An Evolutionary Algorithm with Variable-Length Chromosome for Multi-objective Minimalistic Attack

1st Chengyu Zhou
dept. of Computer Science
Dalian University of Technology
Dalian, China
z2444953564@gmail.com

2nd Yaqing Hou
dept. of Computer Science
Dalian University of Technology
Dalian, China
houyq@dlut.edu.cn

3rd Wenqiang Ma
dept. of Computer Science
Dalian University of Technology
Dalian, China
ranyi@mail.dlut.edu.cn

4th Hua Yu
dept. of Computer Science
Dalian University of Technology
Dalian, China
yhiccd@mail.dlut.edu.cn

5th Hongwei Ge
dept. of Computer Science
Dalian University of Technology
Dalian, China
gehw@dlut.edu.cn

***Abstract*—Evolutionary Algorithms (EAs) have acquired significant achievements in multi-objective optimization. Canonical EAs are mainly based on fixed-length chromosome. However, certain optimization problems require variable-length chromosome based EAs to solve. In this paper, our interest lies in solving the minimalistic attack problem which is formulated as a multi-objective optimization problem aiming to apply perturbations on the input (pictures) of well-trained deep reinforcement learning (DRL) policies. The objective is to mislead the DRL agent to alter its predictions. To achieve this, we propose a novel evolutionary algorithm with variable-length chromosome that dynamically adapts the chromosome length. Experiments show that the proposed algorithm converges more quickly and achieves better results than the baseline algorithm.**

***Index Terms*—multi-objective optimization problem, evolutionary algorithm, variable length chromosome, multi-objective minimalistic attack.**

I. Introduction

With the rapid development of Artificial Intelligence (AI), AI security has gained much concern, mainly on its interpretability and vulnerability [1]. For the latter concern, researchers conducted many experiments on existing AI models [2] [3], for example, in computer vision, they find that with some adversarial examples that are almost identical with the original one to human eyes, well-trained neural networks will alter its prediction from a temple to an ostrich. Another example lies in the field of deep reinforcement learning (DRL), where well-trained agents are easily fooled with adversarial attacks [4].

Adversarial attacks refer to imperceptible non-random perturbations applied to observations that can alter the network's predictions [5]. In the context of DRL, the main goal of adversarial attacks is to generate perturbations that are as little as possible and still be able to deceive DRL models like well-trained neural networks, which can be further captured by the term: *minimalistic attacks* [6]. The perturbations to be applied on the inputs are alterations on grey-level pixels since the input of DRL policy are certain frames that can be described as grey-level pixel matrices. The attack itself can be measured by two main parts: the number of attacked pixels and the degree to which the pixels are altered. For both parts, the fewer, the better. Intuitively, the two parts are exclusive. To conduct an attack, the more pixels are allowed to be attacked, and the more each pixel is allowed to be altered, the easier the attack to be successful. This condition renders the problem a multi-objective problem since it has two conflicting objectives to be optimized. In this paper, we define this problem as a multi-objective minimalistic attack problem since its goal is to conduct a successful attack with as few perturbations as possible, that is where **minimalistic** lies.

Multi-objective optimization (MOO) refers to the simultaneous minimization or maximization of multiple conflicting objective functions [7]. The primary goal of MOO is to acquire or approximate a set of trade-off or compromise solutions, which is commonly known as the Pareto set (PS). A more detailed definition will be elucidated in Section II. There are many ways to solve MOO problems, including the weighted sum method, physical programming, and evolutionary algorithms (EAs). As a population-based approach, EAs are capable of simultaneously exploring different regions of the search space

This work was supported in part by the National Key Research and Development Program of China (No. 2021ZD0112400), the National Natural Science Foundation of China under Grant 62372081, the Young Elite Scientists Sponsorship Program by CAST under Grant 2022QNRC001, the NSFC-Liaoning Province United Foundation under Grant U1908214, the 111 Project, No.D23006, and the Fundamental Research Funds for the Central Universities under grant DUT21TD107, DUT22ZD214.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00043

to find a diverse set of solutions. Therefore, they have been well acknowledged as one of the most popular heuristics for solving multi-objective optimization problems. Among existing methods, a classic and popular method is NSGA-II [8], which has been successfully applied to a wide range of research fields and achieves promising results.

