Last class

- Schema theorem
- Markov chain modeling
- Convergence
- Running time complexity
- Expectation and tail inequalities
- Example of running time analysis

Heuristic Search and Evolutionary Algorithms

Lecture 10: Running Time Analysis of EAs

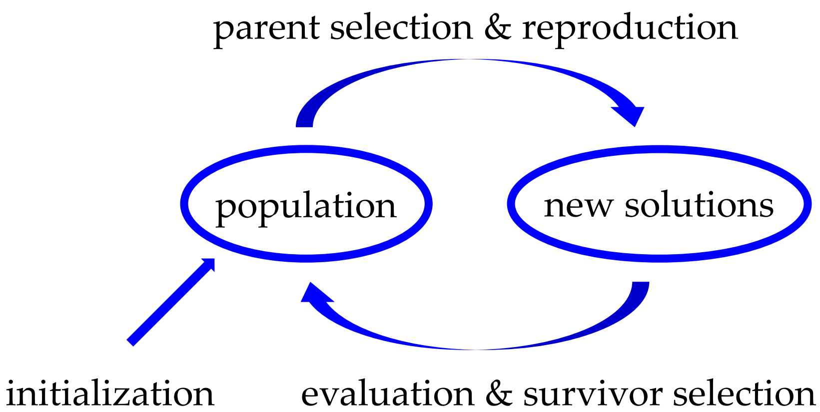
Chao Qian （钱超）

Associate Professor, Nanjing University, China

Email: qianc@nju.edu.cn
Homepage: http://www.lamda.nju.edu.cn/qianc/

Markov chain modeling

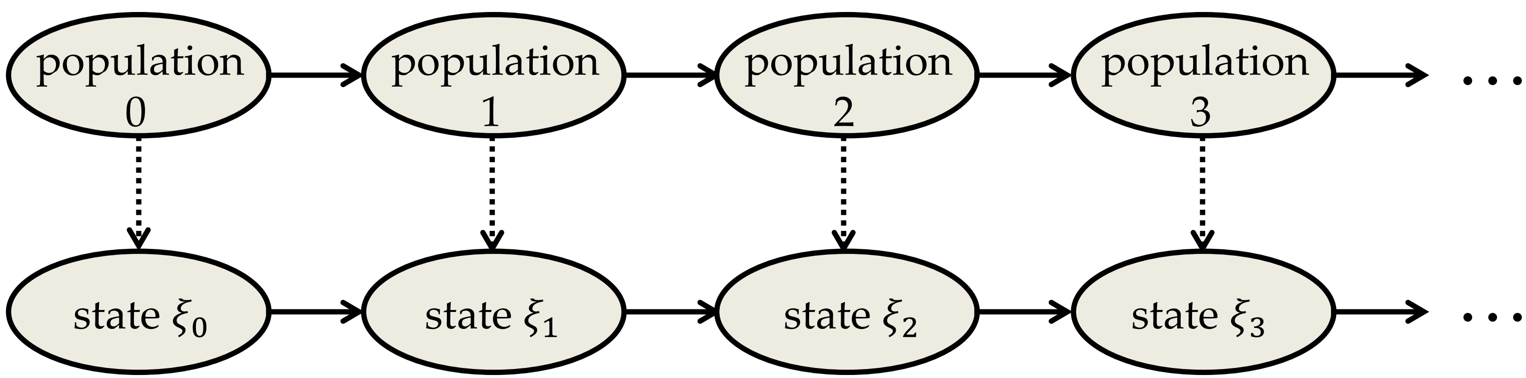
EA:



Running time analysis

$\tau = \min \{t \geq 0 \mid \xi_t \in \mathcal{X}^*\}$

a random variable
- $\mathrm{E}[\tau]$
- $P(\tau \leq T)$



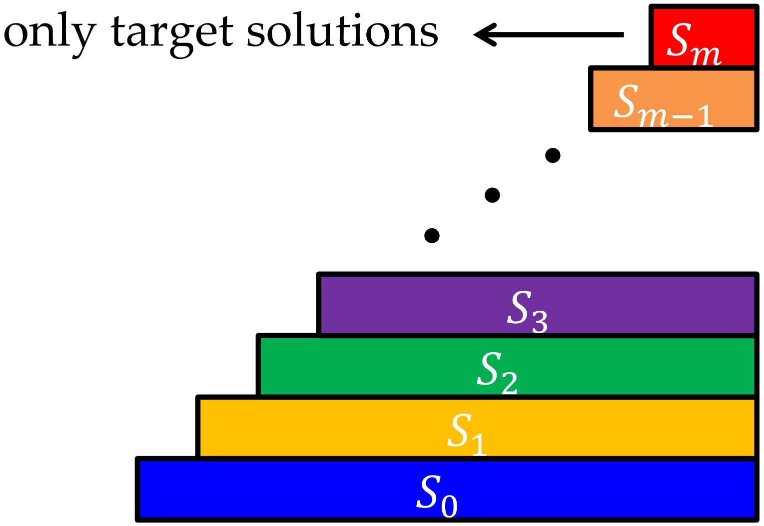
Markov chain: $P(\xi_t | \xi_{t-1}, \ldots, \xi_1, \xi_0) = P(\xi_t | \xi_{t-1})$

Fitness level method

The basic idea [Droste et al., TCS'02]:

1. Divide the solution space S into $m+1$ subspaces $S_0, S_1, \dots, S_m$

- $\forall i \neq j: S_i \cap S_j = \emptyset,\ \bigcup_{i=0}^{m} S_i = S$
- $\forall i < j, x \in S_i, y \in S_j: f(x) < f(y)$



Fitness level method

The basic idea [Droste et al., TCS'02]:

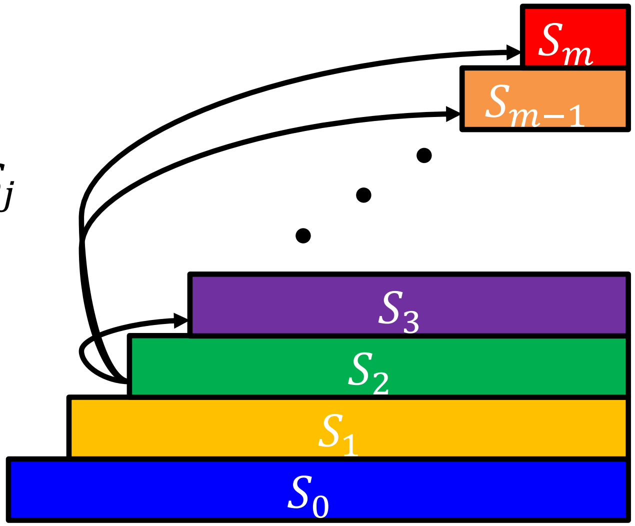
1. Divide the solution space S into $m+1$ subspaces $S_0, S_1, \dots, S_m$

- $\forall i \neq j: S_i \cap S_j = \emptyset,\ \bigcup_{i=0}^{m} S_i = S$
- $\forall i < j, x \in S_i, y \in S_j: f(x) < f(y)$

2. Bounds on the probability of leaving S_i to higher S_j

- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j | \xi_t \in S_i) \geq v_i$
- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j | \xi_t \in S_i) \leq u_i$

The best solution in ξ_t belongs to S_i

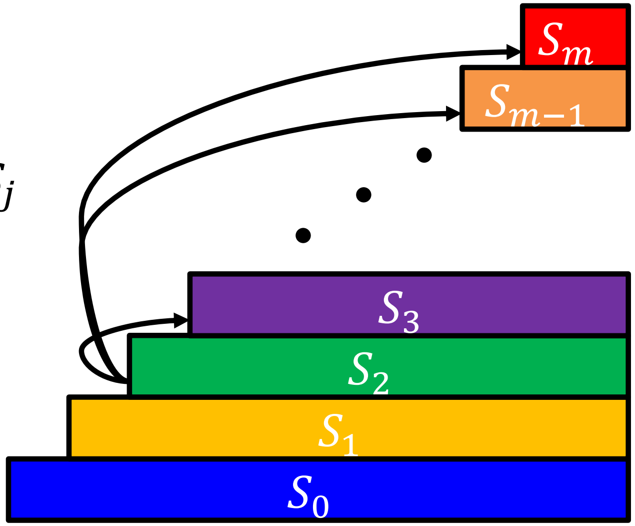


Fitness level method

The basic idea [Droste et al., TCS'02]:

1. Divide the solution space S into $m+1$ subspaces $S_0, S_1, \ldots, S_m$

- $\forall i \neq j: S_i \cap S_j = \emptyset,\ \bigcup_{i=0}^{m} S_i = S$
- $\forall i < j, x \in S_i, y \in S_j: f(x) < f(y)$



2. Bounds on the probability of leaving S_i to higher S_j

- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \geq v_i$
- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \leq u_i$

Expected running time

Upper bound: $\sum_{i=0}^{m-1} \pi_0(S_i) \cdot \sum_{j=i}^{m-1} \frac{1}{v_j}$

the initial distribution

Fitness level method

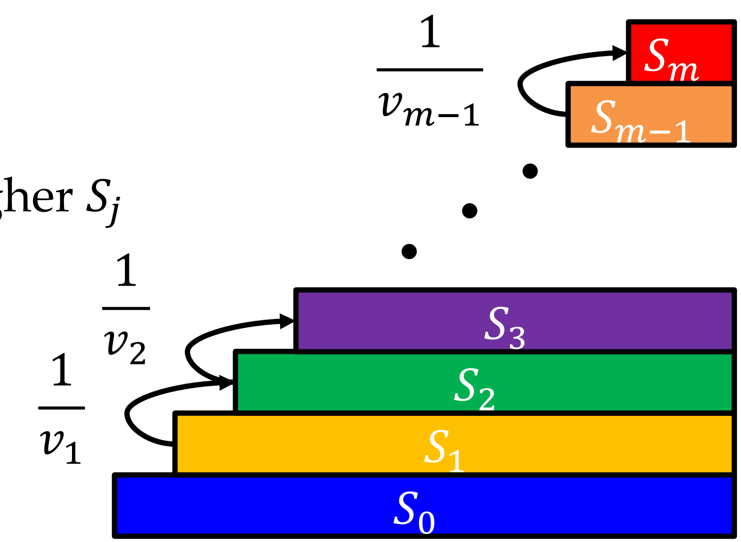
The basic idea [Droste et al., TCS'02]:

1. Divide the solution space S into $m+1$ subspaces $S_0, S_1, \ldots, S_m$

- $\forall i \neq j: S_i \cap S_j = \emptyset,\ \bigcup_{i=0}^{m} S_i = S$
- $\forall i < j, x \in S_i, y \in S_j: f(x) < f(y)$

2. Bounds on the probability of leaving S_i to higher S_j

- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \geq v_i$
- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \leq u_i$



Expected running time

Upper bound: $\sum_{i=0}^{m-1} \pi_0(S_i) \cdot \sum_{j=i}^{m-1} \frac{1}{v_j}$

the initial distribution

Fitness level method

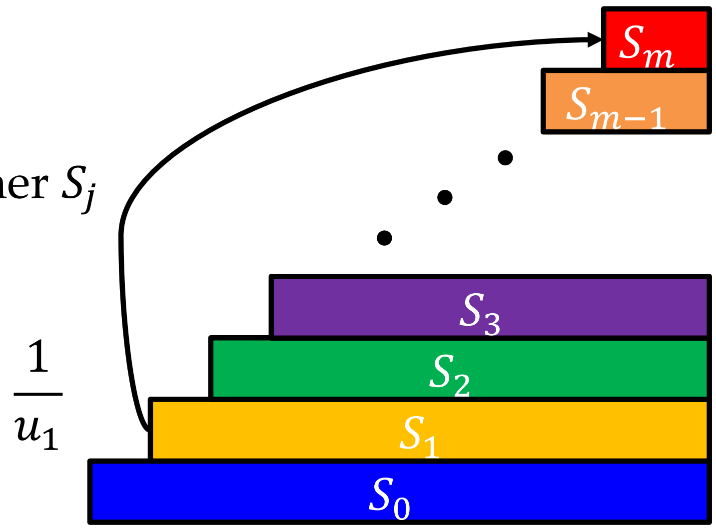
The basic idea [Droste et al., TCS'02]:

1. Divide the solution space S into $m+1$ subspaces $S_0, S_1, \ldots, S_m$

- $\forall i \neq j: S_i \cap S_j = \emptyset,\ \bigcup_{i=0}^{m} S_i = S$
- $\forall i < j, x \in S_i, y \in S_j: f(x) < f(y)$

2. Bounds on the probability of leaving S_i to higher S_j

- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \geq v_i$
- $P(\xi_{t+1} \in \bigcup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \leq u_i$



Expected running time

Upper bound: $\sum_{i=0}^{m-1} \pi_0(S_i) \cdot \sum_{j=i}^{m-1} \frac{1}{v_j}$

Lower bound: $\sum_{i=0}^{m-1} \pi_0(S_i) \cdot \frac{1}{u_i}$

Application illustration: (1+1)-EA for OneMax

(1+1)-EA:

Given a pseudo-Boolean function f:

1. $\boldsymbol{x} :=$ randomly selected from $\{0,1\}^n$.
2. Repeat until some termination criterion is met
3. $\quad \boldsymbol{x}' :=$ flip each bit of $\boldsymbol{x}$ with probability $1/n$.
4. $\quad$ if $f(\boldsymbol{x}') \geq f(\boldsymbol{x})$
5. $\quad\quad \boldsymbol{x} = \boldsymbol{x}'$.

Bit-wise mutation

OneMax:

$$max_{\boldsymbol{x}\in\{0,1\}^n} \sum_{i=1}^{n} x_i$$

Count the number of 1-bits

Theorem. [Droste et al., TCS'02] The expected running time of the (1+1)-EA solving the OneMax problem is $O(n \log n)$.

Proof

Theorem. [Droste et al., TCS'02] The expected running time of the (1+1)-EA solving the OneMax problem is $O(n \log n)$.

