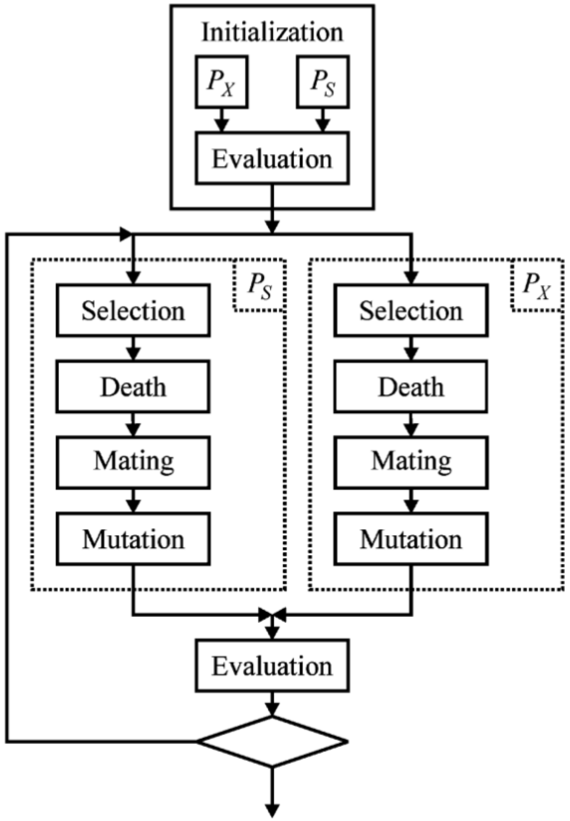
Algorithm 1

Coevolutionary Algorithm (CoEA)

1. 生成初始种群 P_1 P_2
2. 根据种群 P_1 P_2 相互竞争或者合作的结果，对个体进行评估
3. 根据评估结果对 P_1 中个体进行筛选和变异，生成下一代种群
4. 根据评估结果对 P_2 中个体进行筛选和变异，生成下一代种群
5. 回到步骤 2，直到满足终止条件



Algorithm 1

Coevolutionary Algorithm 在许多问题上已经有了成功地应用，包括 design of sorting[1], software patching[2]，problems in cyber security[3]，有很高的研究价值。

[1] W. Daniel Hillis. 1990. Co-evolving parasites improve simulated evolution as an optimization procedure. Physica D: Nonlinear Phenomena 42, 1 (June 1990), 228–234.

[2] Andrea Arcuri and Xin Yao. 2008. A novel co-evolutionary approach to au- tomatic software bug fixing. In 2008 IEEE Congress on Evolutionary Computation, 162–168.

[3] Una-May O'Reilly, Jamal Toutouh, Marcos Pertierra, Daniel Prado Sanchez, Dennis Garcia, Anthony Erb Luogo, Jonathan Kelly, and Erik Hemberg. 2020. Adversarial genetic programming for cyber security: a rising application domain where GP matters Genetic Programming and Evolvable Machines 21, 1- 2 (June 2020), 219–250.

Algorithm 1

但 Coevolutionary Algorithm 的理论分析十分困难，有以下难点[1]：

1. Over-specialisation: 某些种群只在部分场景中有好的表现，其余情况下表现很差
2. Cyclic behaviour: 算法会陷入循环，并不收敛
3. Evolutionary forgetting: 随着进化，之前表现好的个体会被淘汰

正是因为这些困难，在 CoEA 提出的 20 多年里，一直没有严格的理论分析。直到最近的一项研究 [2] 分析了 CoEA 在特定问题下的时间复杂度。

[1] Richard A. Watson and Jordan B. Pollack. 2001. Coevolutionary Dynamics in a Minimal Substrate. In Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO’01). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 702–709.
[2] Per Kristian Lehre. Runtime analysis of competitive co-evolutionary algorithms for maximin optimisation of a bilinear function. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’22, page 1408–1416.

