Genetic Algorithm for the Weight Maximization Problem on Weighted Automata

GECCO 2020

ELENA GUTIÉRREZ, TAKAMASA OKUDONO, MASAKI WAGA & ICHIRO HASUO

institute IMdea software

NII 大学共同利用機関法人 情報・システム研究機構 国立情報学研究所 National Institute of Informatics

Overview

Context

Solution

Problem

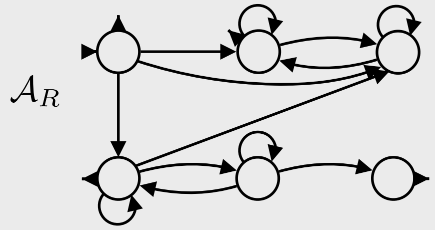
Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

E. Gutiérrez, IMDEA Software Institute, Spain

Overview

Context



Weighted
Automaton
over **reals**

Problem

Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

Solution

Overview

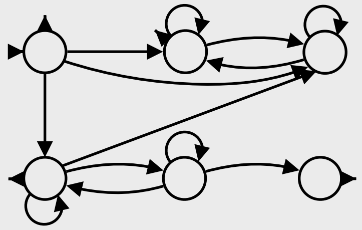
Context

f

Real-valued function

Automata Learning

$\mathcal{A}_R$



Weighted Automaton over **reals**

Problem

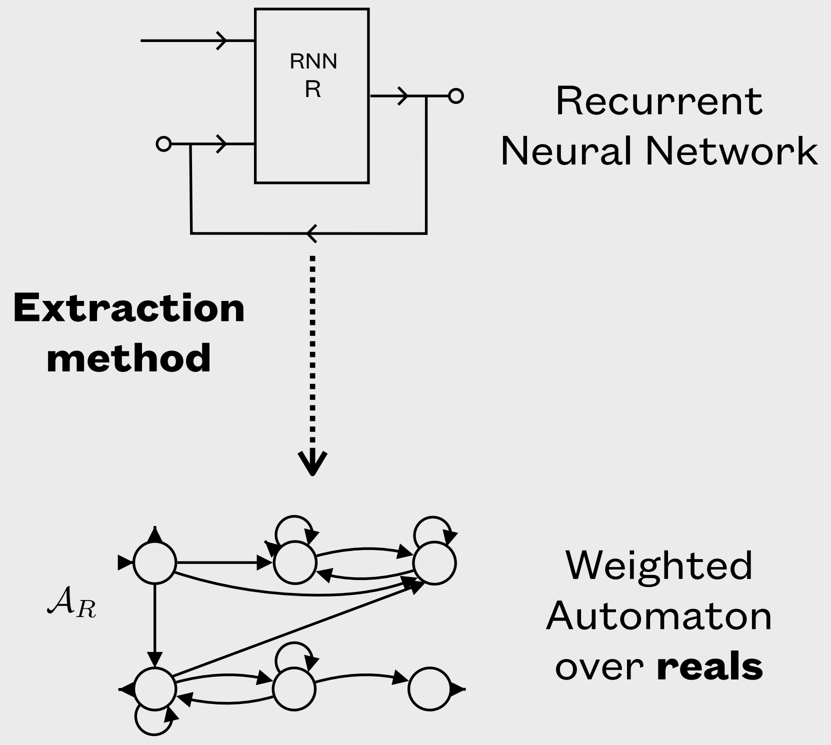
Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

Solution

Overview

Context



[Okudono et al., AAAI 2020]

Problem

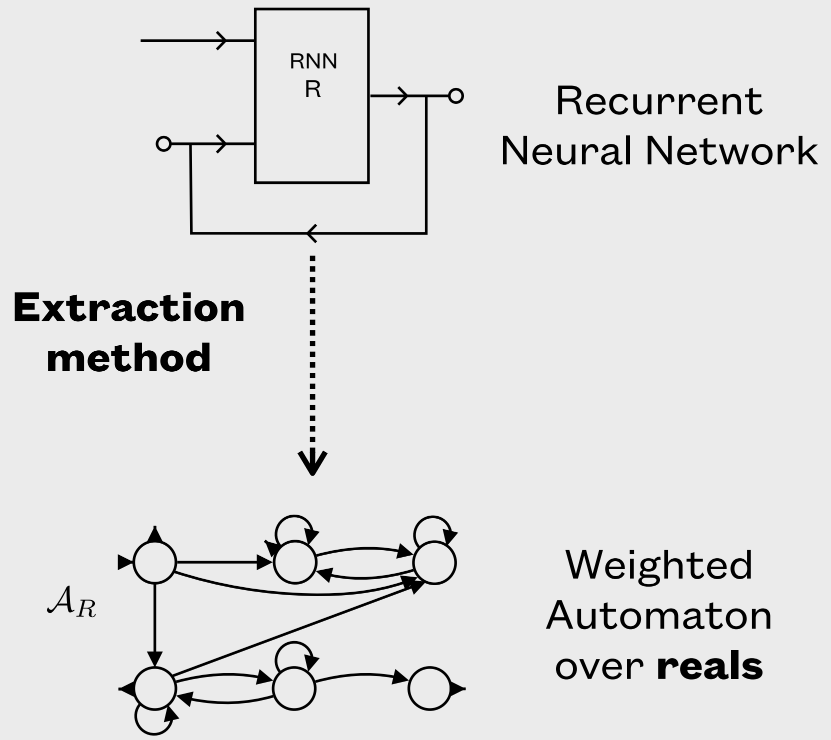
Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