Main idea:

the number of 1-bits

- Divide the solution space $\{0,1\}^n$ into $S_{0,} S_1, \ldots, S_n$ with $S_i = \{x \in \{0,1\}^n \mid |x| = i\}$
- The probability of jumping to higher S_j from S_i is lower bounded by

$$P(\xi_{t+1} \in \cup_{j=i+1}^{m} S_j | \xi_t \in S_i) \geq \boxed{\frac{n-i}{n} \cdot (1-\frac{1}{n})^{n-1}}$$

keep the other bits unchanged

flip one of the $n-i$ 0-bits

$$v_j = \frac{n-j}{n} \cdot (1-\frac{1}{n})^{n-1}$$

Upper bound on the expected running time:

$$\sum_{i=0}^{n-1} \pi_0(S_i) \cdot \sum_{j=i}^{n-1} \frac{1}{v_j} \leq \sum_{j=0}^{n-1} \frac{1}{v_j}$$

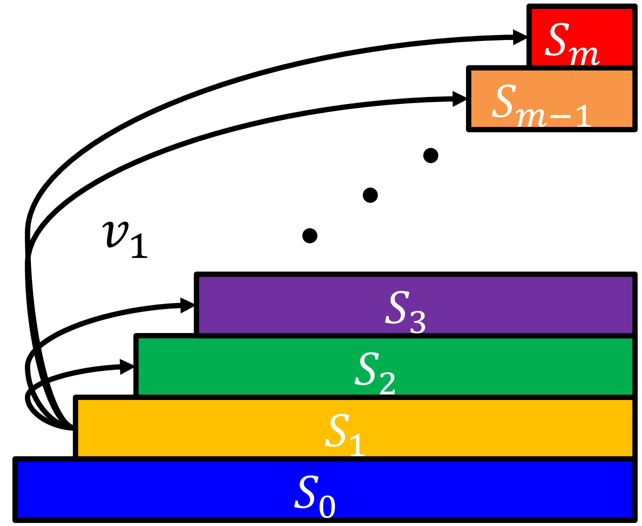
$$= \sum_{j=0}^{n-1} \frac{n}{n-j} \frac{1}{(1-\frac{1}{n})^{n-1}} \leq en \sum_{j=1}^{n} \frac{1}{j} \in O(n \log n)$$

Refined fitness level method

Original [Droste et al., TCS'02]:

$$P(\xi_{t+1} \in \cup_{j=i+1}^{m} S_j \mid \xi_t \in S_i) \geq v_i$$

Upper bound: $\sum_{i=0}^{m-1} \pi_0(S_i) \cdot \sum_{j=i}^{m-1} \frac{1}{v_j}$

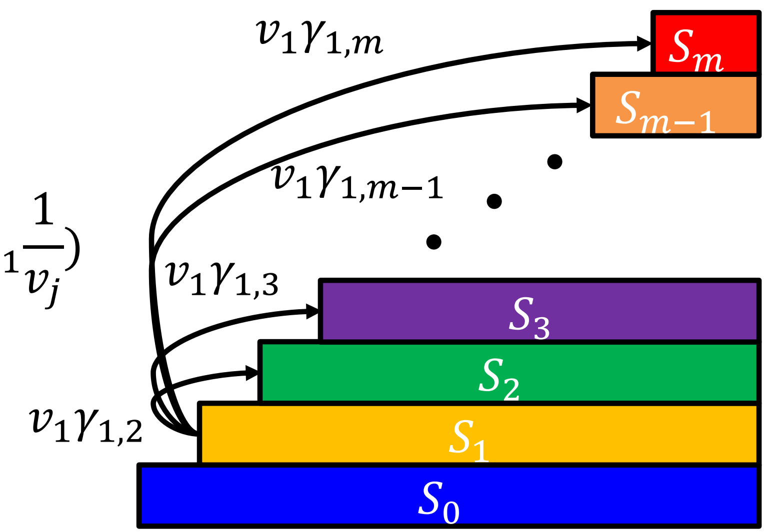


Refined [Sudholt, TEC'13]:

$$P(\xi_{t+1} \in S_j \mid \xi_t \in S_i) \geq v_i \cdot \gamma_{i,j}$$

$\sum_{j=i+1}^{m} \gamma_{i,j} = 1 \qquad \gamma_{i,j} \leq \chi \sum_{k=j}^{m} \gamma_{i,k}$

Upper bound: $\sum_{i=0}^{m-1} \pi_0(S_i) \cdot \left(\frac{1}{v_i} + \chi \sum_{j=i+1}^{m-1} \frac{1}{v_j}\right)$



Original is a specialization of refined with $\chi = 1$

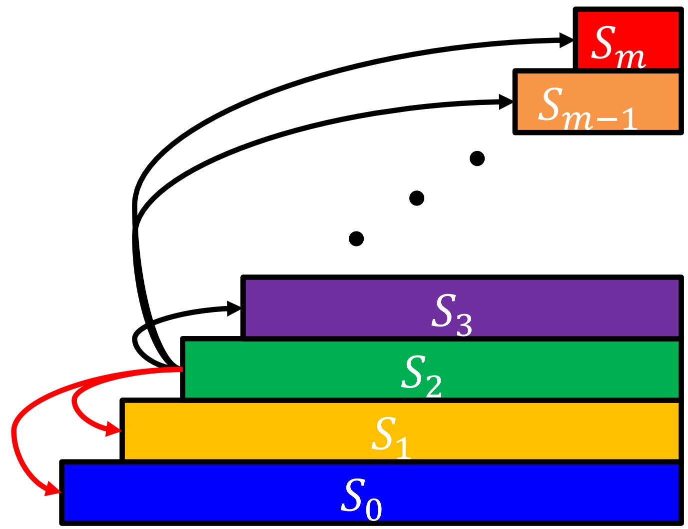
Similar for lower bound analysis

Refined fitness level method

The above two fitness level methods

[Droste et al., TCS'02; Sudholt, TEC'13]

- Only consider jumping to higher levels
- Proposed for elitist EAs



Fitness level method for non-elitist EAs

[Dang & Lehre, Algorithmica'16]

- allow jumping to lower levels

Theorem 8 *Given a function $f : \mathcal{X} \to \mathbb{R}$, and an f-based partition $(A_1, \ldots, A_{m+1})$, let T be the number of selection-variation steps until Algorithm 1 with a selection mechanism p_{sel} obtains an element in A_{m+1} for the first time. If there exist parameters $p_0, s_1, \ldots, s_m, s_* \in (0, 1]$, and $\gamma_0 \in (0, 1)$ and $\delta > 0$, such that*

(C1) $p_{\text{mut}}\left(y \in A_j^+ \mid x \in A_j\right) \geq s_j \geq s_*$ *for all* $j \in [m]$,

(C2) $p_{\text{mut}}\left(y \in A_j \cup A_j^+ \mid x \in A_j\right) \geq p_0$ *for all* $j \in [m]$,

(C3) $\beta(\gamma, P) p_0 \geq (1 + \delta)\gamma$ *for all* $P \in \mathcal{X}^\lambda$ *and* $\gamma \in (0, \gamma_0]$,

(C4) $\lambda \geq \frac{2}{a} \ln\left(\frac{16m}{ac\varepsilon s_*}\right)$ *with* $a = \frac{\delta^2 \gamma_0}{2(1+\delta)}$, $\varepsilon = \min\{\delta/2, 1/2\}$ *and* $c = \varepsilon^4/24$,

$\Rightarrow$

$$\mathbf{E}[T] \leq \frac{2}{c\varepsilon}\left(m\lambda(1 + \ln(1 + c\lambda)) + \frac{p_0}{(1+\delta)\gamma_0}\sum_{j=1}^{m}\frac{1}{s_j}\right)$$

Additive drift analysis

The basic idea [He & Yao, AIJ'01]:

1. Design a distance function $V(x): \mathcal{X} \to \mathbb{R}$ to measure the distance from a state $x \in \mathcal{X}$ to the target state space $\mathcal{X}^*$
   - $\forall x \in \mathcal{X} \setminus \mathcal{X}^*: V(x) > 0$
   - $\forall x \in \mathcal{X}^*: V(x) = 0$

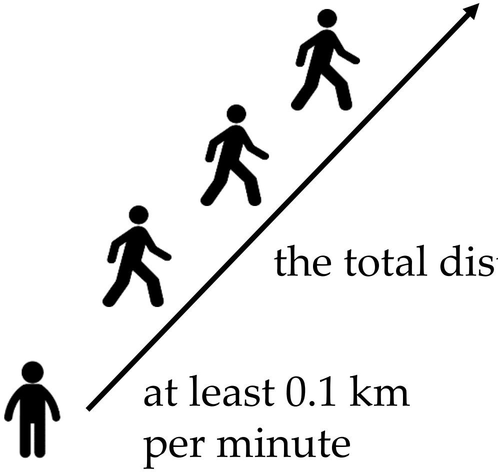
2. Bounds on the expected drift in one step
   - $\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq c_l$
   - $\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \leq c_u$

Expected running time

Upper bound: $\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \dfrac{V(x)}{c_l}$

the initial distribution

The hotel



the time $\leq \frac{3}{0.1} = 30$ minutes

Additive drift analysis

The basic idea [He & Yao, AIJ'01]:

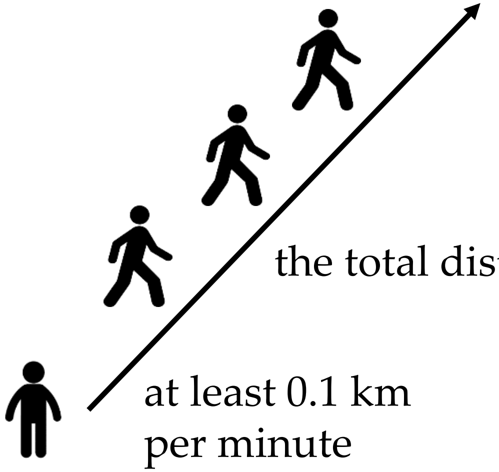
1. Design a distance function $V(x)\colon \mathcal{X} \to \mathbb{R}$ to measure the distance from a state $x \in \mathcal{X}$ to the target state space $\mathcal{X}^*$
   - $\forall x \in \mathcal{X} \setminus \mathcal{X}^*\colon V(x) > 0$
   - $\forall x \in \mathcal{X}^*\colon V(x) = 0$
2. Bounds on the expected drift in one step
   - $\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq c_l$
   - $\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \leq c_u$

Expected running time

Upper bound: $\sum_{x\in\mathcal{X}} \pi_0(x) \cdot \dfrac{V(x)}{c_l}$

Lower bound: $\sum_{x\in\mathcal{X}} \pi_0(x) \cdot \dfrac{V(x)}{c_u}$

The hotel



the total distance 3 km

at least 0.1 km per minute

the time $\leq \frac{3}{0.1} = 30$ minutes

Application illustration: (1+1)-EA for LeadingOnes

(1+1)-EA:

Given a pseudo-Boolean function f:

1. $\boldsymbol{x} :=$ randomly selected from $\{0,1\}^n$.
2. Repeat until some termination criterion is met
3. $\quad \boldsymbol{x}' :=$ flip each bit of $\boldsymbol{x}$ with probability $1/n$.
4. $\quad$ if $f(\boldsymbol{x}') \geq f(\boldsymbol{x})$
5. $\quad\quad \boldsymbol{x} = \boldsymbol{x}'$.

LeadingOnes:

$$max_{x \in \{0,1\}^n} \underbrace{\sum_{i=1}^{n} \prod_{j=1}^{i} x_j}_{LO(\boldsymbol{x})}$$

Count the number of consecutive 1-bits starting from the left

e.g., $f(11010) = 2$, $f(01111) = 0$

Theorem. [He & Yao, AIJ'01] The expected running time of the (1+1)-EA solving the LeadingOnes problem is $O(n^2)$.

Proof

Theorem. [He & Yao, AIJ'01] The expected running time of the (1+1)-EA solving the LeadingOnes problem is $O(n^2)$.

Main idea:

the number of leading 1-bits

- Design the distance function $\mathrm{V}(x) = n - LO(x)$
- The expected drift from a solution x with $LO(x) = i$:

$$\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x]$$
$$= \mathrm{E}[\, LO(\xi_{t+1}) - i \mid \xi_t = x] \geq 1 \cdot \frac{1}{n} \cdot (1 - \frac{1}{n})^i$$

keep the i leading 1-bits unchanged

$LO(x) = i \ \rightarrow \geq i + 1$

flip the first 0-bit

Proof

Theorem. [He & Yao, AIJ'01] The expected running time of the (1+1)-EA solving the LeadingOnes problem is $O(n^2)$.

Main idea:

the number of leading 1-bits

- Design the distance function $\mathrm{V}(x) = n - LO(x)$
- The expected drift from a solution x with $LO(x) = i$ is lower bounded by

$$\mathrm{E}[\,V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = x] \geq \boxed{1 \cdot \frac{1}{n} \cdot (1 - \frac{1}{n})^i}$$

keep the i leading 1-bits unchanged

$LO(x) = i \ \rightarrow \geq i + 1$

flip the first 0-bit

$$c_l = \frac{1}{en}$$

Upper bound on the expected running time:

$$\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{V(x)}{c_l} \leq \frac{n}{c_l}$$

$$= n/(\frac{1}{en}) = en^2 \in O(n^2)$$

Multiplicative drift analysis

Additive [He & Yao, AIJ'01]:

$\mathrm{E}[\,V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq c_l$ not depend on ξ_t

Upper bound: $\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{V(x)}{c_l}$

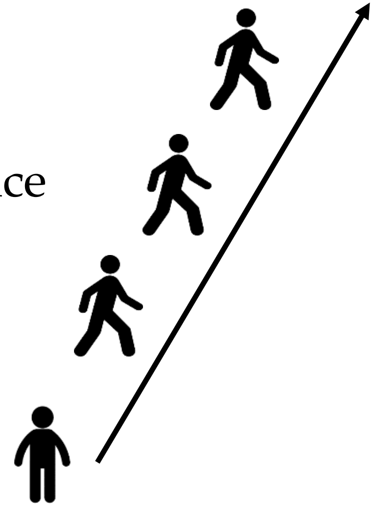
The hotel

Multiplicative [Doerr et al., Algorithmica'12]:

$\mathrm{E}[\,V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq \delta \cdot V(\xi_t)$ proportional to the current distance

Upper bound: $\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{1 + \ln\left(V(x)/V_{min}\right)}{\delta}$

$\min\{V(x) \mid V(x) > 0\}$



Multiplicative vs. Additive

For additive and multiplicative drift analysis, which one is stronger?

Theorem. [Doerr et al., Algorithmica'12] Multiplicative drift analysis can be proved by using additive drift analysis. Stronger

Multiplicative drift analysis:

$$\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq \delta \cdot V(\xi_t)$$

Upper bound:

$$\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{1 + \ln\left(V(x)/V_{min}\right)}{\delta}$$

Construct a distance function $U(x)$

Apply additive drift analysis

$$\mathrm{E}[U(\xi_t) - U(\xi_{t+1}) \mid \xi_t] \geq c_l$$

not depend on ξ_t

Proof

Multiplicative drift analysis:

$$\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq \delta \cdot V(\xi_t)$$

$\Rightarrow$

Upper bound:

$$\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{1 + \ln\left(V(x)/V_{min}\right)}{\delta}$$

Main idea:

- Design the distance function $\mathrm{U}(x) = \begin{cases} 0 & \text{if } V(x) = 0 \\ 1 + \ln(V(x)/V_{min}) & \text{Otherwise} \end{cases}$

- The drift from a state $\xi_t \notin \mathcal{X}^*$ (i.e., $\mathrm{V}(\xi_t) > 0$ or $\mathrm{U}(\xi_t) > 0$):

If $\xi_{t+1} \in \mathcal{X}^*$, then $\quad U(\xi_t) - U(\xi_{t+1}) = 1 + \ln(V(\xi_t)/V_{min}) \geq 1 = \dfrac{V(\xi_t) - V(\xi_{t+1})}{V(\xi_t)}$

If $\xi_{t+1} \notin \mathcal{X}^*$, then $\quad U(\xi_t) - U(\xi_{t+1}) = \ln(V(\xi_t)/V(\xi_{t+1})) \geq \dfrac{V(\xi_t) - V(\xi_{t+1})}{V(\xi_t)}$

Proof

$$\ln(V(\xi_t)/V(\xi_{t+1})) \geq \frac{V(\xi_t) - V(\xi_{t+1})}{V(\xi_t)}$$

$$\frac{u}{w} = 1 + \frac{u-w}{w} \leq e^{\frac{u-w}{w}}$$

$$\Downarrow$$

$$\ln\frac{u}{w} \leq \frac{u-w}{w}$$

$$\Downarrow$$

$$\ln\frac{w}{u} \geq \frac{w-u}{w}$$

Proof

Multiplicative drift analysis:

$$E[\,V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq \delta \cdot V(\xi_t)$$

⇨

Upper bound:

$$\sum_{x\in\mathcal{X}} \pi_0(x) \cdot \frac{1 + \ln\,(V(x)/V_{min})}{\delta}$$

Main idea:

- Design the distance function $\mathrm{U}(x) = \begin{cases} 0 & \text{if } V(x) = 0 \\ 1 + \ln(V(x)/V_{min}) & \text{Otherwise} \end{cases}$

- The drift from a state $\xi_t \notin \mathcal{X}^*$ (i.e., $\mathrm{V}(\xi_t) > 0$ or $\mathrm{U}(\xi_t) > 0$):

$$U(\xi_t) - U(\xi_{t+1}) \geq \frac{V(\xi_t) - V(\xi_{t+1})}{V(\xi_t)}$$

- The expected drift from a state $\xi_t \notin \mathcal{X}^*$ (i.e., $\mathrm{V}(\xi_t) > 0$ or $\mathrm{U}(\xi_t) > 0$):

$$E[U(\xi_t) - U(\xi_{t+1}) \mid \xi_t] \geq \frac{E[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]}{V(\xi_t)}$$

Proof

Multiplicative drift analysis:

$$\mathrm{E}[\,V(\xi_t) - V(\xi_{t+1}) \mid \xi_t] \geq \ \delta \cdot V(\xi_t)$$

Upper bound:

$$\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \frac{1 + \ln\left(V(x)/V_{min}\right)}{\delta}$$

Main idea:

- Design the distance function $\mathrm{U}(x) = \begin{cases} 0 & \text{if } V(x) = 0 \\ 1 + \ln(V(x)/V_{min}) & \text{Otherwise} \end{cases}$

- The expected drift from a state $\xi_t \notin \mathcal{X}^*$ (i.e., $\mathrm{V}(\xi_t) > 0$ or $\mathrm{U}(\xi_t) > 0$):

$$E[U(\xi_t) - U(\xi_{t+1}) \mid \xi_t] \geq \frac{E[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t]}{V(\xi_t)} \geq \delta$$

Additive drift analysis [He & Yao, AIJ'01]:

$$\mathrm{E}[U(\xi_t) - U(\xi_{t+1}) \mid \xi_t] \geq c_l$$

Upper bound: $\sum_{x \in \mathcal{X}} \pi_0(x) \cdot \dfrac{U(x)}{c_l}$

Multiplicative vs. Additive

For additive and multiplicative drift analysis, which one is stronger?

