International Journal of Engineering & Technology, 7 (4.31) (2018) 528-532

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Multistage Genetic Algorithm and Q-learning for Flexible Ligand-Protein Docking

Erzam Marlisah*, Razali Yaakob, Md. Nasir Sulaiman, Mohd Basyaruddin Abdul Rahman, M.N. Shah Zainudin

Faculty of Computer Science and Information Technology, UPM Serdang, 43400 Serdang, Selangor, Malaysia.
**Corresponding author E-mail: erzam@upm.edu.my*

Abstract

Protein-ligand docking is an optimization task involving translation and rotation of orientation and torsional angles of a small molecule (ligand) with respect to a target protein. Traditional genetic algorithm can be used to find the optimal conformation, however it often found poor structure with high docked energy due to premature convergence and its weakness in performing precision search. The proposed algorithm is a multistage genetic algorithm with Q-learning agent to overcome the limitations inherent in genetic algorithm and reinforcement learning algorithm. The idea is to combine the explorative speed of genetic algorithm in finding promising area in the search space and the ability of the reinforcement learning agent to do fine-grained search. Docking of ten ligands to thermolysin as the target protein shows the proposed algorithm is more efficient in finding the lowest docked energies and more reliable in finding similar structure every run compared to traditional genetic algorithm and AutoDock Vina in docking highly flexible ligand. The algorithm almost matches AutoDock Vina in docking less flexible ligands and outperforms it in docking highly flexible ligands.

Keywords: *Protein docking; genetic algorithm; reinforcement learning; Q-learning agent; optimization.*

1. Introduction

Interaction between a ligand (small molecule such as substrates and inhibitors) and macromolecules such as protein, enzymes or DNA, determines the biochemical properties of the resulting new structure. The structure of the protein is modified after binding it with a ligand. As a result, the original function of the protein will be altered. This ability to engineer novel protein structure with desired functionalities is of great relevance to the pharmaceutical and medicinal sciences in discovering new drugs. Protein docking is a process within the drug discovery process [1]. The drug discovery process involves screening thousands of small molecules or ligands to bind to a large protein to inhibit or catalyse its chemical functions. The protein-ligand docking task is to find a suitable ligand's pose. A pose is characterized by the ligand's position, orientation and the torsion angles of the rotatable bonds(conformation) of the ligand in relation to the binding site of the protein. Central to this task is therefore the searching algorithm to find the best conformation with lowest docked energy. The suitability of the pose is scored by an energy function where the lower the interaction energy between the ligand and the protein means the more stable and spontaneous the resulting protein complex will be [2]. Therefore, protein docking is considered as an optimization problem that minimizes an energy function by finding the optimum values for position, orientation and torsion angles [3].

2. Related works

Genetic algorithm (GA) has been used as one of the solutions to the protein docking problem [4-6]. The position, orientation and torsion angles of the ligands in relation to the protein are encoded in chromosome and the energy given by energy function is assigned as the fitness value of the chromosome. Genetic algorithm evolves a population of these chromosomes to find the better docking confirmation. However, it was reported by some works that the standard genetic algorithm must be modified to improve the quality of solution in protein docking problem [7-8].

Q-learning is an algorithm in the realm of reinforcement learning where the decision maker interacts with the environment to learn the best action to take in given state receiving a reward or punishment throughout the interaction [9-10]. The concept of learning while interacting without given specific instructions on how to achieve a task gave popularity to its application in designing an autonomous system for controllers [11-15]. The classic Q-learning uses table representation called Q-Table to map the value of taking an action a from a state s, $Q(s,a)$ or Q-values which is a function of the observed rewards and previously learned value:

$$Q(s,a) = Q(s,a) + \propto[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (1)$$

Where $\propto$ is the learning rate and $\max_{a'} Q(s',a')$ means the value of taking the best action in the successor state. To learn the true value of every possible pair of states' S and actions A, the Q-learning agent needs to visit and update them many times.