Problem 1

Minimax Optimization Problems

优化目标：

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} g(x,y)$$

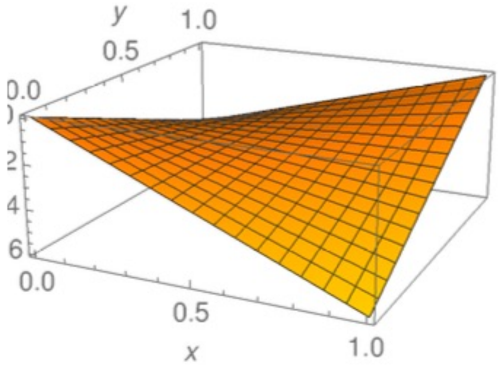
其中 x 代表解 (solution)，y 代表场景 (scenario)，g(x, y) 为问题的目标函数，表示解 x 在场景 y 中的代价

The Bilinear Problem

在 Bilinear Problem 中， x, y 都表示为 n 位 0/1 串，$\alpha,\beta\in(0,1)$ 为问题的两个参数。该问题的目标函数为 Bilinear Function:

$$\textsc{Bilinear}(x,y) := |y|(|x|-\beta n)-\alpha n|x|,$$

|x|，|y| 表示对应 0/1 串中 1 的数量

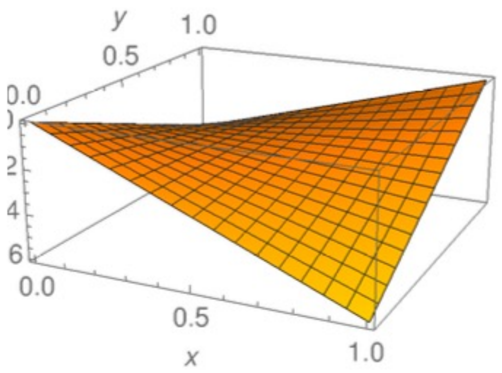


Bilinear 函数图像

Problem 1

[1] 中提出的 Bilinear problem 对于分析 CoEA 来说是一个很好的场景，函数形态简单清晰，而且之前提到的难点在这个场景中 CoEA 都会出现。

但 [1] 中分析的局限性也很明显，能够分析$\alpha < 1/5$ 的小情况，规避了难点 Cyclic behaviour 的分析。



Bilinear 函数图像

[1] Per Kristian Lehre. Runtime analysis of competitive co-evolutionary algorithms for maximin optimisation of a bilinear function. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’ 22, page 1408–1416.

Algorithm 2

在 [1] 中，第一次研究了 CoEA 在 Minimax Optimization 问题中的应用，并提出了经典的 CoEA 算法框架

Coevolutionary Algorithm

1. 随机生成初始种群 P, Q。 P 代表解 x 构成的种群，Q 代表场景 y 构成的种群。
2. 根据 Q 种群对 P 中的个体计算 fitness value $\mathrm{h}(x)$
3. 依据 P 种群对 Q 中的个体计算 fitness value $v(y)$
4. 根据 fitness value 对 P 中个体进行变异生成新个体，并进行筛选，生成新一代种群
5. 根据 fitness value 对 Q 中个体进行变异生成新个体，并进行筛选，生成新一代种群
6. 返回步骤 2，直到满足终止条件

其中：

$$h(x) = \max_{y \in Q} g(x, y).$$

$$v(y) = \min_{x \in P_t} g(x, y).$$

种群 P 偏好 fitness value 小的个体
种群 Q 偏好 fitness value 大的个体

[1] J. W. Herrmann. A Genetic Algorithm for Minimax Optimization Problems. In Angeline et. al., editors, Proc. 1999 Congress on Evolutionary Computation, pages 1099-1103, 1997.

Algorithm 2

我的研究给经典的算法框架补充了具体的筛选和进化机制，提出了右图的 CoEA：

这个 CoEA 在 Bilinear problem 中有很好的实际表现，并且我的研究给出了严格的时间分析。

除此之外还克服了之前研究的缺陷，首次解决了 Cyclic behaviour 这个难点，并且我的分析能够处理所有情况。

Algorithm 1: Co-evolutionary Algorithm (CoEA)

Input: Object function $g : \{0,1\}^n \times \{0,1\}^n \to \mathbb{R}$
Input: Population size $\lambda \in \mathbb{N}$
Output: $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} g(x, y)$