Theorem. [Doerr et al., Algorithmica'12] Multiplicative drift analysis can be proved by using additive drift analysis. Stronger

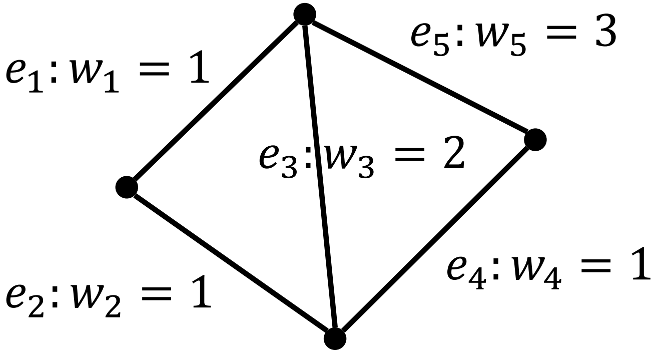
Multiplicative drift analysis can be easier to be used when the expected drift under a natural distance function is proportional to the current distance

Application illustration: (1+1)-EA for MST

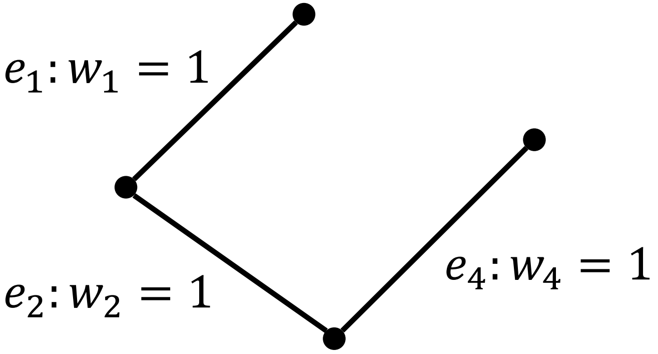
Minimum spanning tree (MST):

- Given: an undirected connected graph $G = (V, E)$ on n vertices and m edges with positive integer weights $w: E \to \mathbb{N}^+$

- The Goal: find a connected subgraph $E' \subseteq E$ with the minimum weight



The original graph



The minimum spanning tree

Application illustration: (1+1)-EA for MST

(1+1)-EA: Given a pseudo-Boolean function f:

1. $\boldsymbol{x} :=$ randomly selected from $\{0,1\}^n$.
2. Repeat until some termination criterion is met
3. $\quad \boldsymbol{x}' :=$ flip each bit of $\boldsymbol{x}$ with probability $1/n$.
4. $\quad$ if $f(\boldsymbol{x}') \geq f(\boldsymbol{x})$ $\longrightarrow$ $f(\boldsymbol{x}') \leq f(\boldsymbol{x})$
5. $\quad\quad \boldsymbol{x} = \boldsymbol{x}'$.

Solution representation: $\boldsymbol{x} \in \{0,1\}^m \leftrightarrow$ a subgraph

e.g., $\{e_1, e_2, e_4\} \rightarrow 11010$ $\quad$ $x_i = 1$ means that edge e_i is selected

Fitness function:

the number of connected components

the maximum edge weight

$$min \quad f(\boldsymbol{x}) = (c(\boldsymbol{x}) - 1) \cdot w_{ub} + \sum_{i:x_i=1} w_i$$

the sum of edge weights

$w_{ub} = n^2 \cdot w_{max}$, to make a subgraph with less connected components better

Proof

Theorem. [Neumann & Wegener, TCS'07; Doerr et al., Algorithmica'12] The expected running time of the (1+1)-EA solving the MST problem is $O(m^2(\log n + \log w_{max}))$.

Main idea:

(1) obtain a connected subgraph $\longrightarrow$ $c(\boldsymbol{x}) = 1$

(2) obtain a minimum spanning tree

The analysis of phase (1): $min \;\; f(\boldsymbol{x}) = (c(\boldsymbol{x}) - 1) \cdot w_{ub} + \sum_{i:x_i=1} w_i$

- $c(\boldsymbol{x})$ cannot increase
- at least $c(\boldsymbol{x}) - 1$ edges, the insertion of which can decrease $c(\boldsymbol{x})$ by 1

the probability of decreasing $c(\boldsymbol{x})$ by 1 is at least $\frac{c(\boldsymbol{x})-1}{m}(1-\frac{1}{m})^{m-1}$

flip one of those $c(x) - 1$ 0-bits   keep the other bits unchanged

Proof

Theorem. [Neumann & Wegener, TCS'07; Doerr et al., Algorithmica'12] The expected running time of the (1+1)-EA solving the MST problem is $O(m^2(\log n + \log w_{max}))$.

Main idea:

(1) obtain a connected subgraph $\longrightarrow$ $c(\boldsymbol{x}) = 1$

(2) obtain a minimum spanning tree

The analysis of phase (1): $min \;\; f(\boldsymbol{x}) = (c(\boldsymbol{x}) - 1) \cdot w_{ub} + \sum_{i:x_i=1} w_i$

- $c(\boldsymbol{x})$ cannot increase
- at least $c(\boldsymbol{x}) - 1$ edges, the insertion of which can decrease $c(\boldsymbol{x})$ by 1

the probability of decreasing $c(\boldsymbol{x})$ by 1 is at least $\frac{c(\boldsymbol{x})-1}{m}(1-\frac{1}{m})^{m-1}$

fitness level method $\Rightarrow$ The expected running time upper bound: $\sum_{c(x)=n}^{2} \frac{em}{c(x)-1} \in O(m \log n)$

Proof

The analysis of phase (2): $min\ \ f(\boldsymbol{x}) = (c(\boldsymbol{x}) - 1) \cdot w_{ub} + \sum_{i:x_i=1} w_i$

- it will always be connected, i.e., $c(\boldsymbol{x}) = 1$ always holds
- to analyze $f(\boldsymbol{x})$ $\rightarrow$ f_{opt} the weight of a minimum spanning tree

Using multiplicative drift analysis:

- design the distance function: $V(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{opt}$
- analyze the expected drift:

$$\mathrm{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = V(\boldsymbol{x}) - \mathrm{E}[V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = f(\boldsymbol{x}) - \mathrm{E}[f(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}]$$

$$\geq f(\boldsymbol{x}) - \left(\sum_{i=1}^{m-(n-1)} f(\boldsymbol{y}^i) \cdot \frac{1}{m} (1 - \frac{1}{m})^{m-1} + \sum_{i=1}^{n} f(\boldsymbol{z}^i) \cdot \frac{1}{m^2} (1 - \frac{1}{m})^{m-2} + (1 - \cdots) f(\boldsymbol{x}) \right)$$

there exists a set of $m - (n-1)$ 1-bit flips and a set of n 2-bit flips such that the average weight decrease is at least $(f(\boldsymbol{x}) - f_{opt})/(m+1)$

Proof

- to analyze $f(\boldsymbol{x})$ $\rightarrow$ f_{opt} the weight of a minimum spanning tree

Using multiplicative drift analysis:

- design the distance function: $V(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{opt}$
- analyze the expected drift:

$$\mathrm{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = V(\boldsymbol{x}) - \mathrm{E}[V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = f(\boldsymbol{x}) - \mathrm{E}[f(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}]$$

$$\geq f(\boldsymbol{x}) - \left(\sum_{i=1}^{m-(n-1)} f(\boldsymbol{y}^i) \cdot \frac{1}{m}(1-\frac{1}{m})^{m-1} + \sum_{i=1}^{n} f(\boldsymbol{z}^i) \cdot \frac{1}{m^2}(1-\frac{1}{m})^{m-2} + (1-\cdots)f(\boldsymbol{x})\right)$$

there exists a set of $m-(n-1)$ 1-bit flips and a set of n 2-bit flips such that the average weight decrease is at least $(f(\boldsymbol{x}) - f_{opt})/(m+1)$

$$= \sum_{i=1}^{m-(n-1)} \left(f(\boldsymbol{x}) - f(\boldsymbol{y}^i)\right) \cdot \frac{1}{m}(1-\frac{1}{m})^{m-1} + \sum_{i=1}^{n} \left(f(\boldsymbol{x}) - f(\boldsymbol{z}^i)\right) \cdot \frac{1}{m^2}(1-\frac{1}{m})^{m-2}$$

Proof

- to analyze $f(\boldsymbol{x}) \rightarrow f_{opt}$ the weight of a minimum spanning tree

Using multiplicative drift analysis:

- design the distance function: $V(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{opt}$
- analyze the expected drift:

$$\mathrm{E}[\,V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = V(\boldsymbol{x}) - \mathrm{E}[\,V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = f(\boldsymbol{x}) - \mathrm{E}[\,f(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}]$$

$$\geq f(\boldsymbol{x}) - \left(\sum_{i=1}^{m-(n-1)} f(\boldsymbol{y}^{\boldsymbol{i}}) \cdot \frac{1}{m}(1-\frac{1}{m})^{m-1} + \sum_{i=1}^{n} f(\boldsymbol{z}^{\boldsymbol{i}}) \cdot \frac{1}{m^2}(1-\frac{1}{m})^{m-2} + (1-\cdots) f(\boldsymbol{x})\right)$$

there exists a set of $m-(n-1)$ 1-bit flips and a set of n 2-bit flips such that the average weight decrease is at least $(f(\boldsymbol{x}) - f_{opt})/(m+1)$

$$\geq \boxed{\frac{1}{m}(1-\frac{1}{m})^{m-1} \cdot \frac{1}{m+1}} \cdot \left(\sum_{i=1}^{m-(n-1)} \left(f(\boldsymbol{x}) - f(\boldsymbol{y}^{\boldsymbol{i}})\right) \cdot + \sum_{i=1}^{n} \left(f(\boldsymbol{x}) - f(\boldsymbol{z}^{\boldsymbol{i}})\right)\right)$$

Proof

- to analyze $f(\boldsymbol{x}) \rightarrow f_{opt}$ the weight of a minimum spanning tree

Using multiplicative drift analysis:

- design the distance function: $V(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{opt}$
- analyze the expected drift:

$$\mathrm{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = V(\boldsymbol{x}) - \mathrm{E}[V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = f(\boldsymbol{x}) - \mathrm{E}[f(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}]$$

$$\geq f(\boldsymbol{x}) - \left(\sum_{i=1}^{m-(n-1)} f(\boldsymbol{y}^{\boldsymbol{i}}) \cdot \frac{1}{m}(1-\frac{1}{m})^{m-1} + \sum_{i=1}^{n} f(\boldsymbol{z}^{\boldsymbol{i}}) \cdot \frac{1}{m^2}(1-\frac{1}{m})^{m-2} + (1-\cdots) f(\boldsymbol{x})\right)$$

there exists a set of $m-(n-1)$ 1-bit flips and a set of n 2-bit flips such that the average weight decrease is at least $(f(\boldsymbol{x}) - f_{opt})/(m+1)$

$$\geq \frac{1}{m}(1-\frac{1}{m})^{m-1} \cdot \frac{1}{m+1} \cdot \left(\sum_{i=1}^{m-(n-1)} \left(f(\boldsymbol{x}) - f(\boldsymbol{y}^{\boldsymbol{i}})\right) \cdot + \sum_{i=1}^{n} \left(f(\boldsymbol{x}) - f(\boldsymbol{z}^{\boldsymbol{i}})\right)\right)$$

Proof

- to analyze $f(\boldsymbol{x}) \rightarrow f_{opt}$ the weight of a minimum spanning tree

Using multiplicative drift analysis:

- design the distance function: $V(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{opt}$
- analyze the expected drift:

$$\mathrm{E}[V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = V(\boldsymbol{x}) - \mathrm{E}[V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] = f(\boldsymbol{x}) - \mathrm{E}[f(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}]$$

$$\geq f(\boldsymbol{x}) - \left(\sum_{i=1}^{m-(n-1)} f(\boldsymbol{y}^{\boldsymbol{i}}) \cdot \frac{1}{m}(1-\frac{1}{m})^{m-1} + \sum_{i=1}^{n} f(\boldsymbol{z}^{\boldsymbol{i}}) \cdot \frac{1}{m^2}(1-\frac{1}{m})^{m-2} + (1-\cdots)f(\boldsymbol{x})\right)$$

there exists a set of $m-(n-1)$ 1-bit flips and a set of n 2-bit flips such that the average weight decrease is at least $(f(\boldsymbol{x}) - f_{opt})/(m+1)$

$$\geq \frac{1}{m}(1-\frac{1}{m})^{m-1} \cdot \frac{f(\boldsymbol{x}) - f_{opt}}{m+1} \geq \frac{1}{em(m+1)} V(\boldsymbol{x})$$

Proof

The analysis of phase (2): $min \ f(x) = (c(x)-1) \cdot w_{ub} + \sum_{i:x_{i=1}} w_i$

- it will always be connected, i.e., $c(x) = 1$ always holds
- to analyze $f(x) \rightarrow f_{opt}$ the weight of a minimum spanning tree

Using multiplicative drift analysis:

- design the distance function: $V(\boldsymbol{x}) = f(\boldsymbol{x}) - f_{opt}$
- analyze the expected drift:

$$\mathrm{E}[\, V(\xi_t) - V(\xi_{t+1}) \mid \xi_t = \boldsymbol{x}] \geq \frac{1}{em(m+1)} V(\boldsymbol{x})$$

proportional to the current distance

Upper bound on the expected running time:

$$\sum_{x \in \mathcal{X}} \pi_0(\boldsymbol{x}) \cdot \frac{1 + \ln(V(\boldsymbol{x})/V_{min})}{\delta} \leq em(m+1)(1 + \ln(mw_{max}))$$

$V(\boldsymbol{x}) \leq mw_{max}$ $\quad V_{min} \geq 1$ $\quad \in O(m^2(\log n + \log w_{max}))$

Proof

Theorem. [Neumann & Wegener, TCS'07; Doerr et al., Algorithmica'12] The expected running time of the (1+1)-EA solving the MST problem is $O(m^2(\log n + \log w_{max}))$.

Main idea:

(1) obtain a connected subgraph

(2) obtain a minimum spanning tree

The expected running time of phase (1): $O(m \log n)$

The expected running time of phase (2): $O(m^2(\log n + \log w_{max}))$

The total expected running time: $O(m^2(\log n + \log w_{max}))$

Negative drift

Simplified negative drift theorem
for proving exponential lower bounds [Oliveto & Witt, Algorithmica'11]

Theorem 4 (Simplified Drift Theorem) *Let X_t, $t \geq 0$, be the random variables describing a Markov process over a finite state space $S \subseteq \mathbb{R}_0^+$ and denote $\underline{\Delta_t(i) := (X_{t+1} - X_t \mid X_t = i)}$ for $i \in S$ and $t \geq 0$. Suppose there exist an interval $[a, b]$ in the state space, two constants $\delta, \varepsilon > 0$ and, possibly depending on $\ell := b - a$, a function $r(\ell)$ satisfying $1 \leq r(\ell) = o(\ell/\log(\ell))$ such that for all $t \geq 0$ the following two conditions hold:*

1. $E(\Delta_t(i)) \geq \varepsilon$ *for* $a < i < b$,
2. $\mathrm{Prob}(\Delta_t(i) \leq -j) \leq \frac{r(\ell)}{(1+\delta)^j}$ *for* $i > a$ *and* $j \in \mathbb{N}_0$.

The expected drift is negative,
i.e., away from the target in expectation

Then there is a constant $c^* > 0$ such that for $T^* := \min\{t \geq 0 \colon X_t \leq a \mid X_0 \geq b\}$ it holds $\mathrm{Prob}(T^* \leq 2^{c^*\ell/r(\ell)}) = 2^{-\Omega(\ell/r(\ell))}$.

