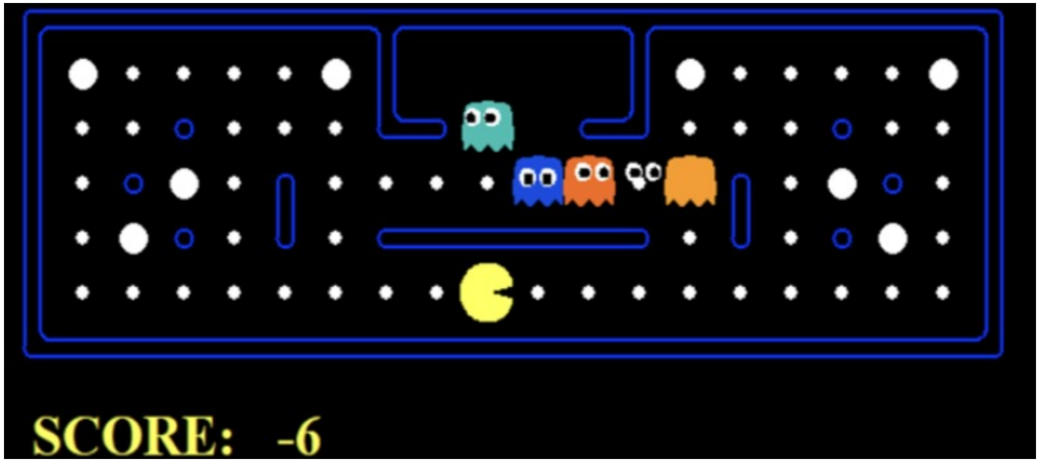
HSEA-2023 Assignment2

2023/10/24

背景

- 在第一次大作业中，同学们了解了基本的启发式搜索算法，并且设计了启发式算法用于玩Pacman游戏



- 最近几次课程，同学们了解了演化算法的基本概念

- 本次作业要求大家使用演化算法求解**子模优化**这一应用问题

任务背景

- 本次作业来自于演化计算重要国际会议 ACM GECCO的竞赛

- Evolutionary submodular optimization 演化子模优化

任务背景

子模优化：“边际效应递减”

$$\forall X \subseteq Y \subseteq V, v \notin Y\colon f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y);$$

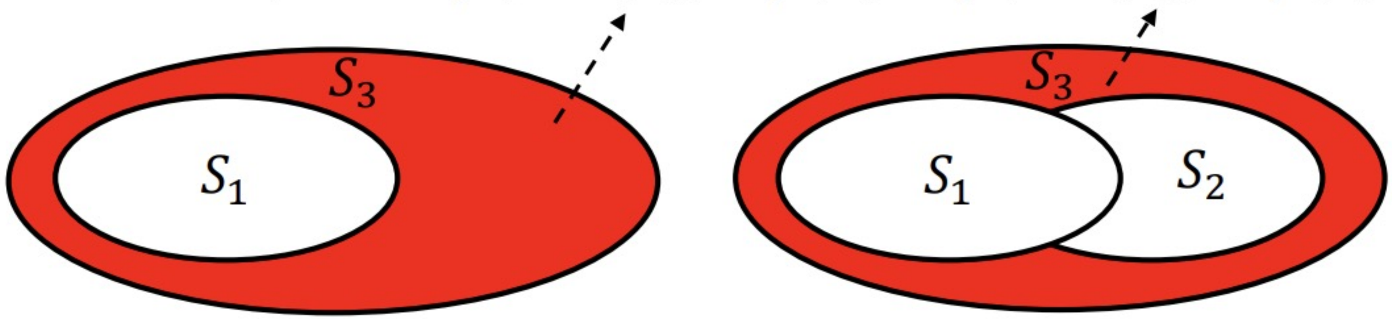
以一个经典的子模优化问题 Maximum coverage为例

Submodular: $\forall X \subseteq Y \subseteq V, v \notin Y\colon f(X \cup \{v\}) - f(X) \geq f(Y \cup \{v\}) - f(Y)$

$X = \{S_1\}$

$Y = \{S_1, S_2\}$

$v = S_3$



任务背景

Maximum Cut

Given an undirected weighted graph $G = (V, E, w)$ with weights $w\colon E \to \mathbb{R}_{\geq 0}$ on the edges, the goal is to select a set $V_1 \subseteq V$ such that the sum of the weight of edges between V_1 and $V_2 = V \setminus V_1$ is maximal.