for $i \in [\lambda]$ **do**
  $P_0(i) \leftarrow \{1\}^n$;
  $Q_0(i) \leftarrow \{1\}^n$;
end
for $t \in \mathbb{N}$ *until algorithm converge* **do**
  for *individual* $x \in P_t$ **do**
    calculate $h(x) = \max_{y \in Q_t} g(x, y)$;
  end
  for *individual* $y \in Q_t$ **do**
    calculate $v(y) = \min_{x \in P_t} g(x, y)$;
  end
  let $x_{\max}$ and $x_{\min}$ be the individuals in P_t with the maximum and minimum $h(\cdot)$ value, respectively;
  let $y_{\max}$ and $y_{\min}$ be the individuals in Q_t with the maximum and minimum $v(\cdot)$ value, respectively;
  obtain x' by flipping exactly one bit in $x_{\min}$ uniformly at random;
  obtain y' by flipping exactly one bit in $y_{\max}$ uniformly at random;
  if $h(x') < h(x_{\min})$ **then**
    $P_{t+1} \leftarrow P_t \cup \{x'\} \setminus \{x_{\max}\}$;
  end
  else
    $P_{t+1} \leftarrow P_t$;
  end
  if $v(y') > v(y_{\max})$ **then**
    $Q_{t+1} \leftarrow Q_t \cup \{y'\} \setminus \{y_{\min}\}$;
  end
  else
    $Q_{t+1} \leftarrow Q_t$;
  end
end

Definition 1

Bilinear 问题中目标函数的梯度为：

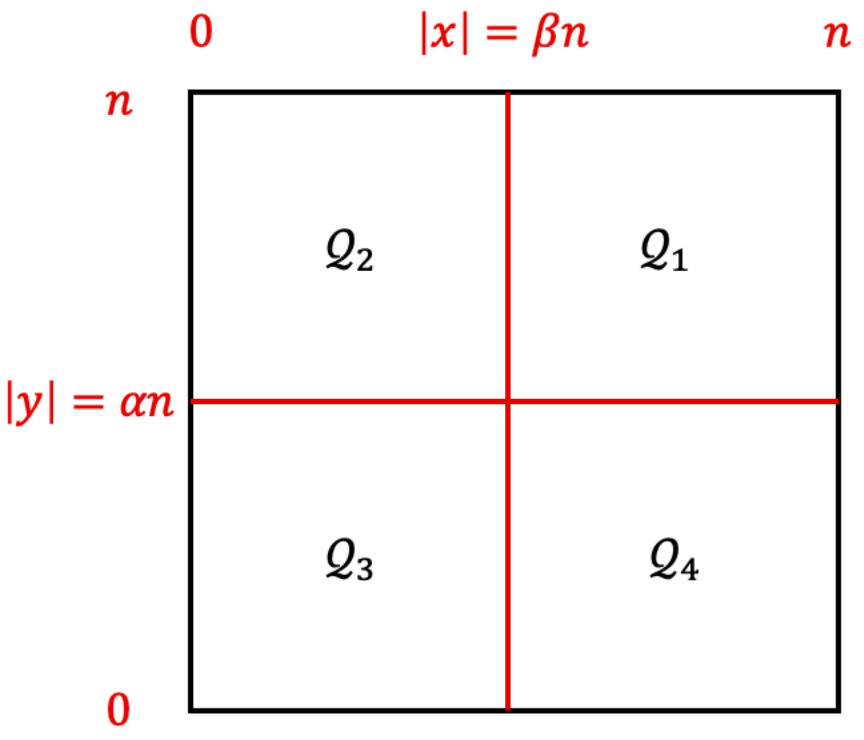
$$\nabla g = (|y| - \alpha n, |x| - \beta n),$$

根据梯度的方向，将整个搜索空间划分为四个象限。

以 $(x_{min}, y_{max}) \in Q_1$ 的情况为例，分析算法的行为：

$$x_{min}^{t+1} = \begin{cases} x_{min}^t - 1 & x_t' = x_{min}^t - 1 \\ x_{min}^t & x_t' = x_{min}^t + 1 \end{cases}$$

$$y_{max}^{t+1} = \begin{cases} y_{max}^t + 1 & y_t' = y_{max}^t + 1 \\ y_{max}^t & y_t' = y_{max}^t - 1 \end{cases}$$