The probability of a drift
towards the target
decreases exponentially

Exponential running time

negative drift  very unlikely 

Negative drift

Simplified negative drift theorem
for proving exponential lower bounds [Oliveto & Witt, Algorithmica'11]

Theorem 4 (Simplified Drift Theorem) *Let X_t, $t \geq 0$, be the random variables describing a Markov process over a finite state space $S \subseteq \mathbb{R}_0^+$ and denote $\Delta_t(i) := (X_{t+1} - X_t \mid X_t = i)$ for $i \in S$ and $t \geq 0$. Suppose there exist an interval $[a, b]$ in the state space, two constants $\delta, \varepsilon > 0$ and, possibly depending on $\ell := b - a$, a function $r(\ell)$ satisfying $1 \leq r(\ell) = o(\ell/\log(\ell))$ such that for all $t \geq 0$ the following two conditions hold:*

a constant

1. $E(\Delta_t(i)) \geq \varepsilon$ *for* $a < i < b$,
2. $\mathrm{Prob}(\Delta_t(i) \leq -j) \leq \frac{r(\ell)}{(1+\delta)^j}$ *for* $i > a$ *and* $j \in \mathbb{N}_0$.

Then there is a constant $c^* > 0$ such that for $T^* := \min\{t \geq 0 \colon X_t \leq a \mid X_0 \geq b\}$ it holds $\mathrm{Prob}(T^* \leq 2^{c^*\ell/r(\ell)}) = 2^{-\Omega(\ell/r(\ell))}$.

relax

Simplified negative drift with self-loops [Rowe & Sudholt, TCS'14]

Simplified negative drift with scaling [Oliveto & Witt, TCS'14]

Negative drift

For proving exponential lower bounds on the running time of EAs:

To use:

Simplified negative drift [Oliveto & Witt, Algorithmica'11]

easier

Simplified negative drift with self-loops [Rowe & Sudholt, TCS'14]

Simplified negative drift with scaling [Oliveto & Witt, TCS'14]

stronger

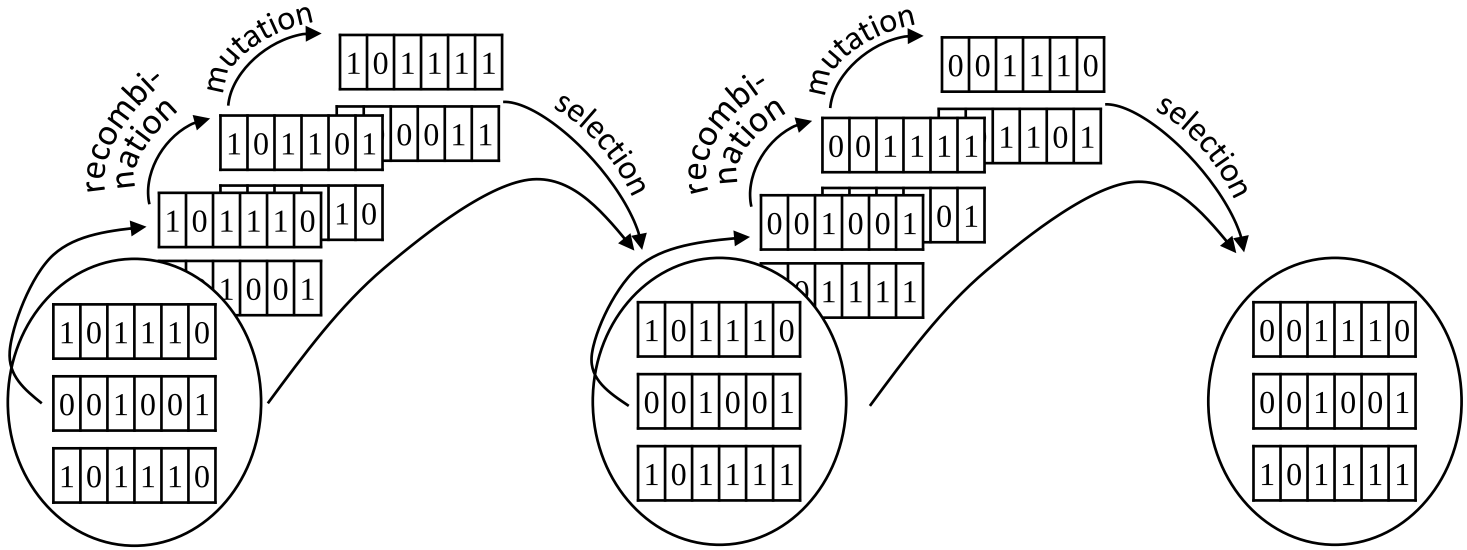
easier

Original negative drift [Hajek, 1982]