To the best of our knowledge, there have not been any attempts to solve the multi-objective minimalistic attack. The problem requires the representation of perturbations that vary with the number of attacked pixels, which demands variable length chromosome (VLC) for solution encoding.

To make the first attempt at solving this problem, we draw inspiration from the setting of this work [6] and came up with the following baseline algorithm: set a maximum and a minimum length for the chromosome, and give it an initial length (at a certain length or at random). At the beginning of every generation, the chromosome length has a probability to go up or down by a fixed step size (1 for example) . By using mutation and crossover operators to change chromosome length, this method is able to generate solutions that can launch an attack.

The baseline algorithm differs from other EAs with VLC [9], which is intuitive and does generates solutions that can launch a attack, but it has two shortcomings:

1) If the step size is too big, it fails to converge and tends to provide diverging solutions incurring expensive and complex attack solutions.
2) If the step size is too small, it often gets stuck in the local optima.

In response to the challenges of the fixed step size for the multi-objective minimalistic attack problem, we propose a new evolutionary algorithm with variable length chromosome that dynamically adapts chromosome length based on the framework of NSGA-II [8]. The proposed approach converges more quickly than the original algorithm and acquires better results.

The organization of the rest of this paper is as follows: Section II presents some preliminaries, and Section III describes the overall framework of the proposed algorithm. Section IV analyzes the experimental results. Section V concludes the paper.

II. Background

This section presents some preliminaries of multi-objective optimization and gives a formal definition of the multi-objective minimalistic attack problem

A. Multi-objective Optimization Problem

The process of simultaneously optimizing a set of often conflicting objective functions is called multi-objective optimization or vector optimization [7]. Without loss of generality, we consider all objective functions as minimization functions. Given a decision space $X \subset R^D$, a multi-objective minimization problem can be defined as follows:

$$\begin{aligned} &minimize\ F(x) = [f_1(x), f_2(x), ..., f_k(x)]^T \\ &subject\ to\ x \in X \end{aligned} \tag{1}$$

where $f_1(x), f_2(x), ..., f_k(x)$ are k different minimization objectives. $F(x)$ is the overall objective vector, and $x \in X$ is a decision vector. For the sake of brevity, additional constraint conditions have been suppressed in Eq. (1).

The feasible objective space Z is defined as the set $\{F(x)|x \in X\}$. Each point in X maps to a point in Z, but the reverse may not be true. The goal of multi-objective optimization is to find a set of decision variables constituting the Pareto set. In order to demonstrate the concept of Pareto set, we need to introduce the notion of Pareto dominance first.

$Definition1$: A solution x_1 is said to dominate another solution x_2 if $f_i(x_1) \leq f_i(x_2)$ for all $i \in \{1, 2, ...m\}$, and $f_j(x_1) < f_j(x_2)$ for at least one $j \in \{1, 2, ...m\}$

$Definition2$: A solution x^* is said to be $Pareto\ optimal$, if there are no other solutions that can dominate it.

The set of all Pareto optimal solutions is called the Pareto set (PS), and the mapping of the Pareto set into the objective space is called the Pareto front (PF). The main goal of multi-objective optimization is to find a set of solutions that is as close to PS as possible while at the same time maintaining diversity so that decision-makers can pick what they want according to their preferences.

B. The Formulation of Multi-objective Minimalistic Attack

Artificial Intelligence (AI) has made significant advancements, but it has also introduced certain challenges, including AI security. An example of this is highlighted in [6]. Minimalistic attack is a kind of deep reinforcement learning (DRL) adversarial attack problems. Here, the difference between the concept of minimalistic attack and sparse and imperceivable adversarial attack [10] needs to be clarified.