A highly flexible ligand has many rotatable bonds increasing the number of variables in optimizing its conformation. As a result, standard reinforcement learning algorithm, genetic algorithm and quasi-newton methods are less efficient in finding an optimum conformation for highly flexible ligands. For example, due to the use of continuous real value and the high number of dimensions, a

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

reinforcement learning agent adopting a Q-Table to store the value of visited state and action pairs will find it impossible to store and update every possible value. Genetic algorithm has also been reported of having decreasing performance in docking when the number of torsion angles increases [16]. This is due to the genetic algorithm encountered premature convergence and stuck in a local optimum [17]. There exist more sophisticated algorithms such as PSOVina, ALPS and AutoDock VINA reported to find best conformation with low docked energy [17-19]. Both PSOVina and AutoDockVina use the same quasi-newton Broyden-Fletcher-Goldfarb-Shanno method. The problem using the gradient methods is that the performance degrades in docking highly flexible ligand due to high number of dimensions. The work on ALPS only compared itself to Lamarckian Genetic Algorithm implemented in the older version of AutoDock prior to the newer and more efficient AutoDock Vina.

3. Methodology

The proposed method is called Steady State Genetic Algorithm with Q-Learning (SSGAQL) that combines the global optimization power of genetic algorithm and local search capability of Q-learning agent to efficiently find low docked energy for highly flexible ligands. We are not comparing the performance in terms of speed and memory requirement. Our priority is on finding the best conformation between two molecules, especially in docking highly flexible ligands.

3.1. Steady-State Genetic Algorithm Solution

A ligand's pose at any time is real-value encoded as a chromosome with three genes: positional, rotational and torsional. Individuals for reproduction are selected using fitness proportionate selection scheme implemented by roulette-wheel algorithm. If the offspring are identical, random offspring generation will be performed to maintain diversity. Crossover operation depends on the gene being selected. Single-point crossover is applied to position and torsion genes. For rotational gene, crossover operation is performed to either axis or angle component. Single-point crossover is performed to the axis component and the angles are just swapped. Mutation assigns new position within the search box and new rotational within gyration radius. Each torsion angles is mutated within $-\pi$ and π. The best individual which survives through the generations will be kept in an elite pool of size 1.

3.2. Q-Learning Solution

In Q-learning solution, protein-ligand docking problem is represented whereby the environment is the protein and ligand in three-dimensional bounded space. The agent can be viewed as a 'navigator' for the ligand on how to move so as to find the best pose. The state is the ligand's root atom's three-dimensional position, its rotational angle and torsions angles, describing its current conformation in relation to the protein. Since optimizing protein structures variables are represented in real values, the state space must be discretized. All possible position of the ligand's root atom in the three-dimensional space is discretized into small cubes specified by grids with length 0.2Å in each axis, as illustrated in Figure 1(a).

The orientation angle is discretized into bins of equal size with unit measurement in radians. The rotation axis used is the initial rotation axis of the ligand and is constant. Torsion angle is also discretized into bins similar to the orientation angle. The value of 5° increments for both rotation angle and torsion angle is used in experiments. The actions the agent can take are to move the position of the root atom, rotate the angle of orientation and rotate the torsion angles (if any). For moving the position of the root atom, the three-dimensional space is partitioned into octants and the position is represented in spherical coordinates, as in Figure 1(b). The agent will move the central atom's initial position into one of these octants. When the agent chose an octant, the current position is translated to a new position with distance 0.2 Å as illustrated in Figure 1(c).