解

For a given search point $\boldsymbol{x} \in \{0,1\}^n$ where $n = |V|$, we have $V_1(\boldsymbol{x}) = \{v_i \mid x_i = 1\}$ and $V_2(\boldsymbol{x}) = \{v_i \mid x_i = 0\}$. Let $C(\boldsymbol{x}) = \{e \in E \mid e \cap V_1(\boldsymbol{x}) \neq \emptyset \wedge e \cap V_2(\boldsymbol{x}) \neq \emptyset\}$ be the cut of a given search point $\boldsymbol{x}$. The goal is to maximize

$$f'(\boldsymbol{x}) = \sum_{e \in C(\boldsymbol{x})} w(e).$$

目标函数

- 参考资料
  - 问题说明
  - 竞赛主页

任务一 (60pts)

实现演化算法求解子模优化问题-Maximum Cut

- 你需要实现基本的演化算法
  - 定义解的表示（至少实现0-1编码，即binary representation）
  - 实现基本的演化算子（至少实现bit-wise mutation，uniform crossover）
  - 实现种群，自然选择

- 你需要汇报相应的实验结果（随机正则图，G1-G10，详见python framework）
  - 演化算法参数设置（种群大小，算子参数，迭代轮数）
  - 问题参数设置（规模等）
  - 如果参考了论文，请引用并标明
  - 画出以评估次数（或迭代轮数）为横轴，fitness为纵轴的曲线，以观察算法的效果

任务二 (20pts)

演化算法改进：实现并比较特定的演化算法（standard bit-wise mutation vs heavy-tailed mutation）

F. Neumann, A. Neumann, C. Qian, V.A. Do, J. de Nobel, D. Vermetten, S. S. Ahouei, F. Ye, H. Wang, T. Bäck (2023): Benchmarking Algorithms for Submodular Optimization Problems Using IOHProfiler. In: [CoRR abs/2302.01464].

- $(1+1)$ **EA**$_{>0}$: The $(1+1)$ EA$_{>0}$ using the standard bit mutation with a static mutation rate $p = 1/n$. The standard bit mutation samples ℓ, the number of distinct bits to be flipped, from a conditional binomial distribution $\mathrm{Bin}_{>0}(n, p)$.
- $(1+1)$ **fast genetic algorithm (fast GA)**: The $(1+1)$ fast GA differs from the $(1+1)$ EA by sampling ℓ from a power-law distribution with $\beta = 1.5$ [27]. The power-law distribution is a heavy-tailed distribution, and its probability of sampling large $\ell > 1$ is higher, compared to the standard bit mutation with $p = 1/n$.

B. Doerr, H. P. Le, R. Makhmara, and T. D. Nguyen, "Fast genetic algorithms," in Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017. ACM, 2017, pp. 777–784. [Online]. Available: https://doi.org/10.1145/3071178.3071301

Algorithm 1: The (1+1) EA with static mutation rate p for maximizing $f\colon \{0,1\}^n \to \mathbb{R}$.

Initialization: Sample $x \in \{0,1\}^n$ uniformly at random;
Optimization: **for** $t = 1, 2, 3, \ldots$ **do**
  Sample $y \in \{0,1\}^n$ by flipping each bit in x with probability p; //mutation step
  if $f(y) \geq f(x)$ **then** $x \leftarrow y$; //selection step;

Algorithm 2: The heavy-tailed mutation operator $\mathtt{fmut}_\beta$.

Input: $x \in \{0,1\}^n$
Output: $y \in \{0,1\}^n$ obtained from applying standard-bit mutation to x with mutation rate α/n, where α is chosen randomly according to $D^{\beta}_{n/2}$
$y \leftarrow x$;
Choose $\alpha \in [1..n/2]$ randomly according to $D^{\beta}_{n/2}$;
for $j = 1$ **to** n **do**
  if random$([0,1]) \cdot n \leq \alpha$ **then**
    $y_j \leftarrow 1 - y_j$;
return y

任务二 (20pts)

演化算法改进：实现并比较特定的演化算法（standard bit-wise mutation vs heavy-tailed mutation）

F. Neumann, A. Neumann, C. Qian, V.A. Do, J. de Nobel, D. Vermetten, S. S. Ahouei, F. Ye, H. Wang, T. Bäck (2023): Benchmarking Algorithms for Submodular Optimization Problems Using IOHProfiler. In: [CoRR abs/2302.01464].

- $(1+1)$ **EA$_{>0}$**: The $(1+1)$ EA$_{>0}$ using the standard bit mutation with a static mutation rate $p = 1/n$. The standard bit mutation samples ℓ, the number of distinct bits to be flipped, from a conditional binomial distribution $\mathrm{Bin}_{>0}(n, p)$.
- $(1+1)$ **fast genetic algorithm (fast GA)**: The $(1+1)$ fast GA differs from the $(1+1)$ EA by sampling ℓ from a power-law distribution with $\beta = 1.5$ [27]. The power-law distribution is a heavy-tailed distribution, and its probability of sampling large $\ell > 1$ is higher, compared to the standard bit mutation with $p = 1/n$.