Sparse and imperceivable adversarial attack refers to the attack of well-trained neural networks that classify samples. On the other hand, minimalistic attack refers to the attack of well-trained DRL policies, which accomodates the three key settings:

(1) black-box policy access: where the attacker only has access to the input (state) and output (action probability) of an RL policy;

(2) fractional-state adversary: where only several pixels are perturbed, with the extreme case being a single-pixel adversary;

(3) tactically-chanced attack: where only significant frames are tactically chosen to be attacked.

The main objective of the attack is to deceive DRL policies by perturbing pixels in selected keyframes. For each keyframe, the goal is to maximize the discrepancy between frames before and after the attack.

In an DRL environment, the agent takes action a_t based on the state s_t and receives a reward r_t from the environment at time step t. Assuming a finite set of n available actions $a_t^1, a_t^2, ..., a_t^m$, the action probability distribution $\pi(\cdot|s_t)$ over those n actions can be described as:

$$\begin{aligned} \pi(\cdot|s_t) &= [p(a_t^1), p(a_t^2), ..., p(a_t^m)], \\ s.t. &\sum_{j=1}^{m} p(a_t^j) = 1. \end{aligned} \tag{2}$$

Herein, $p(a_t^j)$ represents the probability that the agent chooses action a_t^j. As expected, the agent guided by a deterministic policy selects the action $o = \arg\max_j p(a_t^j)$. With this, the goal of minimalistic attacks is to maximize the discrepancy between action distributions before and after the attack, which can be formulated as follows:

$$\max_{\delta_t} \max_{j \neq o} \pi(\cdot|s_t + \delta_t)_j - \pi(\cdot|s_t + \delta_t)_o \tag{3}$$

Where δ_t represents the perturbation to be added to the original state s_t at time step t. Further, o and j represent the action taken by the trained agent before and after the attack and $\pi(\cdot|s_t)_j$ represents the probability that the agent chooses action j under the guidance of $\pi(\cdot|s_t)$. In the context of minimalistic attacks, δ_t is limited to perturb only a small fraction of the input state. The perturbation is composed of two parts: the number of attacked pixels and the attack intensity(measured by the pixel matrix difference before and after the attack). Notably, a successful attack is achieved when the fitness value in Eq. (3) is greater than 0, indicating that the agent has been deceived and takes a different action.

Given the above introduction of our attack problem, we define the objectives as follow:

$$\begin{aligned} &minimize\ f_1 = n, \\ &minimize\ f_2 = ||M_a - M_b||_F \\ &subject\ to\ Eq.\ (3) > 0\ and\ n\ \in Z_+ \end{aligned} \tag{4}$$

Here, n is the number of attacked pixels. M_b is the matrix of pixel grey level before the attack, and M_a is the matrix after the attack. The second objective is the Frobenius norm of matrix difference before and after the attack. Both objectives affect the intensity of minimalistic attacks (attack intensity).

III. Proposed Method

In this section, we discuss the detailed realization of our proposed multi-objective minimalistic attack optimization method that adapts the chromosome length dynamically. The goal hers is to enhance the exploration capabilities and attack efficiency of evolutionary methods without losing the ability to exploit. The following will provide detailed insights into our method, including the overall framework of the proposed algorithm.

A. Dynamic Adaptation of Chromosome Length

This part describes how the proposed algorithm adapts the chromosome length dynamically.

As we mentioned in the introduction, the existing method takes a different approach that differs from other VLC-based EAs [9]. It overemphasizes the exploitation during the evolution process, thus lacking exploration in the search space. This is undesirable as the optimization methods are expected to converge faster and by exploring more useful solutions through the search space. In this work, our proposed algorithm dynamically adapts the chromosome length, which renders a quicker convergence trend and achieves better results in the experiments conducted later. The main idea of the proposed approach is to spread all individuals across the entire search space, and then conduct detailed exploitation to find optimal solutions. Readers can refer to Algorithm 1 for more details.