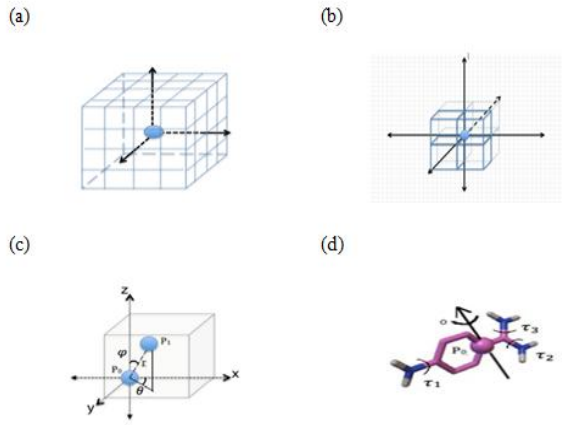


Fig. 1: Agent's state and action representation, (a) Ligand's root atom in the center of search space discretized into grids, (b) Translational action space divided into octants, (c) The agent's moving Benzamidine's root atom's position, P_0, to a new position P_1 in the first octant region, (d) The agent increases/decreases the angle O and torsion angle τ_1, τ_2 and τ_3.

The agent keeps track of the last position it took in every octant and subsequent translation in that particular octant will be from that last recorded position. When the new position exceeded the boundary of an octant, the atom's position will be reset to the initial position. Therefore, the agent will only explore the translational action space within the agent's chosen octant. The agent can increase or decrease the rotation and torsion angle by 5°.

3.3. Q-Learning with Torsion Prototypes (QLTP)

Even with the state and action space discretized as proposed above, the size of the resulting Q-Table will still be very large. Regular Kanerva coding [20] reduces the size by choosing a subset from S. QLTP method differs from the regular Kanerva coding as the subset was derived by discretizing only the torsion angles T and not the other state variables. To generate a torsion prototype, randomly pick one of the three bins for the first torsion angle and do the same for the remaining torsion angles. The remaining prototypes are generated by making sure there are no duplicates. The number of visits, m, to each prototype is recorded. Prototypes that are visited too frequently will be split and those prototypes visited infrequently will be deleted. The decision to split or delete depends on the probability based on the exponential function: $\rho = e^{-m}$. Split is done by choosing randomly one of the current torsion angle and randomly choose an immediate neighbor to the current bin. The algorithm checks for relevant prototypes every time a visit to any prototype exceeds certain maximum limit. Every time a state transition occurs, the value of the approximate Q-Table, Q(s,a) is updated as below:

$$Q_i(s,a) = Q_i(s,a) + \mu_i(s,a)[\alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (2)$$

where i refers to the torsion prototype contained by (s,a) pair and $\mu_i(s,a)$ is the membership function assigning the pair to adjacent prototypes. This has the effect of multiple updates following an agent's visit to a state s and its action a.

3.4. Reward Function

The reward for each conformation produced by the agent is calculated based on the scoring function, C:

$$C = C_{intramolecular} + C_{intermolecular} \quad (3)$$

where $C_{intramolecular}$ is the sum of forces interacting between atoms within a molecule and $C_{intermolecular}$ is the sum of forces interacting between molecules.

Interested reader may obtain a detailed description of the scoring function and its derivation from the work of Trott and Olson [19]. Better conformation gives lower energy. In general, a positive energy value denotes a very poor resulting structure. The difference of energy is the current conformation energy that QLTP found minus the starting conformation's energy. The reward function assigns more rewards to the agent in improving its search. An improvement of 0.2kcal/mol will be rewarded with the value 1. An improvement of more than 0.2kcal/mol but under 0.5kcal/mol will gain a reward of 5. A bigger improvement will be given 100 reward. Likewise, the agent will be punished if its search found worse conformation energy. The reward function pseudo-code used by QLTP agent is summarized in Figure 2.

```
if the difference is positive then
    if (difference > -0.2) then reward 1
    else if(difference <= -0.2 && difference > -0.5) then reward 5;
    else reward 100;
if the difference is negative, i.e. the current conformation's energy is
worse than starting's energy
    if (difference < 0.2) then punish -1;
      else if(difference >= 0.2 && difference< 0.5) then punish -5;
    else punish -100;
```