Switch analysis

Hard to be analyzed directly

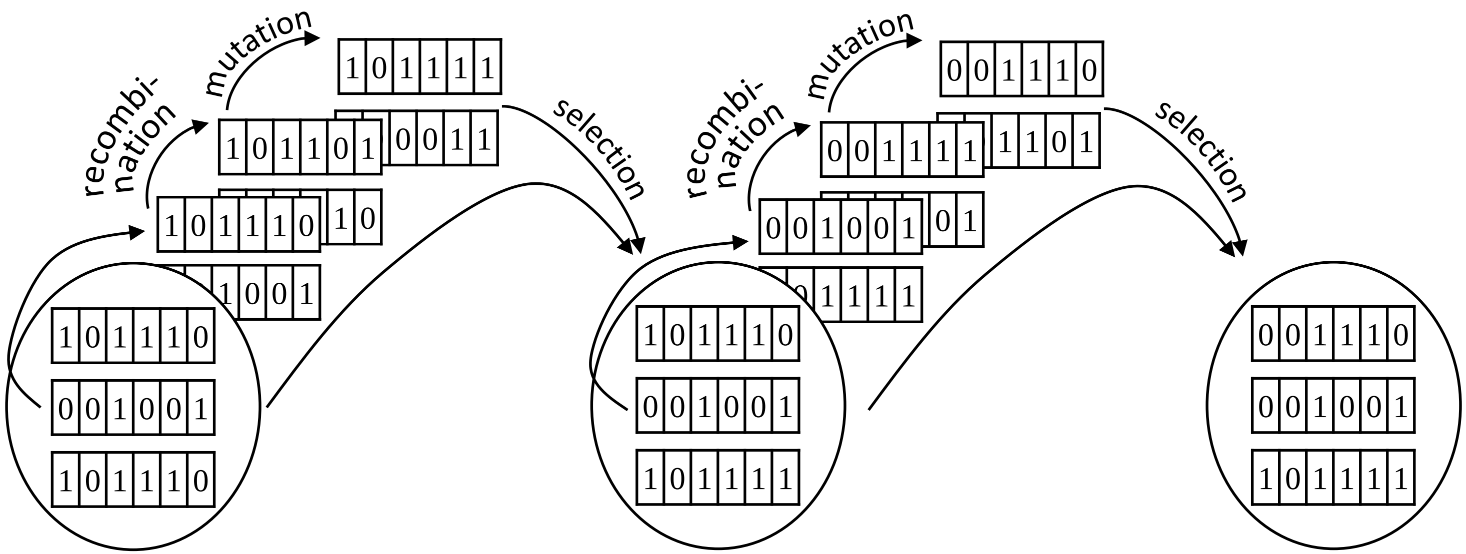
EA with recombination



Switch analysis

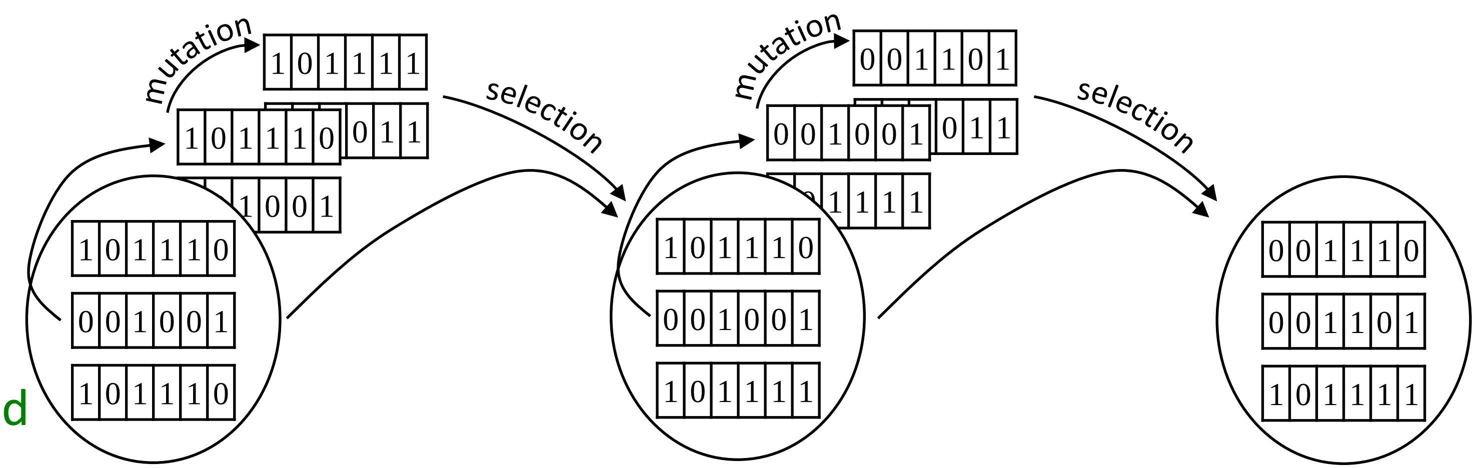
Hard to be analyzed directly

EA with recombination



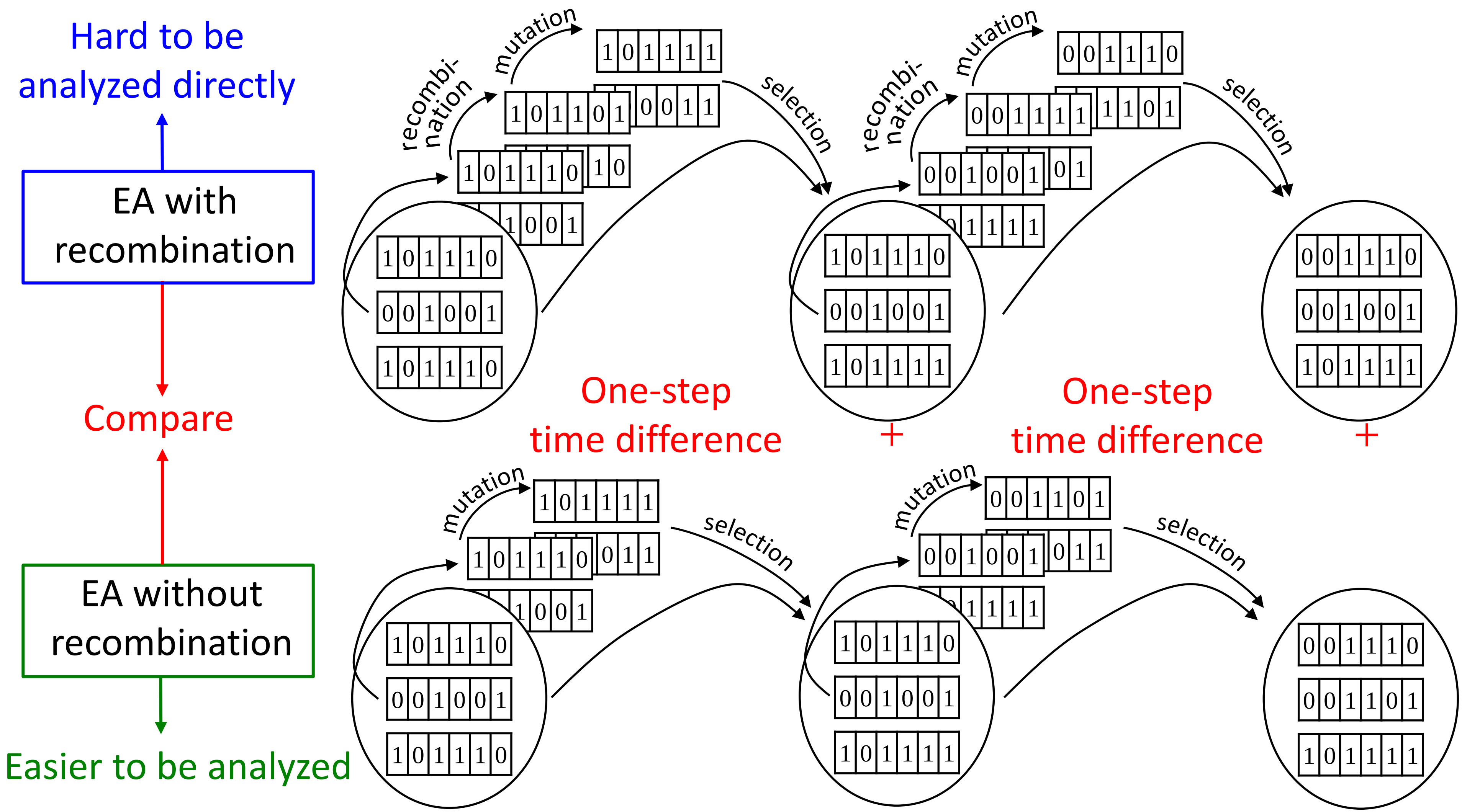
EA without recombination

Easier to be analyzed



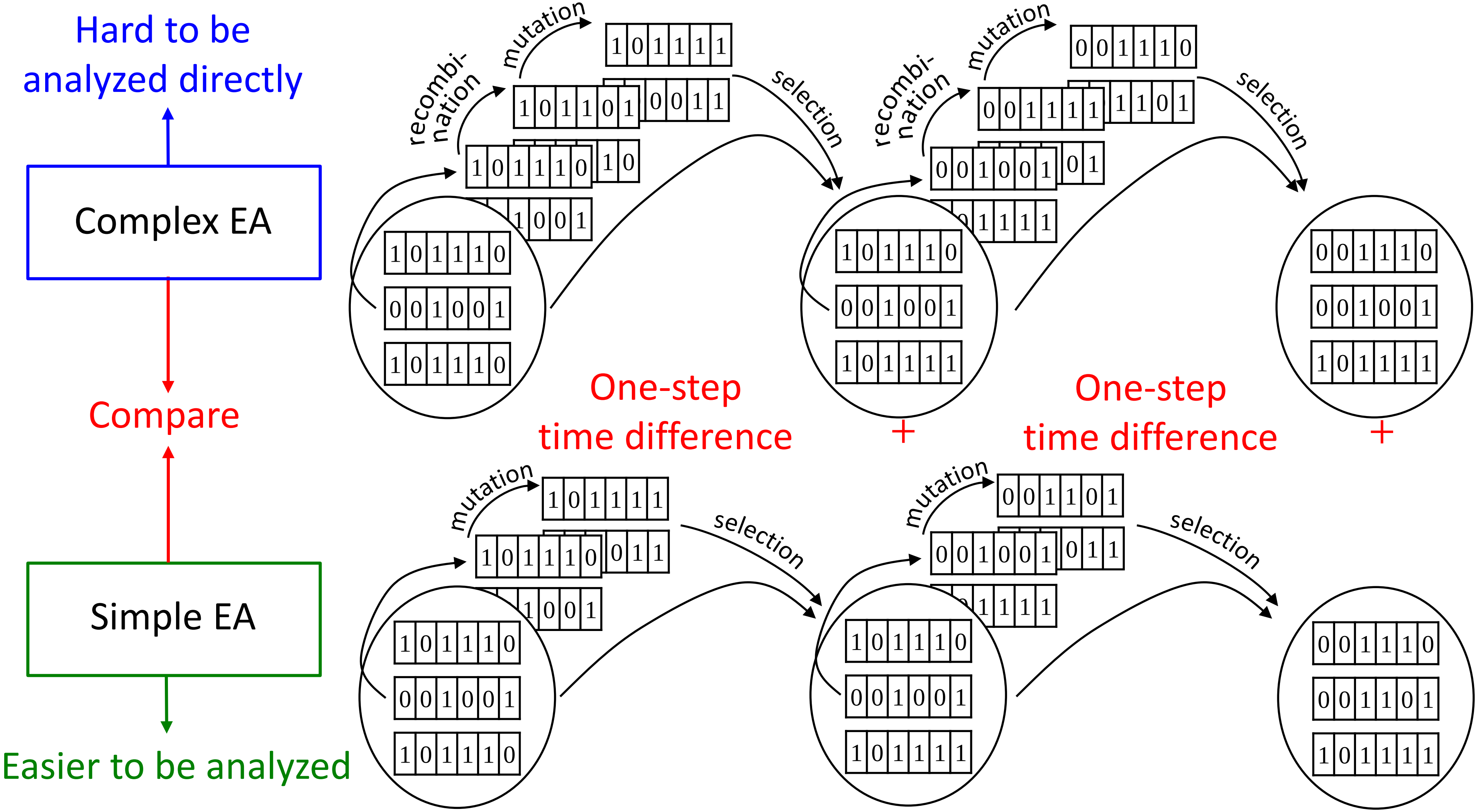
Switch analysis

Simplify the analysis



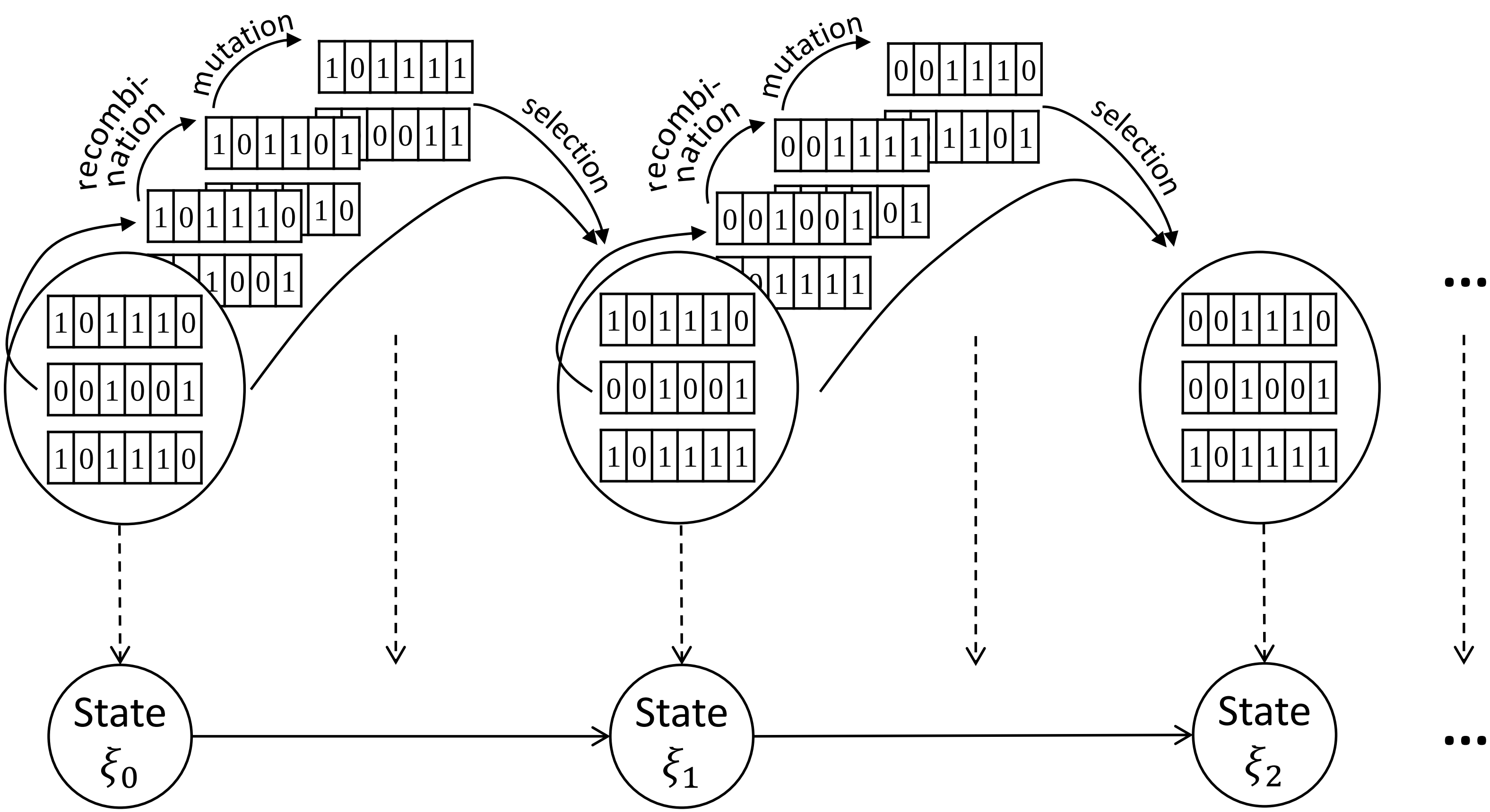
Switch analysis

Simplify the analysis



Switch analysis

Model an EA process as a Markov chain

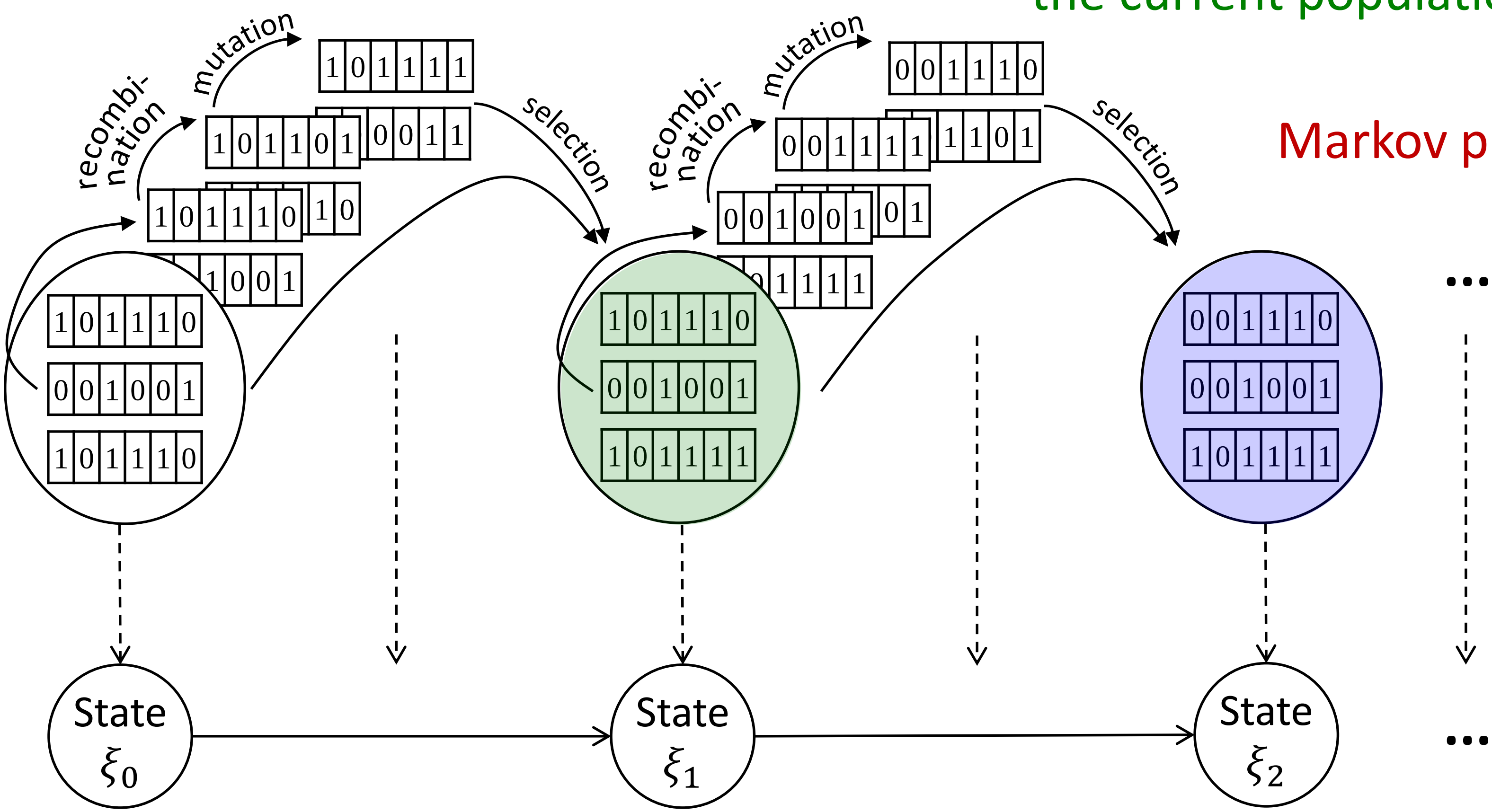


Switch analysis

Model an EA process as a Markov chain

The generation of the next population only depends on the current population

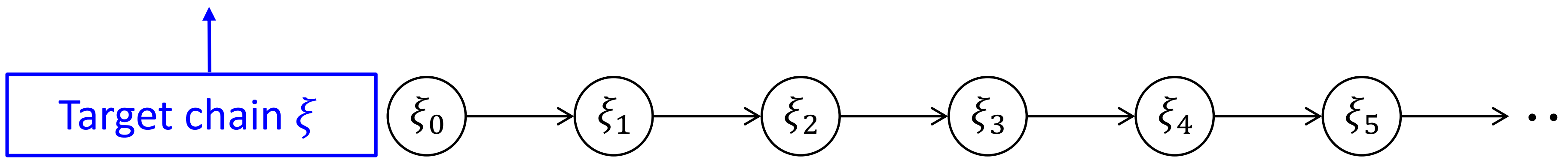
Markov property



Switch analysis

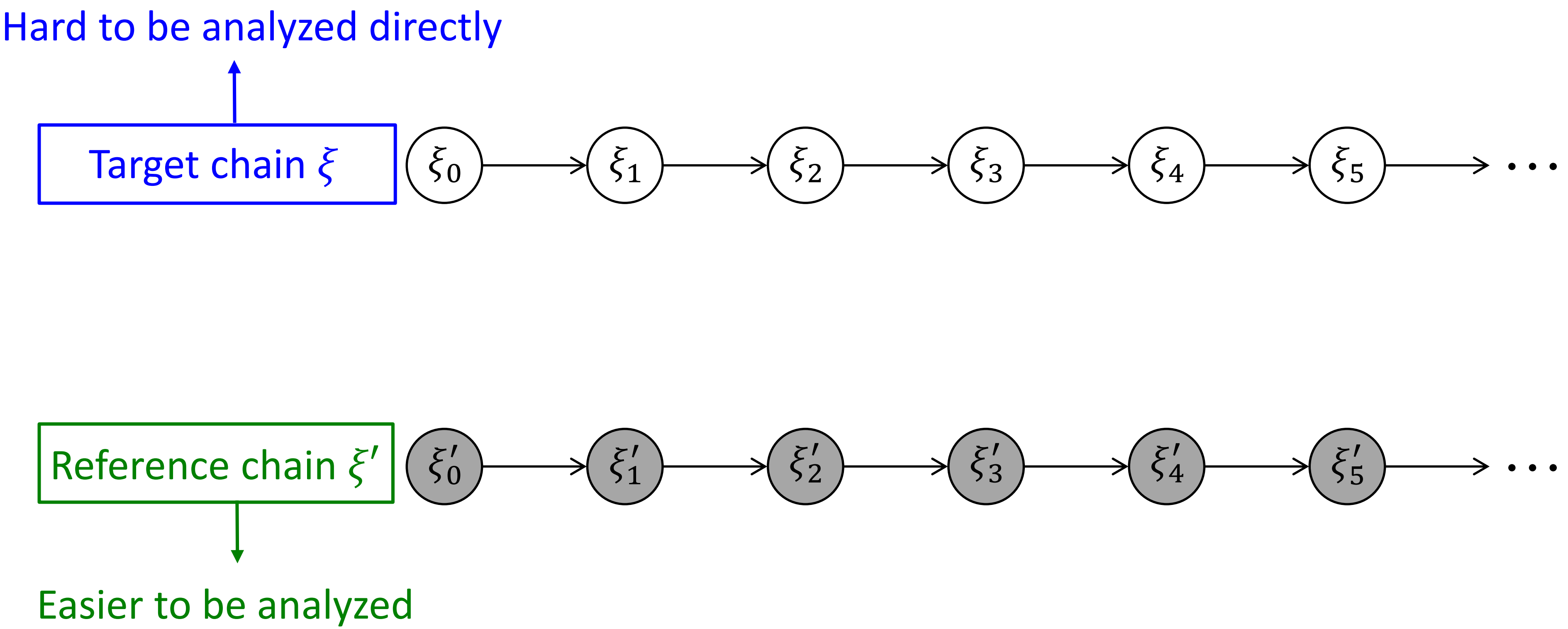
Main idea [Yu, Qian & Zhou, TEC'15]:

Hard to be analyzed directly



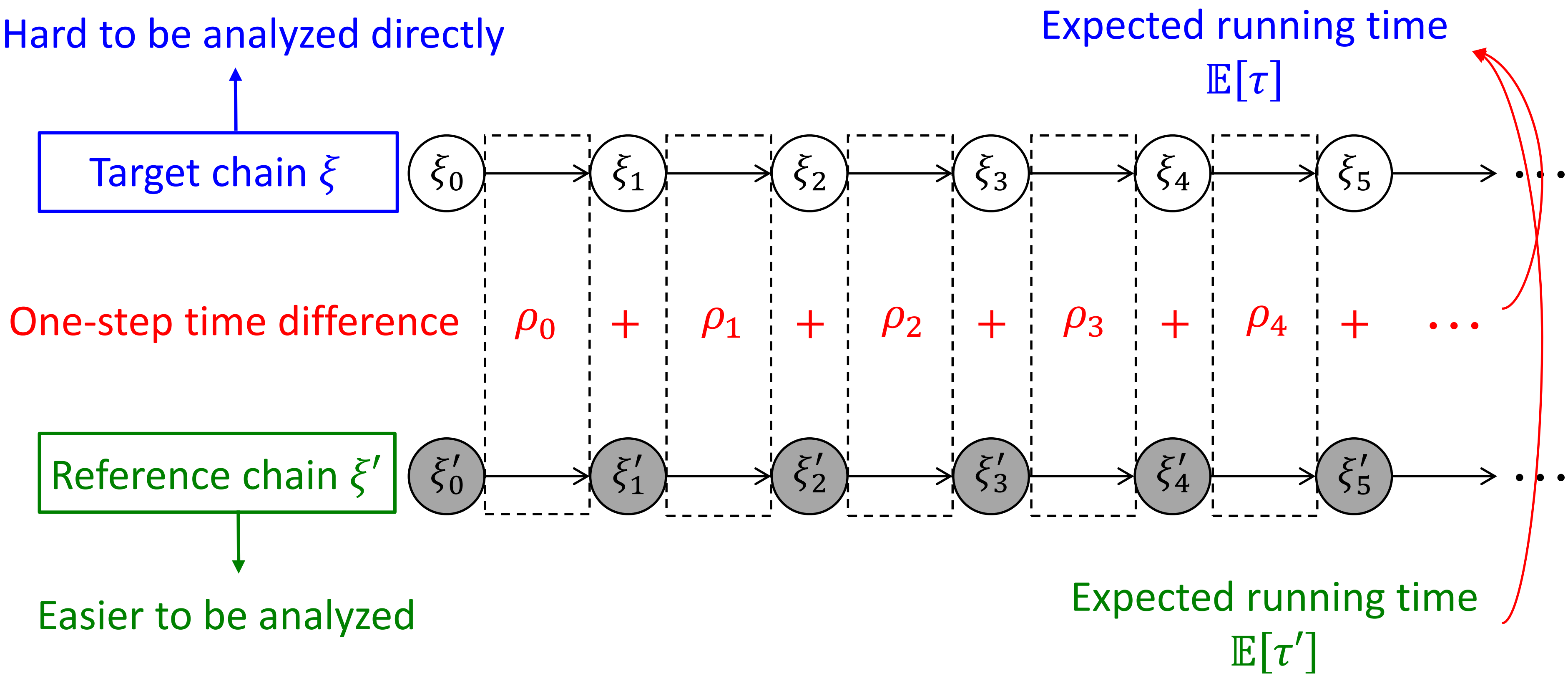
Switch analysis

Main idea [Yu, Qian & Zhou, TEC'15]:



Switch analysis

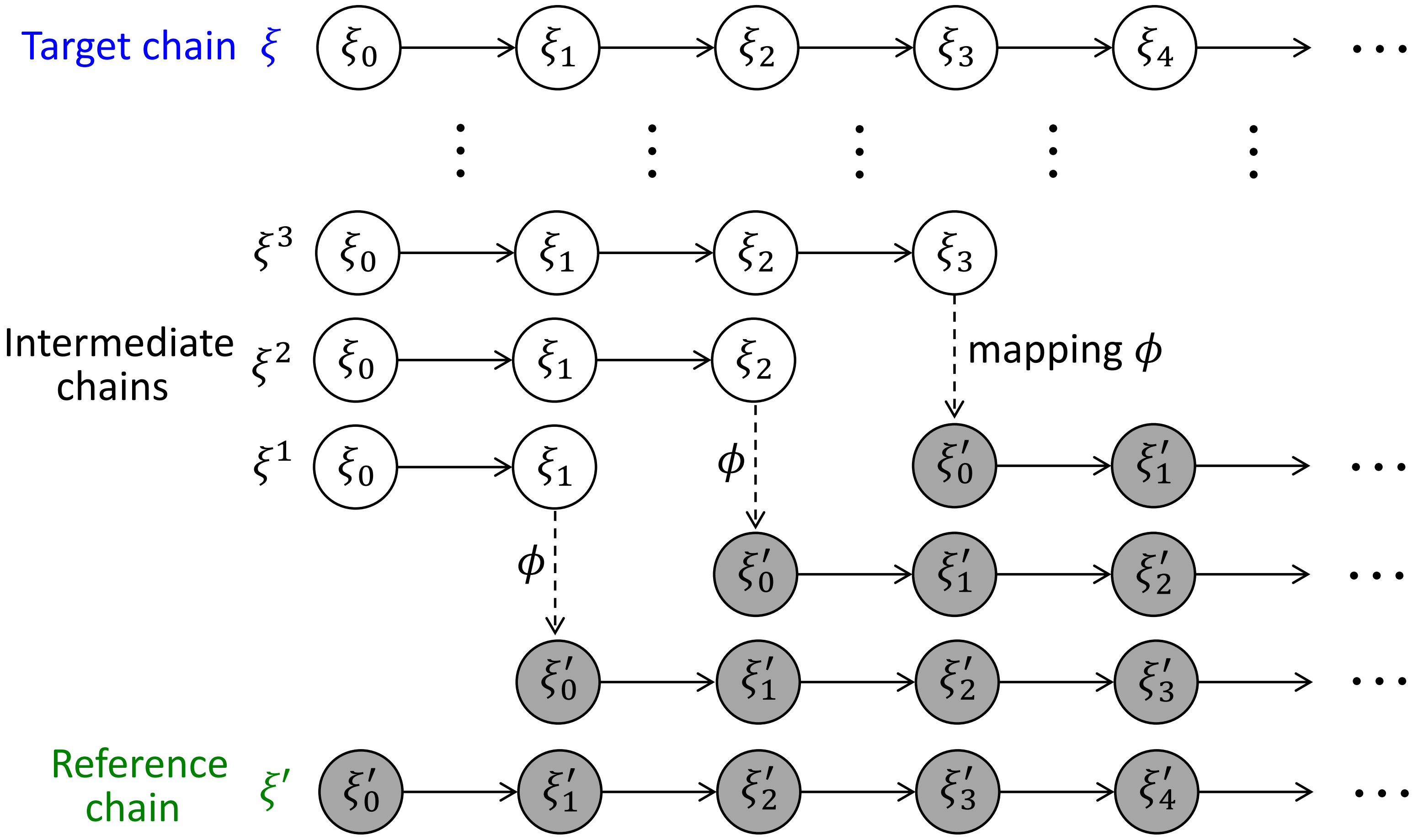
Main idea [Yu, Qian & Zhou, TEC'15]:



How to estimate one-step time difference ρ_t?

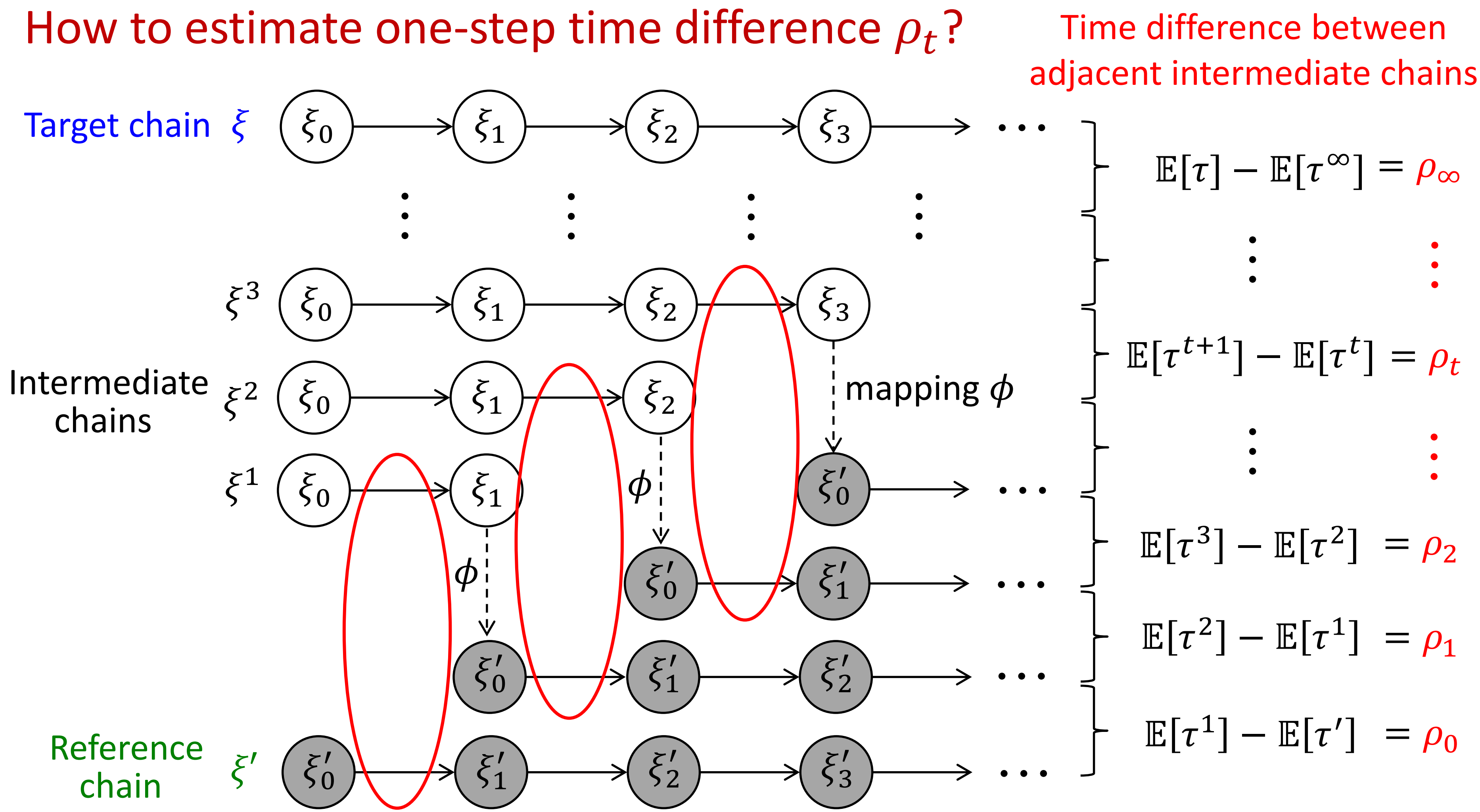
Switch analysis

How to estimate one-step time difference ρ_t?



Switch analysis

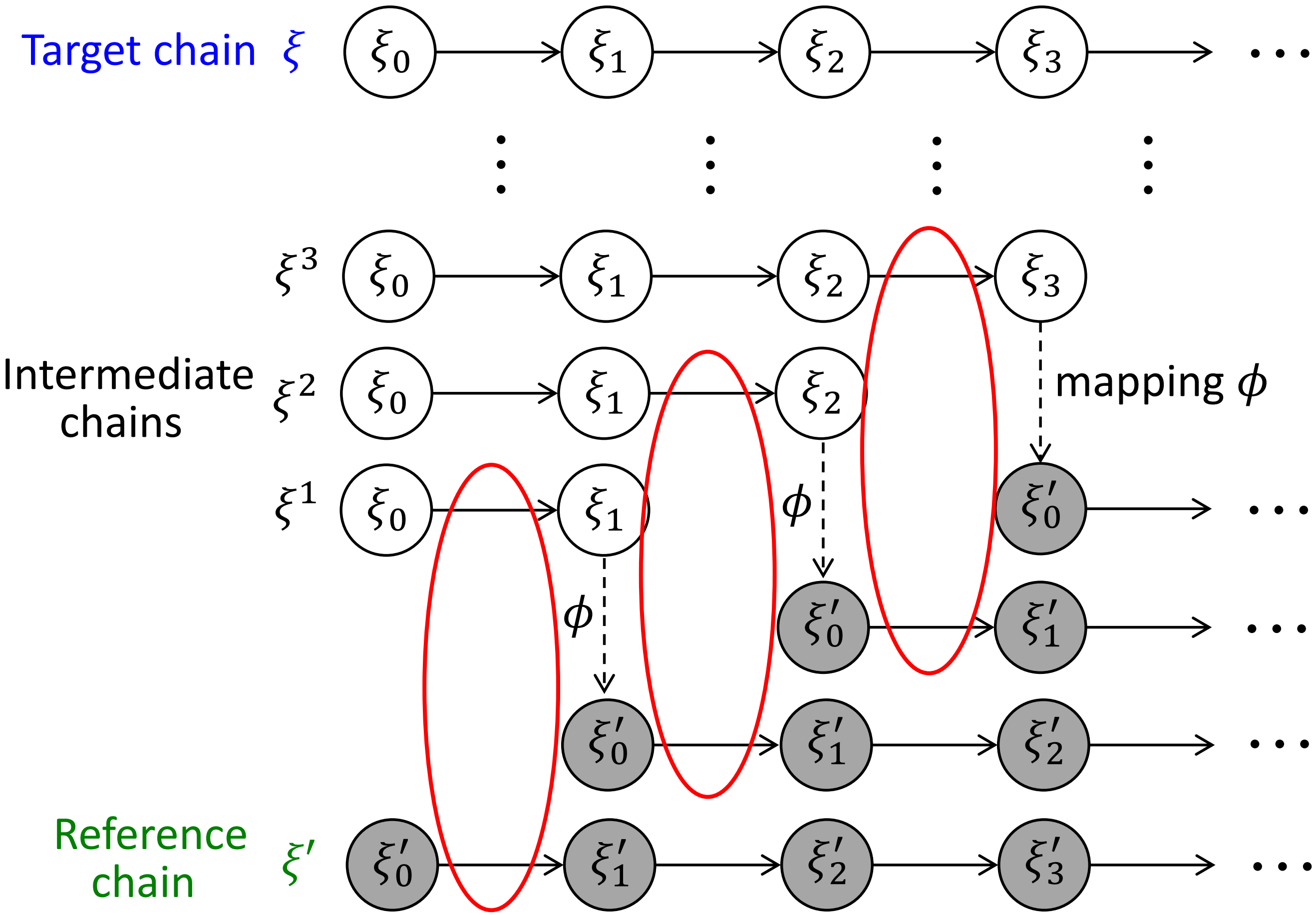
How to estimate one-step time difference ρ_t?



Switch analysis

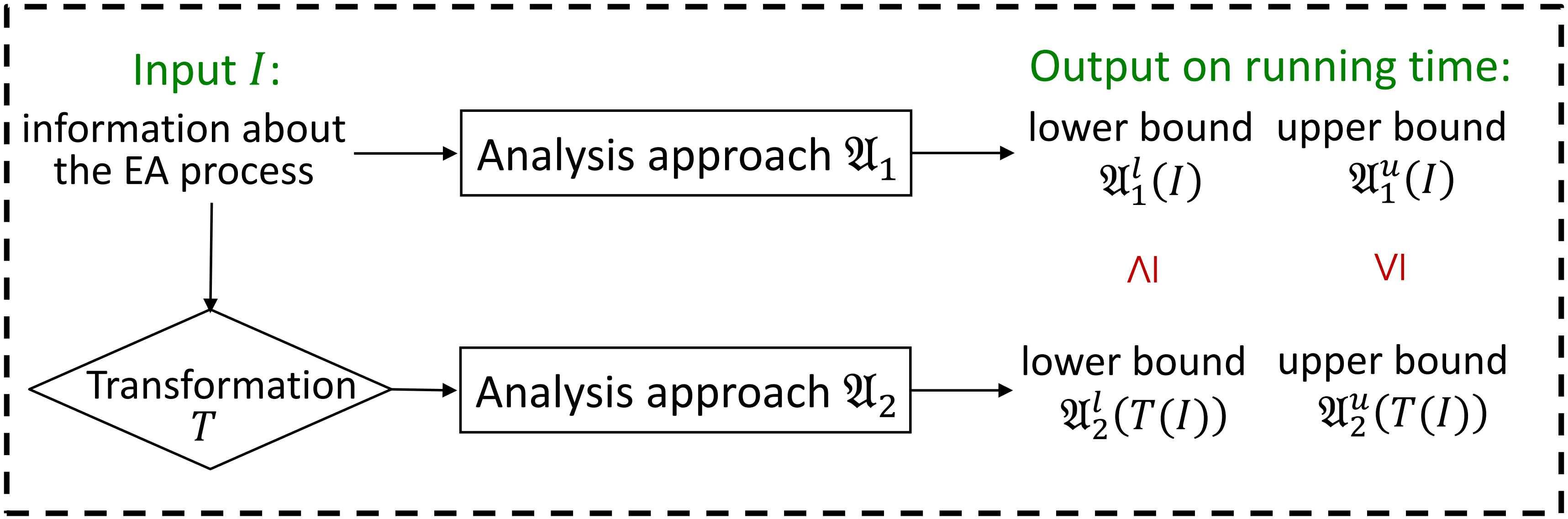
How to estimate one-step time difference ρ_t?

Time difference between adjacent intermediate chains



$$\mathbb{E}[\tau] - \mathbb{E}[\tau^{\infty}] = \rho_\infty$$
$$+$$
$$\vdots$$
$$+$$
$$\mathbb{E}[\tau^{t+1}] - \mathbb{E}[\tau^{t}] = \rho_t$$
$$+$$
$$\vdots$$
$$+$$
$$\mathbb{E}[\tau^{3}] - \mathbb{E}[\tau^{2}] = \rho_2$$
$$+$$
$$\mathbb{E}[\tau^{2}] - \mathbb{E}[\tau^{1}] = \rho_1$$
$$+$$
$$\mathbb{E}[\tau^{1}] - \mathbb{E}[\tau'] = \rho_0$$

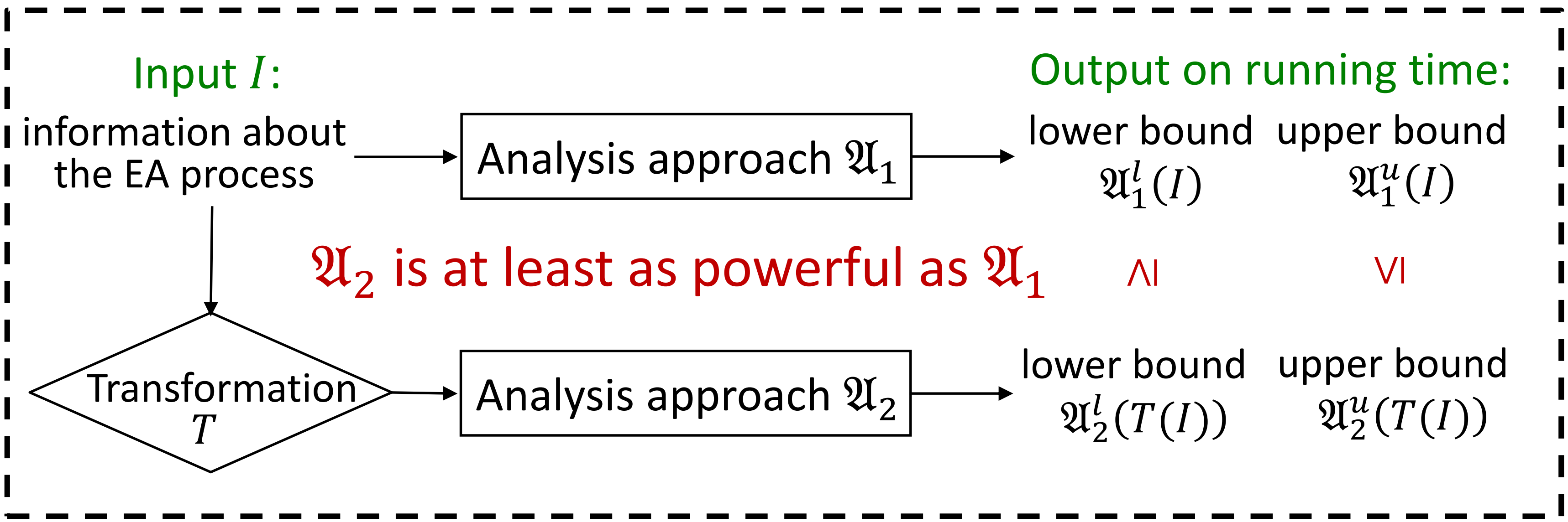
Ability of switch analysis



$\mathfrak{A}_2$ is at least as powerful as $\mathfrak{A}_1$

$\mathfrak{A}_2$ can derive at least the same tight bound as $\mathfrak{A}_1$ while requiring no more information