Algorithm 1 Chromosome Length Adaptation

1: **Input:** max length of chromosome max_len, current length of chromosome cur_len and current generation number g.
2: **while** $stopping\ condition\ not\ satisfied$ **do**
3: **if** $g <= round(\sqrt{max\_len})$ **then**
4: step = $round(\sqrt{max\_len})$;
5: **if** $rand(0,1) >= 0.5$ **then**
6: cur_len = cur_len + step;
7: **else**
8: cur_len = cur_len - step;
9: **end if**
10: **else**
11: step = $round(\sqrt{cur\_len})$
12: **if** $rand(0,1) >= 0.5$ **then**
13: cur_len = cur_len + step;
14: **else**
15: cur_len = cur_len - step;
16: **end if**
17: **end if**
18: **end while**

Here, max_len is the maximum chromosome length. cur_len is the current chromosome length. If the chromosome length exceeds the boundary length, it will take the value of the boundary. In the first $round(\sqrt{max\_len})$ generation, the population will take big steps to spread out to explore the entire search space (lines 4-9). In the rest generations, it will adjust its step based on its current chromosome length $round(\sqrt{cur\_len})$ so as to conduct detailed exploitation (lines 11-16). By doing so, it dynamically adjusts the chromosome length.

B. Overall Framework of the Proposed Algorithm

This part presents the overall framework of the proposed algorithm. The proposed algorithm takes the framework of NSGA-II [8] as the basic multi-objective optimizer. Details of the entire framework are presented in the following Algorithm 2.

Algorithm 2 Proposed Algorithm

1: **Input**: max generation G, population size N, and the objectives of minimalistic attacks $F(x)$.
2: Initialize population P_0 with size N;
3: Evaluate population P_0 using $F(x)$;
4: **for** $g = 1\ to\ G$ **do**
5: Apply crossover and mutation to generate offspring O;
6: Call Algorithm 1;
7: Evaluate the whole population $P_g \cup O$ using $F(x)$;
8: Conduct non-dominated sort and Compute crowding distance for $P_g \cup O$;
9: Select next generation P_{g+1} from $P_g \cup O$;
10: **end for**

In this setting, let G represent the maximum generation, N denote the population size, and $F(x)$ stand for the objectives of the minimalistic attack problem. At the very beginning, initialize the population P_0 with population size N and evaluate each individual using the objective function $F(x)$ (lines 2-3). In every generation, the current population reproduces new offspring O through crossover and mutation, and then dynamically adjusts the chromosome length using Algorithm 1 (lines 5-6). Subsequently, the entire population undergoes non-dominated sorting and their crowding distances are computed (line 8). The population of the next generation is selected based on the two criteria (line 9).

Many realistic multi-objective problems come with constraints, which is also true for multi-objective minimalistic attacks. In order to conduct a successful attack, the algorithm has to generate perturbations that satisfy Eq. (3) > 0. During the application of genetic algorithms like NSGA-II to optimization problems, occurrences of constraint violations are frequent. In our proposed algorithm, we incorporate the same constraint handling technique in [8].

IV. Experimental Results

This section analyzes the experimental results of the baseline algorithm and the proposed method on multi-objective minimalistic attack problem.