Fig. 2: QLTP reward function

3.4. Proposed Algorithm

The GA representation described previously is adopted by SSGAQL. The difference between the GA used in SSGAQL and the traditional GA is in the number of generations used. In SSGAQL, the number of generations is kept small. The value used in the experiment is 300 generations, often inadequate to find the best solution in complex problem. After running for 300 generations, the GA restarts with a new population, keeping the elite individual to survive from the previous evolution. The number of repeated evolutions in this stage is predefined to be half of the maximum number of permitted evolutions in the first stage. The pseudo-code for the algorithm is shown in Figure 3.

```
Procedure SSGAQL
Input: search box coordinates, number of evolutions n
Output: ligand's pose
01. Repeat for n/2 evolutions
02. Generate new population of chromosomes
03. Repeat for some short generations
04.     Check for convergence
05.     Select individuals using fitness proportionate selection
scheme
06.     Do crossover
07.     Do mutation
08.         If offspring identical then do random offspring generation
09.     Replace a portion of weakest individuals with the fittest
10.     elite = best_individual;
11.     Put elite individual in elite pool of size 1
12. local_best = QLTP(elite)
13. If local_best fitness > elite fitness then swap
14. Adjust range of search space at a certain n/2 +1 evolution
15. Return local_best
```

Fig. 3: SSGAQL algorithm

When the first stage is complete, the QLTP agent starts its search based on the evolved elite individual. Figure 4 illustrates the steps taken by agent to explore the search space. After the agent's search is complete, it enters the second stage by running the GA again but now with a narrowed down search space to provide search intensification. The QLTP agent runs again based on the elite individual found. The prototypes visited too frequently are split into new prototypes in hope that these prototypes would accurately represent more relevant torsion angle states. A prototype split is done by incrementing or decrementing the value of one of its randomly chosen element. The new prototype is then added to the existing torsion prototypes set. Prototypes that are infrequently visited could be deleted and replaced by the new prototypes produced from the split. The algorithm will reinitialize the Q-Table with new state-action pairs based on the new torsion prototype set and resetting the visit count to all prototypes to zero.

```
Procedure QLTP
Input: An elite individual's pose
Output: A refined ligand's pose
01: Setup Q-Table(best_inividual.position);
02: local_best = best_individual;
03: Repeat for a number of steps
04:         Decrease ε according to a log function;
05:         Choose action a according to e-greedy;
06:       S_{t+1};
07:         Reward = QLTP_reward_function;
08:         If individual's E at S_{t+1} < local_best E then local_best =
individual at s_{t+1};
09:         update_Q-table(S_t,S_{t+1},reward);
10: Return local_best;
```

Fig. 4: QLTP algorithm

4. Result and Discussion

The performance of SSGAQL is compared against GA and VINA in docking ten ligands to thermolysin. Thermolysin is a metalloenzyme originates from thermophilic bacterium Bacillus thermoproteolyticus [21]. The ligands chosen were of varying size, torsions and physical properties and their codes in PDB together with information on their physical properties are summarised in Table 1. In the table, n is the number of rotatable bonds.

Table 1: The flexibility and the physical properties of the test ligands.

	PDB code	n	Surface Area ($Å^2$)	Volume ($Å^{3)}$	Size
Phosphoethanolamine	PSE	8	319.85	469.71	Large
Phenylalanine	PHE	5	349.85	530.21	Large
Propandiol	PDO	4	233.10	309.26	Large
P-aminobenzamidine	PBZ	3	314.96	464.19	Medium
Phenylacetic acid	PAC	3	306.08	455.24	Medium
Benzamidine	BEN	2	296.97	436.23	Medium
Propanol	POL	2	218.94	282.89	Medium
Ethanolamine	ETA	1	204.31	263.04	Small
Ethanol	EOH	1	187.31	233.92	Small
Phenanthroline	PHN	0	356.26	550.56	Small

The binding of these ligands helps in synthesizing new artificial enzymes with many applications in biotechnology related industries [23]. Both the protein and ligands structures were obtained from The Protein Data Bank website (http://www.rcsb.org/pdb/) [22].