Ability of switch analysis

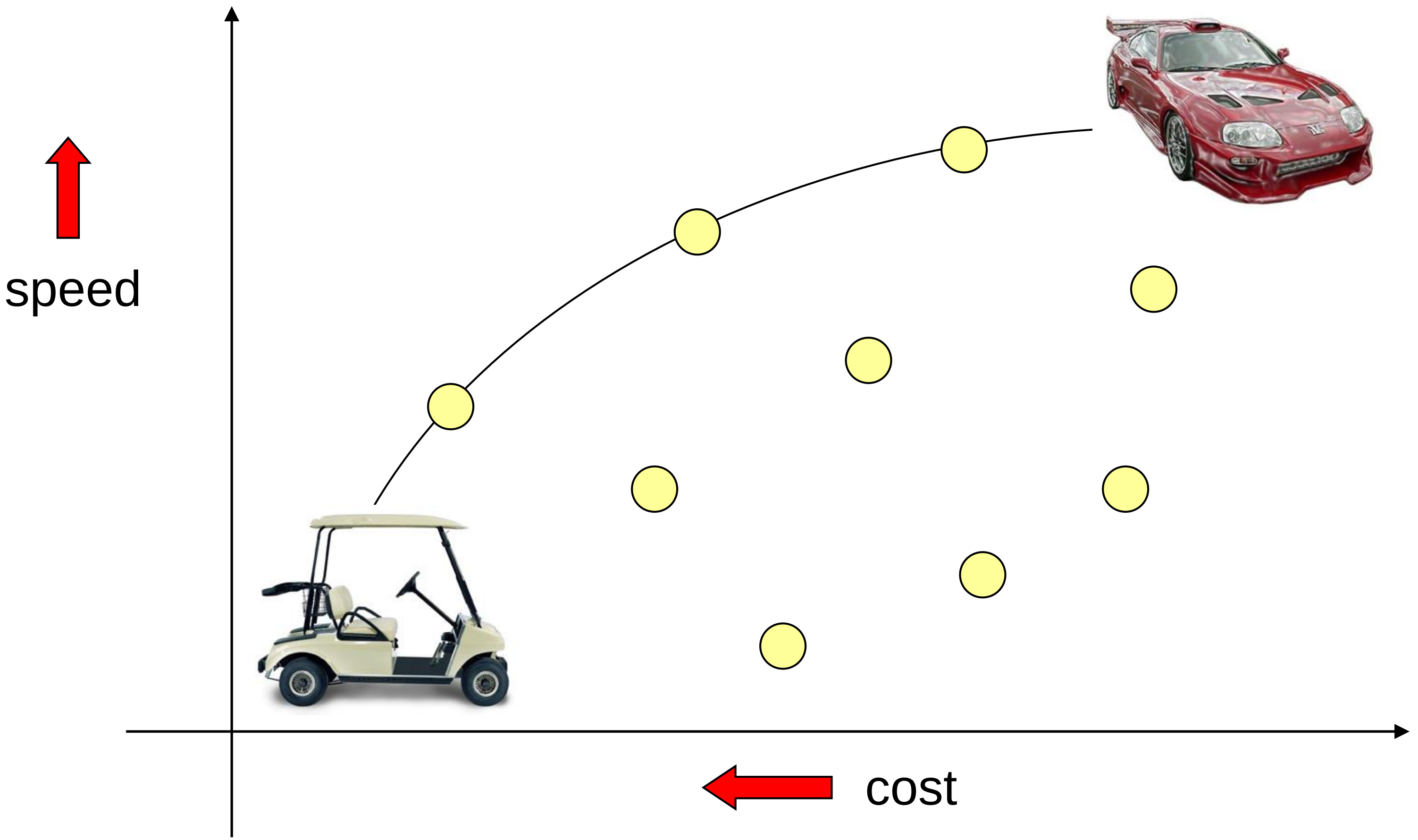


Switch analysis is at least as powerful as

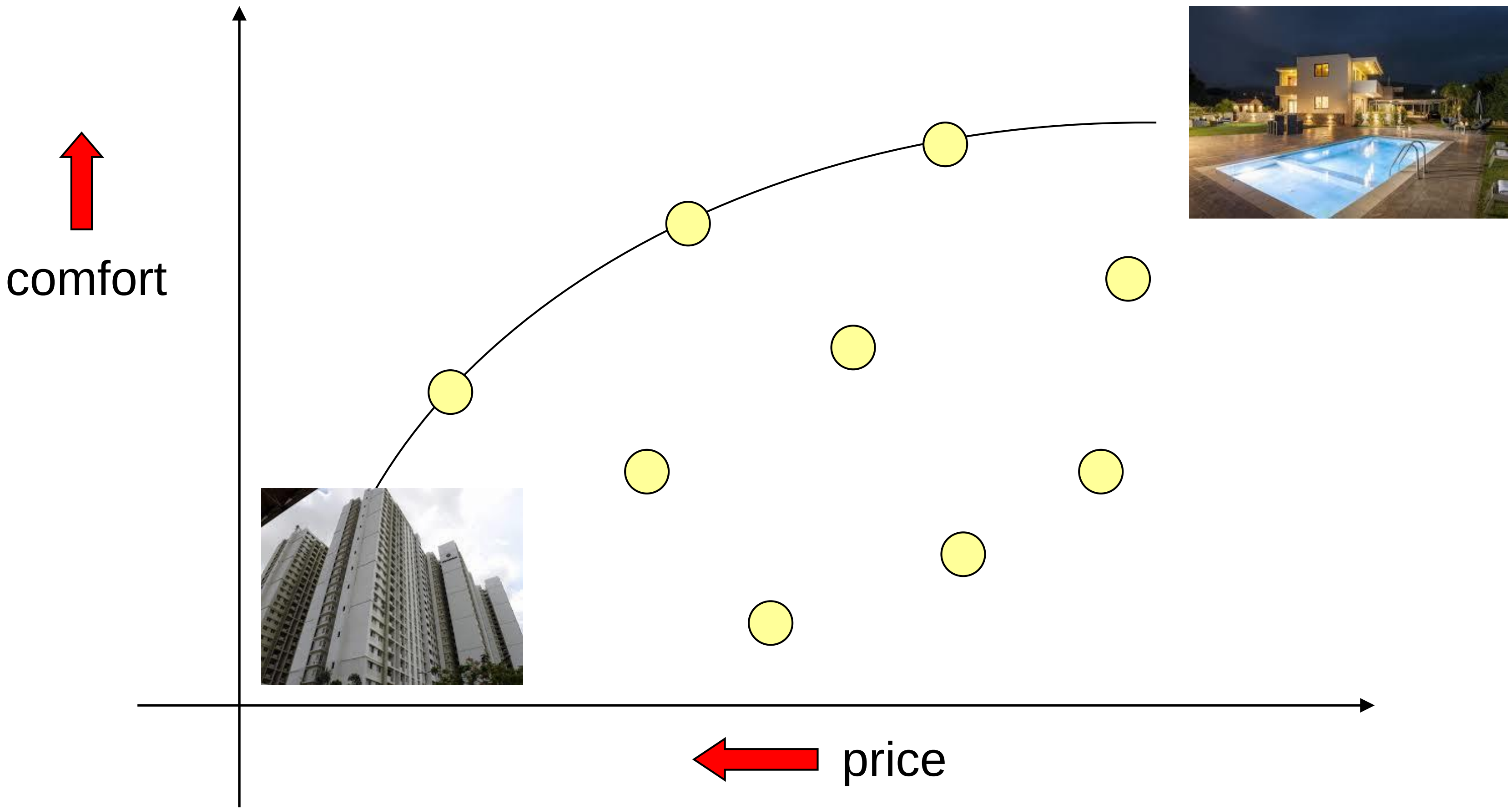
- Fitness level
- Drift analysis

Each one can be viewed as switch analysis with a specific reference chain

Example of multi-objective optimization



Example of multi-objective optimization



Running time analysis tools

- Fitness level
  - Original fitness level
  - Refined fitness level
- Drift analysis
  - Additive drift
  - Multiplicative drift
  - Negative drift
- Switch analysis

When EAs are applied to solve multi-objective optimization problems, can these approaches be easily used to analyze their running time?

Multi-objective optimization

Multi-objective optimization: optimize multiple objectives (which are usually conflicting) simultaneously

$$max_{x \in \mathcal{X}} \ \left(f_1(x), f_2(x), \dots, f_m(x)\right)$$

- x weakly dominates y, denoted as $x \succcurlyeq y$, if

$$\forall i \in \{1,2,\dots,m\}: \ f_i(x) \geq f_i(y)$$

- x dominates y, denoted as $x \succ y$, if

$$\forall i \in \{1,2,\dots,m\}: \ f_i(x) \geq f_i(y) \text{ and } \exists i \in \{1,2,\dots,m\}: \ f_i(x) > f_i(y)$$

- x is incomparable with y, if neither $x \succcurlyeq y$ nor $y \succcurlyeq x$

Multi-objective optimization

Multi-objective optimization: optimize multiple objectives (which are usually conflicting) simultaneously

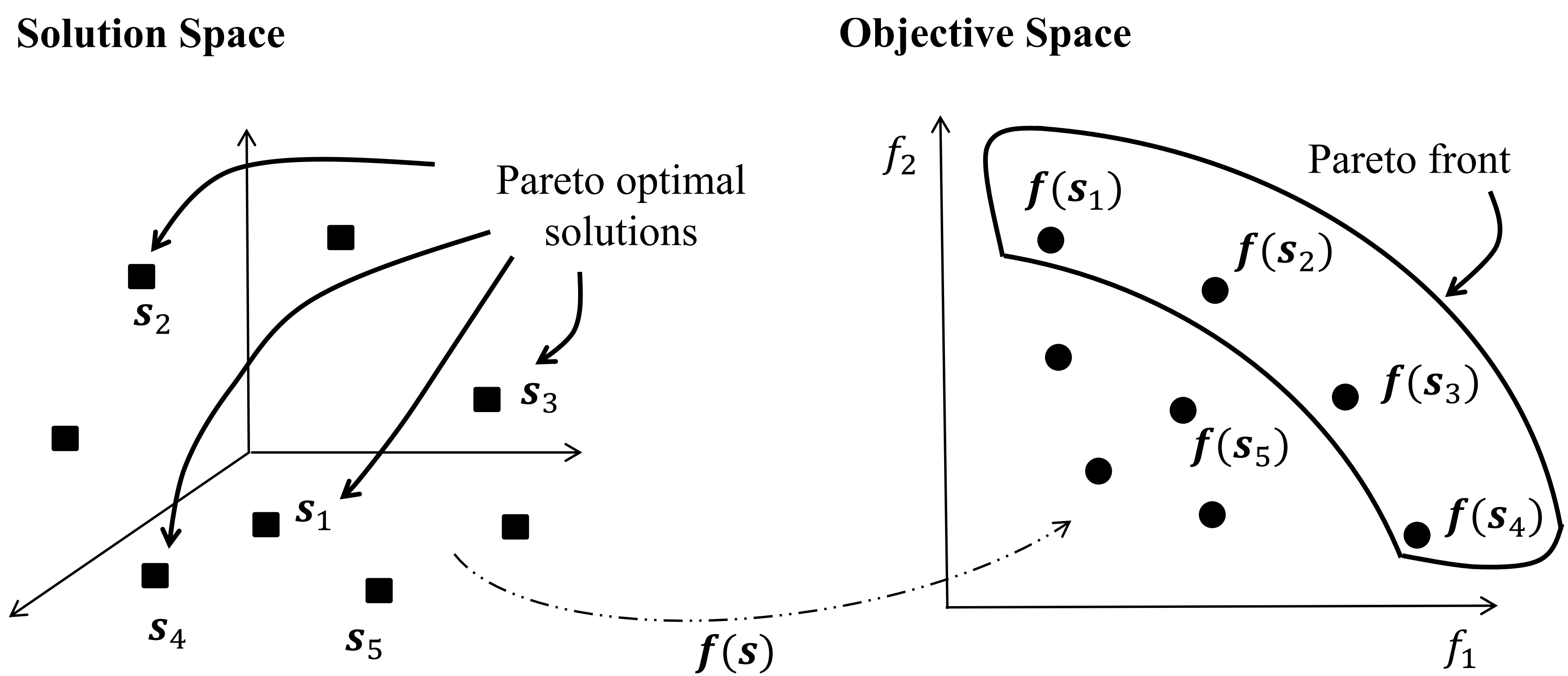
$$max_{x \in \mathcal{X}} \; \big(f_1(x), f_2(x), \ldots, f_m(x)\big)$$

A solution is Pareto optimal if no other solution dominates it

The collection of objective vectors of all Pareto optimal solutions is called the Pareto front

The goal of multi-objective optimization is to find a set of solutions whose objective vectors cover the Pareto front

Example of multi-objective optimization



$s_2 \succcurlyeq s_5$ $\quad$ $s_2 \succ s_5$ $\quad$ s_2 is incomparable with s_3

Assume the solution space contains only these 8 solutions

Example of running time analysis

GSEMO: Given a pseudo-Boolean function vector $\boldsymbol{f}$:

1. $\boldsymbol{x} :=$ randomly selected from $\{0,1\}^n$.
2. $P := \{\boldsymbol{x}\}$.
3. Repeat until some termination criterion is met
4. Choose $\boldsymbol{x}$ from P uniformly at random.
5. $\boldsymbol{x}' :=$ flip each bit of $\boldsymbol{x}$ with probability $1/n$.
6. if $\nexists\, \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ \boldsymbol{x}'$
7. $P := (P - \{\boldsymbol{z} \in P \mid \boldsymbol{x}' \succcurlyeq \boldsymbol{z}\}) \cup \{\boldsymbol{x}'\}$.

Keep the non-dominated solutions generated so-far

LOTZ: $$max_{\boldsymbol{x}\in\{0,1\}^n}\ \left(\sum_{i=1}^{n}\prod_{j=1}^{i}x_j,\ \sum_{i=1}^{n}\prod_{j=i}^{n}(1-x_j)\right)$$

Count the number of leading 1-bits   Count the number of trailing 0-bits

The Pareto set: $00\ldots00,\ 10\ldots00,\ \ldots,\ 11\ldots10,\ 11\ldots11$

The Pareto front: $(0,n), (1,n-1), \ldots, (n-1,1), (n,0)$

Proof

Theorem. [Giel, CEC'03] The expected running time of the GSEMO solving the LOTZ problem is $O(n^3)$.

Main idea:

(1) obtain the Pareto optimal solution $11 \ldots 11$

(2) obtain the Pareto set $\{00 \ldots 00, 10 \ldots 00, \ldots, 11 \ldots 10, 11 \ldots 11\}$

The analysis of phase (1):

the first objective $\sum_{i=1}^{n} \prod_{j=1}^{i} x_j$

- consider the largest LO value in the population, which never decreases
- select the solution with the largest LO value, and only flip its first 0 bit
- the probability is at least $\frac{1}{n+1} \frac{1}{n} (1 - \frac{1}{n})^{n-1}$

because the population size is no larger than $n + 1$, and a solution is uniformly selected at random

Proof

Theorem. [Giel, CEC'03] The expected running time of the GSEMO solving the LOTZ problem is $O(n^3)$.