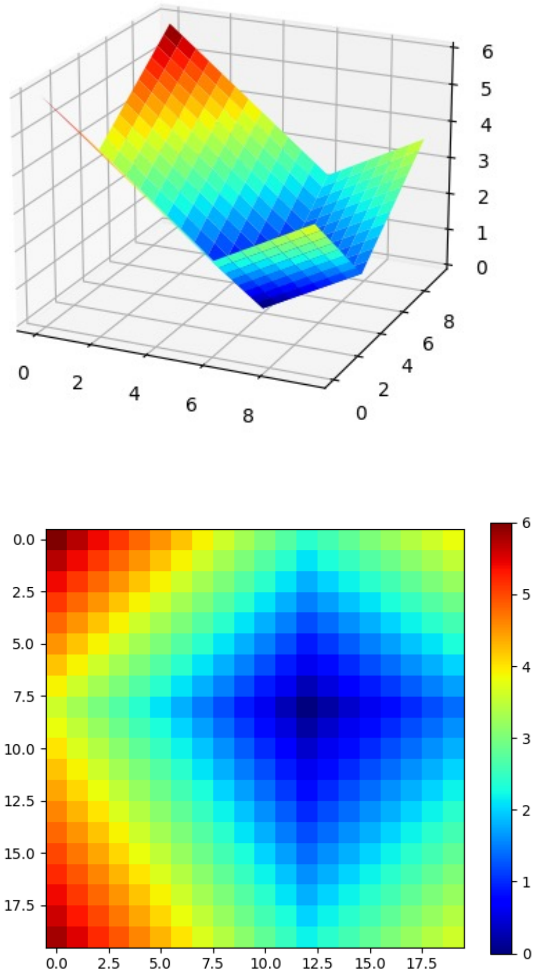
搜索空间的划分

Definition 2

为了追踪算法的运行状态，为个体和种群定义 potential function

$$
p(x,y)=\begin{cases}\alpha(x-n)-(\beta-1)y & x\geq\beta n \text{ and } y\geq\alpha n\\(\alpha-1)(x-n)-(\beta-1)(y-n) & x<\beta n \text{ and } y\geq\alpha n\\(\alpha-1)x-\beta(y-n) & x<\beta n \text{ and } y<\alpha n\\\alpha x-\beta y & x\geq\beta n \text{ and } y<\alpha n\end{cases}
$$

$$
p(P,Q)=p(x_{min},y_{max})
$$

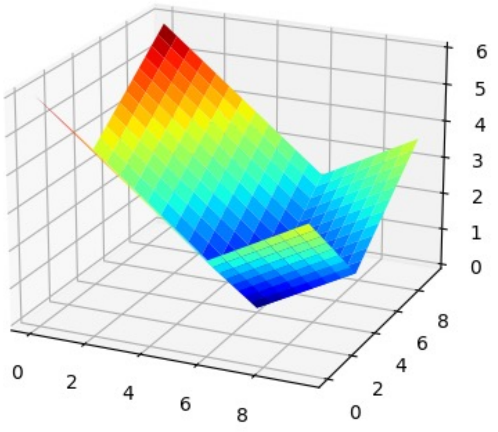


Potential function 函数图像

Theorem 1

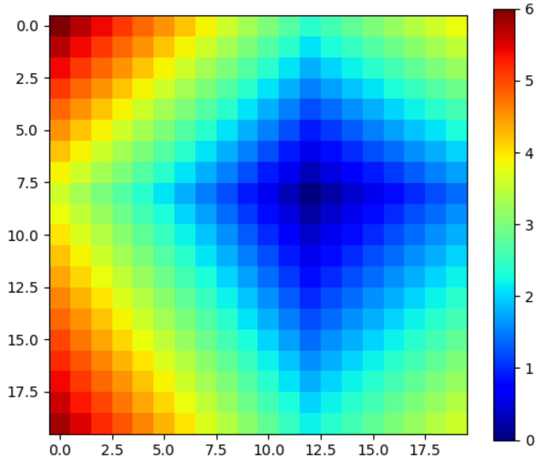
分析每次迭代 potential value 的期望减少量，提出 lemma

Lemma 3. *Assume after t iterations algorithm 1 is in phase 2. Let P, Q are the populations of Algorithm 1, α, β are the parametre of $Bilinear_{\alpha,\beta}$. Assume that for a positive constant c it holds $c\sqrt{n} \le \lambda \in poly(n)$. If $\omega = min\{\alpha, \beta, 1-\alpha, 1-\beta\} \ge n^{-\frac{1}{2}}$, then $\mathbb{E}(p_t - p_{t+1} | p_t = s) \ge \frac{s}{n}$*



应用 Multiplicative Drift 分析技巧，得出算法运行的期望时间

Theorem 1. *Assume that for a positive constant c it holds $c\sqrt{n} \le \lambda \in poly(n)$. Let α, β are the parameter of $Bilinear_{\alpha,\beta}$. Define $T = min\{\lambda t | (\beta n, \alpha n) \in (P_t, Q_t)\}$ where P_t and Q_t are the populations of Algorithm 1 applied to $Bilinear_{\alpha,\beta}$. There exists a constant c_0 that $\mathbb{E}[T] \le c_0 \lambda n \ln n$*



Potential function 函数图像