The performance observed in terms of effectiveness and reliability. Effectiveness is measured by algorithm's ability in finding the best conformation with the lowest energy. Reliability is the algorithm's consistency in finding similar best conformation each run. It is measured by the mean of best conformation's energy in 10 runs. The number of individual in a population used in this experiment is 50. Crossover probability is fixed at 0.8 so that more individual in the population will be subjected to change so that the agent will explore more of the local search space. The mutation rate is kept at 0.02 so that the new individual could be produced occasionally to avoid stagnation. The values used for QLTP's parameters are described in Table 2. It is important for the algorithms to run in approximately the same number of energy evaluations to be judged fairly. As it is difficult to fix the number of energy evaluation in VINA runs, therefore VINA was run first and the number of energy evaluations used by VINA was set as the limit for SSGAQL and GA.

Table 2: QLTP parameters

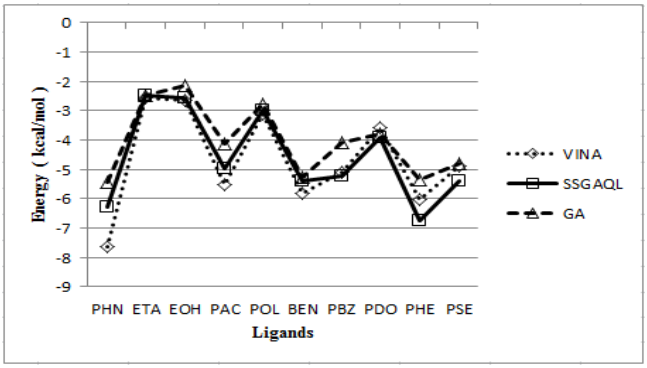
	α	γ	Granularity	Cubes	No. of proto-types	No. of steps
PHN, EOH, ETA	0.8	0.5	0.3	2	3	5000
BEN, POL, PAC, PBZ	Polynomial	0.5	0.3	2	3	5000
PDO, PHE, PSE	0.1	0.5	0.3	2	4	2500

Table 3 compares the lowest and the mean energy found by all three algorithms. In the table, V represents VINA and SG is SSGAQL. The number in bracket in Mean Energy column is the standard deviation.

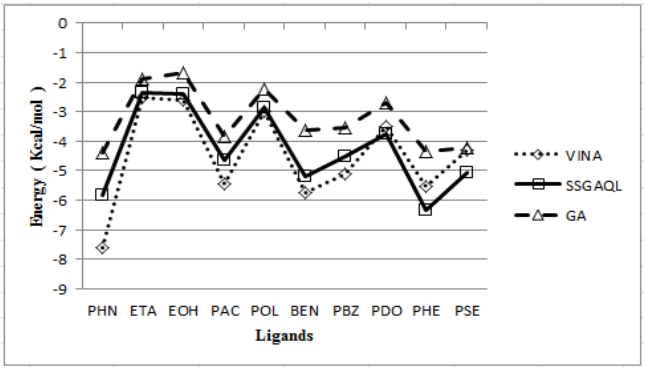
Table 3: Comparison of the lowest and mean energy found by VINA, SSGAQL and GA. Test cases where SSGAQL, GA found better energy from VINA are highlighted.