A. Experimental Setup

In our experiments, we use actor critic using kronecker-factored trust Rregion (ACKTR) [11] as our reinforcement learning policy on two atari games: Qbert and Seaquest [12]. In order to facilitate the implementation of the algorithm details, on the first objective, we specify that the maximum number of attacked pixels is 100, the minimum number is 1 and the initial number is 80. On the second objective, we encode every pixel attacked in three consecutive positions on the chromosome, each representing the x-coordinate, y-coordinate, and the magnitude of alteration applied on the pixel in grey level number. In the first generation, the actual length of the chromosome is still 100 * 3 (3 consecutive positions for one attack pixel) but only the first 80 * 3 genes can express its phenotype. The baseline algorithm mentioned in the graph is the existing algorithm introduced in the introduction, which takes the step size of 1 for ease of use. The proposed algorithm mentioned in the graph is the algorithm discussed in section III. Under such settings, we run the two algorithms respectively. More details settings of hyper-parameters are listed as follows:

1) Representation:
   a) Objective 1: discrete.
   b) Objective 2: real-valued coded.
2) Max chromosome length: 300.
3) Min chromosome length: 3.
4) Initial chromosome length: 240.
5) Population size: 10.
6) Max generation: 50.
7) Repetition: 30.
8) Evolutionary operators:
   a) Single point crossover. [13].
   b) mutation when reproducing next population: Gaussian mutation with probability $p_m = 1/d$ (d is the dimensionality of the target optimization problem) and distribution index η_m=10.

B. Performance Metrics

Metric 1: average values of objectives 1 and 2 acquired by the two algorithms at every 10 generations.

Metric 2: hypervolume (HV) [14]. Let $y^* = (y_1^*,...,y_m^*)$ be a reference point in the objective space that is dominated by all Pareto optimal solutions. Let P be the approximation to the PF gained by the algorithm. The HV value of P with regard to y^* is the volume of the region which is dominated by P and dominates y^*.

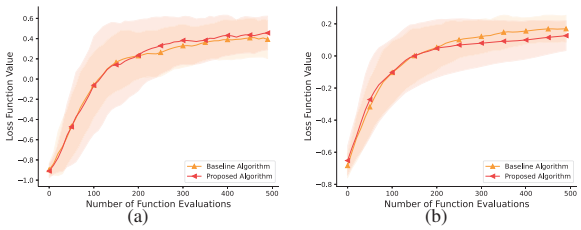

Fig. 1. Convergence trend of discrepancy Eq. (3) for (a) Qbert and (b) Seaquest

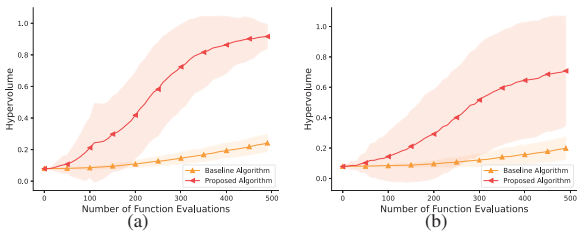

Fig. 2. Convergence trend of HV for (a) Qbert and (b) Seaquest

C. Results Analysis

Fig. 1 shows the discrepancy before and after the attack, i.e., the result of Eq. (3). Although both algorithms launch a successful attack (Eq. (3) > 0) at almost the same generation, we can see from the two tables that they launch the attack with different objective values, which means the proposed algorithm launches a successful attack at a smaller attack intensity. This further demonstrates the effectiveness of our proposed algorithm.

Table. I and Table. II describe the experimental results of metric 1. We can see that both algorithms improve their acquired solutions (with less attack intensity), but the proposed algorithm converges more quickly because of its dynamic adjustment of chromosome length, which indicates that the

proposed algorithm takes smaller attack intensity and still launches a successful attack to the RL policy.

In terms of the HV values, Fig 2 shows that the proposed algorithm grows much quicker than the baseline algorithm due to the fact that it explores the search space more swiftly and therefore reaches global optima with expedition. This is also because the proposed algorithm uses the rest generation to exploit into details of the search space, thereby launching a successful attack at an early stage.

TABLE I
RESULTS **(OBJECTIVE1 VALUE, OBJECTIVE2 VALUE)** OF BASELINE AND PROPOSED ALGORITHM FOR QBERT ENVIRONMENT AT EVERY 10 GENERATIONS. THE BEST ARE SHOWN IN **BOLD**.

