADock(m4) | SADock(m4,LS) |
|------------------------|-----------|-----|------------|------------|---------------|
| 0 ~ 5 (A)              | 39        | 32  | 35 (0.89)  | 35 (0.89)  | 35 (0.89)     |
| 6 ~ 10 (B)             | 40        | 27  | 32 (0.80)  | 36 (0.90)  | 35 (0.87)     |
| 11 ~ 15 (C)            | 6         | 1   | 4 (0.66)   | 5 (0.83)   | 5 (0.83)      |
| total                  | 85        | 60  | 71 (0.83)  | 76 (0.89)  | 75 (0.88)     |

\(^a\)All the complexes are classified into three groups (A, B, and C) according to the number of torsional angles. \(^b\)After docking success number, success ratio are surrounded by parentheses. \(^c\)The cells with the bold type have the best success ratio.

3. Docking Experiment

We used Astex diverse set\(^7\) with 85 protein-ligand complexes in the docking experiment. Gasteiger partial charge was generated for every ligand and protein with AutodockTools. The grid box of each receptor was calculated for its corresponding ligand to be moved freely in its extended state. As a success criterion of the docking, the root mean squared deviation (RMSD) was calculated between the reference structure and predicted conformation. When the RMSD was less than 2.0, the associated docking run was considered as a success.

We implemented three different algorithms for comparison: SADock(m1), SADock(m4), SADock(m4,LS). Four mutation strategies used in our strategy adaptation were DE/rand/1, DE/current to best/2, DE/best/1, and DE/rand/2. Since the DE/current to best/2 was the best strategy in a preliminary experiment, SADock(m1) used only this mutation. SADock(m4) had four mutation strategies. In SADock(m4,LS), Hooke Jeeves local search\(^8\) was embedded to refine the current global best. This refinement was applied only when the new global best was found and a random number was less than 0.01.

Each docking run was iterated for 10 times and one conformation with the best (lowest) energy was chosen for comparison. LGA was assigned the population size of 150 and the evaluation number of 1,000,000 and another parameters were chosen with the default values. In the all SADocks, the learning period were 50 and the CR\(_m\) learning period was 20. However, the evaluation number for all SADock algorithms was assigned to be one fifth (200,000) of that of LGA.

4. Results and Discussion

Table 1 shows the molecular docking result of four algorithms and the docking result is summarized by three groups according to the number of rotatable bonds. In group A, all SADock algorithms produced the same best success rate (89%) whereas LGA produced the lower success rate (82%) than the others. In group B, SADock(m4) yielded the best success rate of 90%. LGA and SADock(m1) showed the distinct weak performance in this group. In group C, SADock(m4) and SADock(m4,LS) brought about the best result (83%). In a whole, SADock(m4) algorithm produced the best success ratio (89%). SADock(m4,LS) did not show significant improvement than SADock(m4) and it is likely that SADock(m4,LS) will need additional docking experiments using different refinement ratios.

Figure 1 shows that four mutation strategies and associated parameters are adapting to the evolution stage. The probability to select each mutation strategy is drawn in the panel (a) of Figure 1. In the early stage of the evolution, DE/current to best/2 and DE/rand/2 showed high selection probability whereas DE/rand/1 and DE/best/1 showed low probability. As the evolution proceeds, the selection probability of DE/current to best/2 increased continuously; On the contrary, the other probabilities decreased continuously. The CR\(_m\) value of each strategy
is drifting about the range between 0.01 and 0.2 in the panel (b) of Figure 1. In most of our docking experiments, the $CR_{m}$ value was observed in the range between 0.01 and 0.25. It is likely that the molecular docking problem is generally considered to be multi-modal.

The primary idea about SADock was to treat docking problems without adjusting the major parameters manually. Without such manual tuning, our SADock showed better docking success ratio than LGA when only one fifth of the total evaluations of LGA was used. Moreover, SADock(m4) showed better success ratio (89%) than SADock(m1) (83%) without the strategy adaptation. This explains that the strategy adaptation is essential for the efficiency and robustness of SADock.

5. Conclusion
The result of this study indicated that the SADock outperformed AutoDock LGA in the self-docking of the Astex diverse set due to its adaptation of the mutation strategies and parameters. Our SADock was both fast and robust in reproducing the bounded conformation of ligands. These findings means that our SADock can be applied to the virtual screening applications for the drug discovery. In the future, we need to develop more mutation strategies and to test more diverse conditions when embedding the local search.

Acknowledgments
We would like acknowledge an institutional research program of Korean Institute of Science and Technology (KIST) (Grant No. 2E22790).

References
[1] Morris G M, Goodsell D S, Halliday R S, Huey R, Hart W E, Belew R K and Olson A J 1998 Journal of Computational Chemistry 19 1639–1662 ISSN 1096-987X
[2] Thomsen R 2003 The 2003 Congress on Evolutionary Computation, 2003. CEC ’03 vol 4 pp 2354 – 2361 Vol.4
[3] Storn R and Price K 1997 J. of Global Optimization 11 341359 ISSN 0925-5001
[4] Qin A, Huang V and Suganthan P 2009 Evolutionary Computation, IEEE Transactions on 13 398 –417 ISSN 1089-778X
[5] Morris G, Huey R, Lindstrom W, Sanner M, Belew R, Goodsell D and Olson A 2009 Journal of computational chemistry 30 27852791
[6] Huey R, Morris G M, Olson A J and Goodsell D S 2007 Journal of Computational Chemistry 28 11451152 ISSN 1096-987X
[7] Hartshorn M J, Verdonk M L, Chessari G, Brewerton S C, Mooij W T M, Mortenson P N and Murray C W 2007 J. Med. Chem. 50 726–741 ISSN 0022-2623
[8] Hooke R and Jeeves T A 1961 J. ACM 8 212229 ISSN 0004-5411