	Lowest Energy			Mean Energy		
	V	SG	GA	V	SG	GA
PHN	-7.63	-6.30	-5.45	-7.62 (0.01)	-5.84 (0.41)	-4.413 (0.62)
ETA	-2.55	-2.46	-2.47	-2.51 (0.06)	-2.34 (0.07)	-1.875 (0.37)
EOH	-2.65	-2.58	-2.13	-2.61 (0.04)	-2.42 (0.11)	-1.678 (0.26)
PAC	-5.53	-4.99	-4.13	-5.45 (0.09)	-4.65 (0.36)	-3.823 (0.27)
POL	-3.11	-3.00	-2.78	-3.01 (0.06)	-2.85 (0.08)	-2.223 (0.36)
BEN	-5.82	-5.4	-5.27	-5.76 (0.01)	-5.21 (0.25)	-3.63 (0.47)
PBZ	**-5.11**	**-5.22**	-4.08	-5.10 (0.01)	-4.53 (0.25)	-3.532 (0.47)
PDO	**-3.60**	**-3.90**	**-3.79**	**-3.52 (0.07)**	**-3.76 (0.15)**	-2.712 (0.49)
PHE	**-6.01**	**-6.74**	-5.35	**-5.55 (0.32)**	**-6.35 (0.32)**	-4.353 (0.69)
PSE	**-4.91**	**-5.39**	-4.82	**-4.32 (0.33)**	**-5.05 (0.21)**	-4.203 (0.51)

Figure 5 visualizes these comparisons by arranging the 10 ligands with complexity from small to large in the direction from left to right on the horizontal axis. It is evident from Figure 5(a) that SSGAQL found better structure than VINA and GA with lowest energy for highly flexible ligands highlighted in the table. In general, VINA finds the lowest energy for very small ligand with small number of torsion. There is no remarkable difference in docking ETA, EOH and POL. The more flexible the ligand, the more SSGAQL outperformed VINA such as in docking PBZ, BEN, PDO, PHE and PSE. GA's lowest energy line is poorer than the others, except in docking PDO hence surpassing VINA. It can be seen from Figure 5(b) that SSGAQL found significantly lower mean energy in docking highly flexible ligands such as the ones highlighted in the table. Overall, in 10 times docking for each ligand, VINA always find low energy in ligands in the category of small and medium number of torsions. However, there is no remarkable difference between SSGAQL and VINA as their mean energy in docking ETA, EOH, POL and PDO. SSGAQL produced lowest mean energy when docking ligands with high number of torsions such as PDO, PHE and PSE.



(a)



(b)

Fig. 5: Comparison on (a) the lowest energy and (b) the mean energy

QLTP agent in SSGAQL could be the factor in its performance on most small and largest ligands. The small step made by QLTP in traversing the search space is suitable for docking small ligands in the small but deeply located active site of thermolysin. However, SSGAQL could not find better conformation than VINA through this mechanism. Unlike VINA, the QLTP agent in SSGAQL does not continuously 'threading' its way to the minima, but does this in combination of small fixed size steps and neighbourhood jumps. SSGAQL might have simply step over the global optimum because of the fixed size steps and the quadrant jumps. However, due to the same reasons, SSGAQL outperformed VINA in docking ligands with high number of torsions. VINA simply cannot thread huge search space, but SSGAQL elegantly jumps into more promising regions in the search space promising good rewards.

5. Conclusion

In this study, a machine learning approach to solve protein docking problem was adopted for both fast global search and local search aiming for accuracy and refinement of the solution. A multistage genetic algorithm and the Q- learning agent called SSGAQL was presented. The proposed method is made up of iterative steady-state genetic algorithm with dynamic range and a Q-learning agent implementing a modified version of Kanerva coding (QLTP) for state-action space generalization. SSGAQL was tested on thermolysin with ten candidate ligands. The results showed that SSGAQL gives better performance than traditional GA and AutoDock VINA's algorithm in docking highly flexible ligands.

Acknowledgement

This work has been supported by the Malaysia Ministry of Science, Technology and Innovation.

References

[1] Shoichet, B. K., McGovern, S. L., Wei, B., & Irwin, J. J. Lead discovery using molecular docking. Curr. Opin. Chem. Biol. 6(4): 439–446, 2002.

[2] Simonson, T., Archontis, G., & Karplus, M. Free energy simulations come of age: protein- ligand recognition. Acc. Chem. Res. 35(6), 430–437,2002.