B. Doerr, H. P. Le, R. Makhmara, and T. D. Nguyen, "Fast genetic algorithms," in Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017. ACM, 2017, pp. 777–784. [Online]. Available: https://doi.org/10.1145/3071178.3071301

The discrete power-law distribution $D^{\beta}_{n/2}$: Let $\beta > 1$ be a constant. Then the discrete power-law distribution $D^{\beta}_{n/2}$ on $[1..n/2]$ is defined as follows. If a random variable X follows the distribution $D^{\beta}_{n/2}$, then

$$\Pr[X = \alpha] = (C^{\beta}_{n/2})^{-1}\alpha^{-\beta}$$

for all $\alpha \in [1..n/2]$, where the normalization constant is $C^{\beta}_{n/2} := \sum_{i=1}^{n/2} i^{-\beta}$. Note that $C^{\beta}_{n/2}$ is asymptotically equal to $\zeta(\beta)$, the Riemann zeta function ζ evaluated at β. We have

$$\zeta(\beta) - \frac{\beta}{\beta - 1}\left(\frac{n}{2}\right)^{-\beta+1} \le C^{\beta}_{n/2} \le \zeta(\beta)$$

for all $\beta > 1$. As orientation, e.g., $\zeta(1.5) \approx 2.612$, $\zeta(2) \approx 1.645$, and $\zeta(3) = 1.202$ are some known values of the ζ function.

Algorithm 2: The heavy-tailed mutation operator $\mathtt{fmut}_{\beta}$.

```
Input: x ∈ {0,1}^n
Output: y ∈ {0,1}^n obtained from applying standard-bit mutation to
    x with mutation rate α/n, where α is chosen randomly according to
    D^β_{n/2}
y ← x;
Choose α ∈ [1..n/2] randomly according to D^β_{n/2};
for j = 1 to n do
    if random([0,1]) · n ≤ α then
        y_j ← 1 − y_j;
return y
```

任务二 (20pts)

演化算法改进：实现并比较特定的演化算法（standard bit-wise mutation vs heavy-tailed mutation）

F. Neumann, A. Neumann, C. Qian, V.A. Do, J. de Nobel, D. Vermetten, S. S. Ahouei, F. Ye, H. Wang, T. Bäck (2023): Benchmarking Algorithms for Submodular Optimization Problems Using IOHProfiler. In: [CoRR abs/2302.01464].

- $(1+1)$ **EA**$_{>0}$: The $(1+1)$ EA$_{>0}$ using the standard bit mutation with a static mutation rate $p = 1/n$. The standard bit mutation samples ℓ, the number of distinct bits to be flipped, from a conditional binomial distribution $\text{Bin}_{>0}(n, p)$.
- $(1+1)$ **fast genetic algorithm (fast GA)**: The $(1+1)$ fast GA differs from the $(1+1)$ EA by sampling ℓ from a power-law distribution with $\beta = 1.5$ [27]. The power-law distribution is a heavy-tailed distribution, and its probability of sampling large $\ell > 1$ is higher, compared to the standard bit mutation with $p = 1/n$.

- 你需要汇报相应的实验结果（随机正则图，G1-G10，详见python framework）
  - 演化算法参数设置（种群大小，算子参数，迭代轮数）
  - 问题参数设置（规模等）
  - 如果参考了论文，请引用并标明
  - 画出以评估次数（或迭代轮数）为横轴，fitness为纵轴的曲线，比较两种算法的效果

不限于使用（1+1）EA

任务三 (20pts)

演化算法改进：如何设计更适合Maximum Cut问题的演化算法?

- 可能的改进方向：
  - 编码方式
  - 演化算子改进
  - 参数优化
  - 多目标化

    (可参考Qian et al. Subset Selection by Pareto Optimization. Nips'15)
  - …

- 你需要汇报相应的实验结果（随机正则图，G1-G10，详见python framework）
  - 演化算法参数设置（种群大小，算子参数，迭代轮数）
  - 问题参数设置（规模等）
  - 如果参考了论文，请引用并标明
  - 画出以评估次数（或迭代轮数）为横轴，fitness为纵轴的曲线，以观察算法的效果，与任务一中的原始算法对比

作业提交与评分

- 你需要提交一份压缩文件，以“学号_姓名”的方式命名，如“201240001_张三.zip”
  - 文件中需要包含完整的项目代码和实验报告，在作业截止日期(11月27日23:59) 前发送到 shanghp@lamda.nju.edu.cn，邮件标题命名和压缩文件一致

- 作业的评分主要参考演化算法的实现、实验效果、实验报告
  - 本次作业旨在以子模优化为例让大家熟悉演化算法流程，因此一份体现逻辑清晰的实验报告十分重要
  - 延期提交的作业会有相应分数折扣，请按时提交

Thank you!

shanghp@lamda.nju.edu.cn