Algorithm	(f_1, f_2) at gen 10	(f_1, f_2) at gen 20	(f_1, f_2) at gen 30	(f_1, f_2) at gen 40	(f_1, f_2) at gen 50
Baseline	(79, 1305)	(73, 1260)	(65, 1193)	(57, 1121)	(49, 1025)
Proposed	**(66, 1140)**	**(34, 747)**	**(15, 368)**	**(7, 238)**	**(2, 155)**

TABLE II
RESULTS **(OBJECTIVE1 VALUE, OBJECTIVE2 VALUE)** OF BASELINE AND PROPOSED ALGORITHM FOR SEAQUEST ENVIRONMENT AT EVERY 10 GENERATIONS. THE BEST ARE SHOWN IN **BOLD**.

Algorithm	(f_1, f_2) at gen 10	(f_1, f_2) at gen 20	(f_1, f_2) at gen 30	(f_1, f_2) at gen 40	(f_1, f_2) at gen 50
Baseline	(79, 1308)	(77, 1297)	(73, 1261)	(68, 1216)	(62, 1156)
Proposed	**(72, 1221)**	**(57, 1028)**	**(41, 795)**	**(29, 610)**	**(19, 460)**

V. CONCLUSION

This paper specifies an instance of the practical problem: the multi-objective minimalistic attack problem, which comes from the field of deep reinforcement learning (DRL). It is a bi-objective problem aiming to deceive well-trained DRL agent to change its predictions. It does so by perturbing the input of DRL policy, which is typically pictures described with pixel grey level value. The perturbations are composed of two parts: the number of attacked pixels and the Frobenius norm of the difference pixel matrix before and after the attack. This is a problem that requires EAs with variable-length chromosome to solve. The baseline algorithm changes its chromosome length with a fixed step size, which makes it perform badly. Therefore, this paper proposed a novel method that dynamically adapts the chromosome length, thereby expediting its convergence trend while not losing the ability to exploit already explored search space. Experiments showcase the effectiveness of the proposed algorithm and demonstrate its advantages.

REFERENCES

[1] Z. Kong, J. Xue, Y. Wang, L. Huang, Z. Niu, and F. Li, "A survey on adversarial attack in the age of artificial intelligence," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–22, 2021.

[2] D. Wang, W. Yao, T. Jiang, G. Tang, and X. Chen, "A survey on physical adversarial attack in computer vision," *arXiv preprint arXiv:2209.14262*, 2022.

[3] T. Long, Q. Gao, L. Xu, and Z. Zhou, "A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions," *Computers & Security*, p. 102847, 2022.

[4] K. Mo, W. Tang, J. Li, and X. Yuan, "Attacking deep reinforcement learning with decoupled adversarial policy," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 758–768, 2022.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[6] Q. Xinghua, Z. Sun, Y. Ong, A. Gupta, and P. Wei, "Minimalistic attacks: How little it takes to fool deep reinforcement learning policies," *IEEE Transactions on Cognitive and Developmental Systems*, vol. PP, pp. 1–1, 02 2020.

[7] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 201–221, 2022.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[9] B. J. Yu, "A genetic algorithm framework using variable length chromosomes for vehicle maneuver planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2022.

[10] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[11] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, and J. Ba, "Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation," *Advances in neural information processing systems*, vol. 30, 2017.

[12] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao, "Mastering atari games with limited data," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 25 476–25 488. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/d5eca8dc3820cad9fe56a3bafda65ca1-Paper.pdf

[13] P. Kora and P. Yadlapalli, "Crossover operators in genetic algorithms: A review," *International Journal of Computer Applications*, vol. 162, no. 10, 2017.

[14] A. P. Guerreiro, C. M. Fonseca, and L. Paquete, "The hypervolume indicator: Computational problems and algorithms," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–42, 2021.