Main idea:

(1) obtain the Pareto optimal solution $11 \ldots 11$

(2) obtain the Pareto set $\{00 \ldots 00, 10 \ldots 00, \ldots, 11 \ldots 10, 11 \ldots 11\}$

The analysis of phase (1):

the first objective $\sum_{i=1}^{n} \prod_{j=1}^{i} x_j$

- consider the largest LO value in the population, which never decreases
- select the solution with the largest LO value, and only flip its first 0 bit
- the probability is at least $\frac{1}{n+1} \frac{1}{n} (1 - \frac{1}{n})^{n-1}$

Proof

Theorem. [Giel, CEC'03] The expected running time of the GSEMO solving the LOTZ problem is $O(n^3)$.

The expected running time upper bound $n \cdot en(n+1) \in O(n^3)$

Main idea:

(1) obtain the Pareto optimal solution $11 \ldots 11$

(2) obtain the Pareto set $\{00 \ldots 00, 10 \ldots 00, \ldots, 11 \ldots 10, 11 \ldots 11\}$

The analysis of phase (1):

- consider the largest LO value in the population, which never decreases
- select the solution with the largest LO value, and only flip its first 0 bit
- the probability is at least $\frac{1}{n+1}\frac{1}{n}(1-\frac{1}{n})^{n-1}$

the probability of increasing the largest LO value by 1 is at least $\frac{1}{en(n+1)}$

it is sufficient to increase n times

Proof

The analysis of phase (2):

- the found Pareto optimal solutions will always be kept
- follow the path: $1^n \to 1^{n-1}0 \to \cdots \to 10^{n-1} \to 0^n$

the probability is at least $\frac{1}{n+1} \cdot \frac{1}{n}(1-\frac{1}{n})^{n-1}$

The expected running time upper bound: $n \cdot en(n+1) \in O(n^3)$

The expected running time of phase (1): $O(n^3)$

The total expected running time: $O(n^3)$

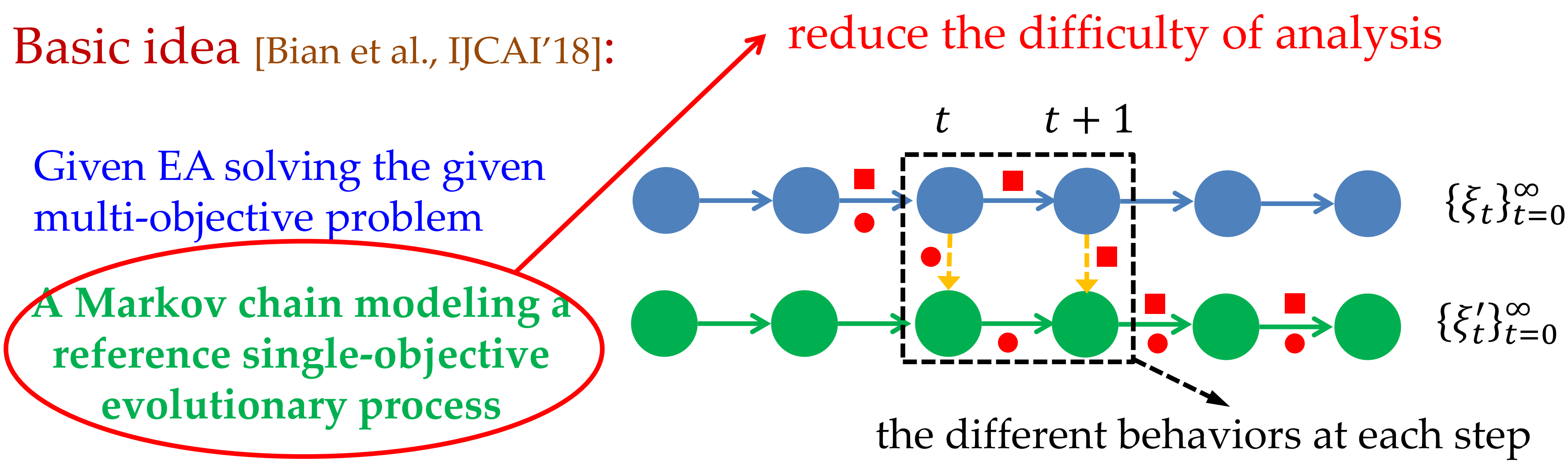
Switch analysis for multi-objective optimization

Theorem: $\xi \in \mathcal{X}$ modeling an EA solving a multi-objective optimization problem, a well-defined function $h_{\alpha,c}: \mathcal{X} \to \mathbb{N}_0$ and a Markov chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$ such that $\forall x \notin \mathcal{X}^*, \forall t \geq 0$,

$$\sum_{i \in [r]} \mathrm{P}(\min\{h(\xi_{t+1}), r\} = i \mid \xi_t = x) \mathrm{E}\left[\tau' \mid \xi_0' = 1^i 0^{r-i}\right]$$
$$\leq \sum_{y \in \mathcal{Y}} \mathrm{P}\left(\xi_1' = y \middle| \xi_0' = 1^{h(x)} 0^{r-h(x)}\right) \mathrm{E}[\tau' \mid \xi_1' = y] + \delta,$$

$$\Rightarrow \mathrm{E}[\tau \mid \xi_0 = x_0] \leq \mathrm{E}\left[\tau' \mid \xi_0' = 1^{\min\{h(x_0), r\}} 0^{r - \min\{h(x_0), r\}}\right] / (1 - \delta)$$

Basic idea [Bian et al., IJCAI'18]:



Application of switch analysis

Application [Bian, Qian and Tang, IJCAI'18]:

GSEMO	Problem	Previous result	Our result
Bi-objective	LOTZ	$O(n^3)$ [Giel, CEC'03]	$\le 6n^3$
	COCZ	$O(n^2 \log n)$ [Qian et al., AIJ'13]	$\le 3n^2 \log n$
Many-objective	mCOCZ	$O(n^{m+1})$ [Laumanns et al., TEC'04]	$O(n^m)$ for $m > 4$, $O(n^3 \log n)$ for $m = 4$
Approximate analysis	WOMM	—	$1/n$-approximation: $O(n^2(\log_l n + \log_l(w_n/w_1)))$

gives the leading constants

is asymptotically tighter than

L. Thiele

Member of Academia Europaea

Switch analysis is general and powerful

Running time analysis tools

- Fitness level
  - Original fitness level
  - Refined fitness level
- Drift analysis
  - Additive drift
  - Multiplicative drift
  - Negative drift
- Switch analysis

Results in single-objective optimization

(1+1)-EA	linear function $\Theta(n \log n)$ [Droste et al., TCS'02] minimum spanning tree $O(m^2 \log(n + w_{max}))$ [Neumann & Wegener, TCS'07] partition $O(n^2)$ with $\frac{4}{3}$ approximation [Witt, STACS'05]
(u+1)-EA	OneMax $O(un + n \log n)$; LeadingOnes $O(un \log n + n^2)$ [Witt, ECJ'06] maximum clique $O(un \log n)$ on sparse graphs [Storch, TCS'07] vertex cover $O(un \log n)$ on bipartite graphs [Oliveto et al., TEC'09]
(1+λ)-EA	linear function $O(\lambda n + n \log n)$ [Doerr & Kunnemann, TCS'15] vertex cover *exponential* on bipartite graphs [Oliveto et al., TEC'09]
(N+N)-EA	OneMax $O(Nn \log\log n + n \log n)$; LeadingOnes $O(Nn \log n + n^2)$ [Chen et al., TSMCB'09]

EDA [Chen et al., TEC'10]; ACO [Doerr et al., TCS'11]; PSO [Sudholt & Witt, TCS'10]; GP [Wagner et al., ECJ'15]

Results in multi-objective optimization

SEMO — LOTZ $\Theta(n^3)$; COCZ $O(n^2 \log n)$ [Laumanns et al., TEC'04]

LOTZ $O(n^3)$ [Giel, CEC'03]; $\Omega(n^2/p)$ [Doerr et al., CEC'13]

GSEMO — bi-objective minimum spanning tree $O(m^3 w_{min}(|C| + \log n + \log w_{max}))$ with 2 approximation [Neumann, EJOR'07]

DEMO — multi-objective all-pairs-shortest-path $O(nP_{max}g)$ *with* $r^{3g \log n}$ approximation [Neumann & Theile, PPSN'10]

REMO — LOTZ $\Theta(n^2)$; COCZ $\Theta(n \log n)$; bi-objective minimum spanning tree

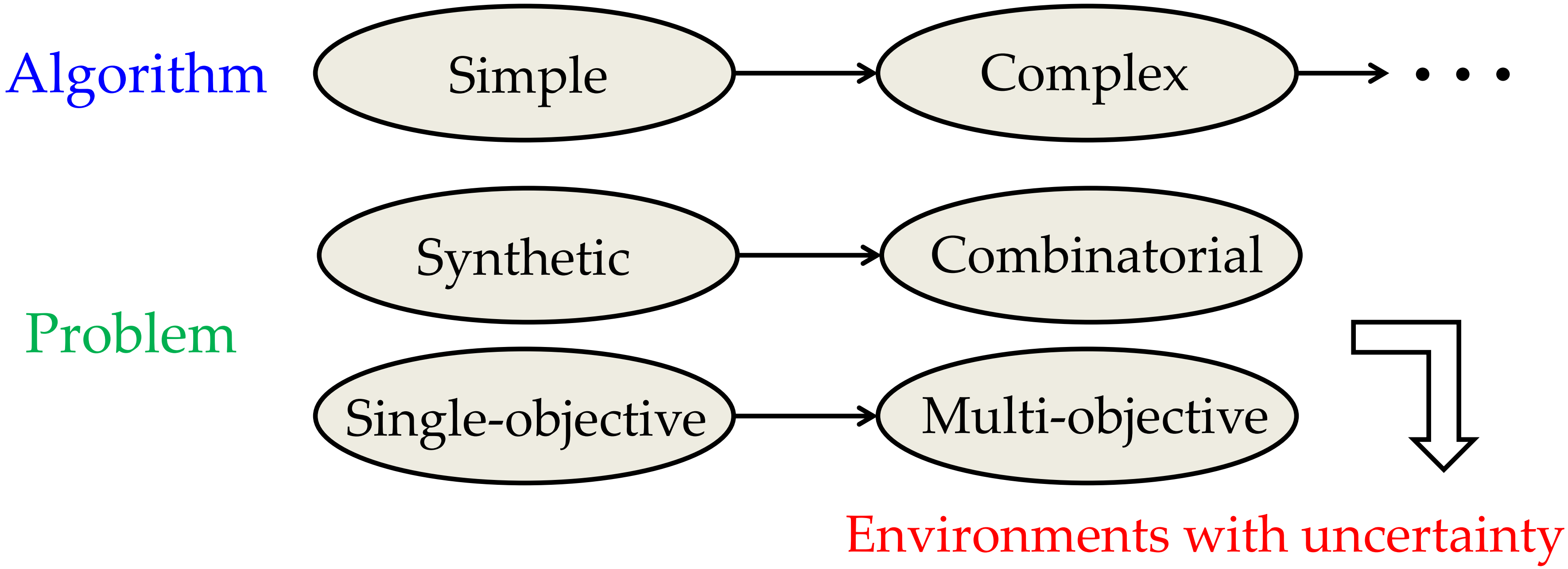
$O\left(m^2 n w_{min}(|C| + \frac{\log n + \log w_{max}}{n w_{min}} - N_{gc}(1 - \frac{1}{m}))\right)$ with 2 approximation

[Qian et al., AIJ'13]

MOEA/D — LOTZ $O(n^2 \log n)$; COCZ $\Theta(n \log n)$ [Li et al., TEC'16]

Theoretical analysis of EAs

Running time analysis



Theory community

Ingo Wegener (1950-2008), TU Dortmund, Germany

Stefan Droste, Thomas Jansen, Ingo Wegener:
A Rigorous Complexity Analysis of the (1 + 1) Evolutionary Algorithm for Separable Functions with Boolean Inputs. *Evolutionary Computation* 6(2): 185-196 (1998)

UK	University of Sheffield: Pietro Oliveto University of Birmingham: Per Kristian Lehre Aberystwyth University: Thomas Jansen
Germany	University of Potsdam: Tobias Friedrich University of Passau: Dirk Sudholt
France	Ecole Polytechnique: Benjamin Doerr Sorbonne University: Carola Doerr
Switzerland	ETH Zürich: Johannes Lengler
Denmark	Technical University of Denmark: Carsten Witt
Australia	University of Adelaide: Frank Neumann

China

Nanjing University

Sun Yat-sen University

Southern University of Science and Technology

Summary

- Fitness level
  - Original fitness level
  - Refined fitness level
- Drift analysis
  - Additive drift
  - Multiplicative drift
  - Negative drift
- Switch analysis
- Results of running time analysis

References

- C. Bian, C. Qian and K. Tang. A general approach to running time analysis of multi-objective evolutionary algorithms. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, Stockholm, Sweden, 2018.

- D. C. Dang and P. K. Lehre. Runtime analysis of non-elitist populations: From classical optimisation to partial information. *Algorithmica,* 2016, 75(3): 428–461

- B. Doerr, D. Johannsen and C. Winzen. Multiplicative drift analysis. *Algorithmica*, 2012, 64: 673-697

- S. Droste, T. Jansen and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 2002, 276(1-2): 51-81

- O. Giel. Expected runtimes of a simple multiobjective evolutionary algorithm. In *Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC'03)*, 2003, pages 1918–1925, Canberra, Australia

References

- B. Hajek. Hitting time and occupation time bounds implied by drift analysis with applications. *Advances in Applied Probability,* 1982, 14(3): 502–525
- J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 2001, 127(1): 57-85
- M. Laumanns, L. Thiele and E. Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 2004, 8(2): 170-182
- F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science*, 2007, 378(1): 32-40
- P. S. Oliveto and C. Witt. Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica*, 2011, 59(3): 369-386

References

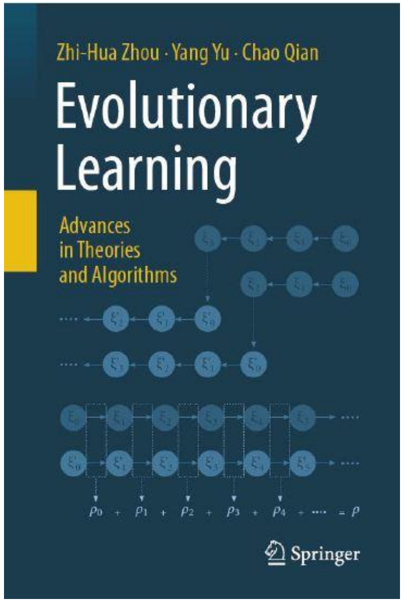
- P. S. Oliveto and C. Witt. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*, 2014, 545: 2-19
- C. Qian, Y. Yu and Z.-H. Zhou. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence*, 2013, 204: 99-119
- J. E. Rowe and D. Sudholt. The choice of the offspring population size in the $(1,\lambda)$ evolutionary algorithm. *Theoretical Computer Science*, 2014, 545: 20-38
- D. Sudholt. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2013, 17(3): 418-435
- Y. Yu, C. Qian and Z.-H. Zhou. Switch analysis for running time analysis of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2015, 19(6): 777-792

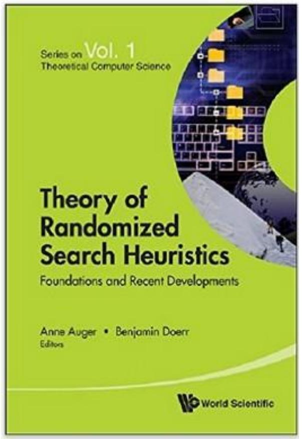
Reading books

Bioinspired Computation in Combinatorial Optimization

Algorithms and Their Computational Complexity

Authors: **Neumann**, Frank, **Witt**, Carsten





Theory of Randomized Search Heuristics

Foundations and Recent Developments

Edited by: **Anne Auger** (*INRIA, France*), **Benjamin Doerr** (*Max-Planck-Institut für Informatik, Germany*)

Assignment - 3

Task:

- Apply fitness level method to analyze the upper bound on the expected running time of (1+1)-EA solving LeadingOnes
- Apply multiplicative and additive drift analysis to analyze the upper bound on the expected running time of (1+1)-EA solving OneMax
- Analyze the upper bound on the expected running time of (1+1)-EA solving BinVal

Deadline: Dec. 17　　　　共5题，详见课程网站