[3] Chen, H.-M., Liu, B.-F., Huang, H.-L., Hwang, S.-F., & Ho, S.-Y. SODOCK: swarm optimization for highly flexible protein-ligand docking. J. Comput. Chem. 28(2): 612–623, 2007.

[4] Gardiner, E. J., Willett, P., & Artymiuk, P. J. Protein docking using a genetic algorithm. Proteins: Struct., Funct., Bioinf. 44(1): 44–56, 2001.

[5] Jones, G., Willett, P., Glen, R. C., Leach, A. R., & Taylor, R. Development and validation of a genetic algorithm for flexible docking. J. Mol. Biol. 267(3): 727–48, 1997.

[6] Thomsen, R. Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids. Biosystems. 72(1–2): 57–73, 2003.

[7] Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. J. Comput. Chem. 19(14): 1639-1662, 1998.

[8] Yang, J. and Kao, C. Flexible ligand docking using a robust evolutionary algorithm. J. Comput. Chem. 21(11): 988–998, 2000.

[9] Sutton, R. S., & Barto, A. G. Reinforcement learning: An introduction (Vol. 1). MIT press Cambridge, 1998.

[10] Watkins, C., & Dayan, P. Q-learning. Machine Learning 8(3-4): 279–292, 1992.

[11] Abdoos, M., Mozayani, N., Ana, L., & Bazzan, C. Hierarchical control of traffic signals using Q-learning with tile coding. Applied Intelligence 40(2): 201-213, 2014.

[12] Beom, H. R., & Cho, H. S.A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics 25(3): 464–477, 1995.

[13] Forbes, J. R. N. Reinforcement learning for autonomous vehicles. University of California, Berkeley, 2002.

[14] Kim, H. J., Jordan, M. I., Sastry, S., and Ng, A. Y. Autonomous helicopter flight via reinforcement learning. In: Advances in neural information processing systems. MIT 799-806, 2004.

[15] Wen, Z., O'Neill, D., & Maei, H. Optimal demand response using device-based reinforcement learning. IEEE Transactions on Smart Grid 6(5): 2312–2324, 2015.

[16] De Magalhães, C. S., Barbosa, H. J. C., and Dardenne, L. E. A genetic algorithm for the ligand-protein docking problem. Genet. Mol. Biol. 27: 605–610, 2004.

[17] Atilgan, E. and Hu, J. Efficient protein-ligand docking using sustainable evolutionary algorithms. In: 10th International Conference on Hybrid Intelligent Systems. 23 August 2010. IEEE 113–118, 2010.

[18] Ng, M. C. K., Fong, S., and Siu, S. W. I. PSOVina: The hybrid particle swarm optimization algorithm for protein-ligand docking. J. Bioinform. Comput. Biol. 13(03): 1541007, 2015.

[19] Trott, O. and Olson, A. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J. Comput. Chem. 31(2): 455–461, 2010.

[20] Allen, M., and Fritzsche, P. Reinforcement learning with adaptive Kanerva coding for Xpilot game AI. In: Congress of Evolutionary Computation (CEC). 5 Jun 2011. IEEE 1521–1528, 2011.

[21] Rahman, M. B. A., Jaafar, A. H., Basri, M., Rahman, R. N. Z. R. A., Salleh, A. B., and Wahab, H. A. Design of novel semisynethetic metalloenzyme from thermolysin. BMC Syst. Biol. 1(1): 1–2, 2007.

[22] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I.N. and Bourne, P. E. The Protein Data Bank. Nucleic Acids Research 28: 235-242, 2000.

[23] Rahman, M. B. A., Jaafar, A. H., Basri, M., Rahman, R. N. Z. R. A., and Salleh, A. B. Biomolecular Design and Receptor-Ligand Interaction of a Potential Industrial Biocatalsyt: A Thermostable Thermolysin-Phosphoeth-anolamine-Ca 2 Protein Complex. J. Adv. Catal. Sci. Technol. 1: 1–5, 2014.