Solution

Overview

Context



[Okudono et al., AAAI 2020]

Problem

NP-COMPLETE
(bounded decisional version)

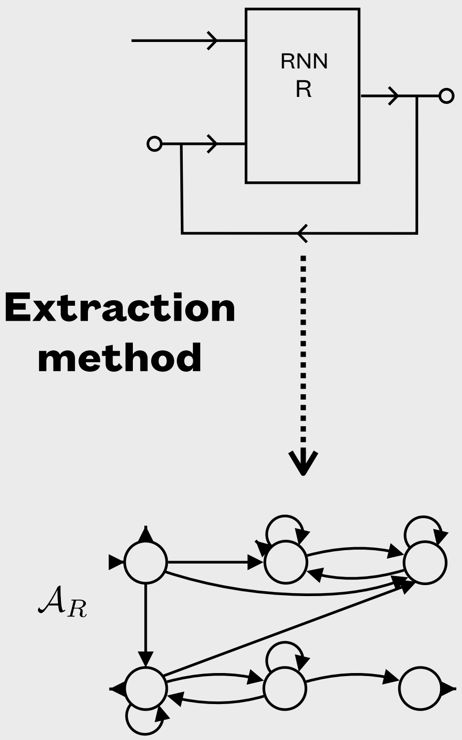
Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

Solution

Overview

Context



Recurrent Neural Network

Extraction method

Weighted Automaton over **reals**

[Okudono et al., AAAI 2020]

Problem

NP-COMPLETE

(bounded decisional version)

Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

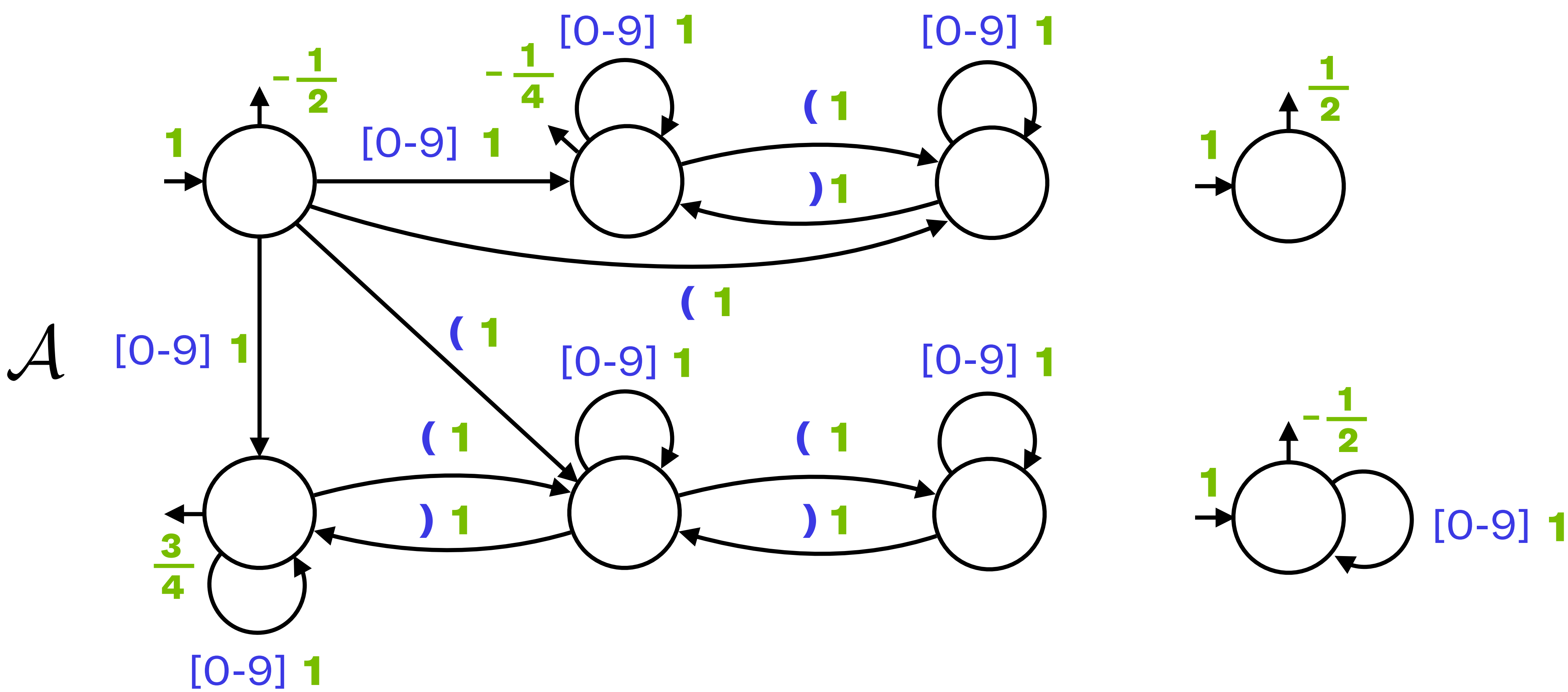
Solution

Genetic Algorithm for approximate solutions

Weighted Automata

Transition system representation

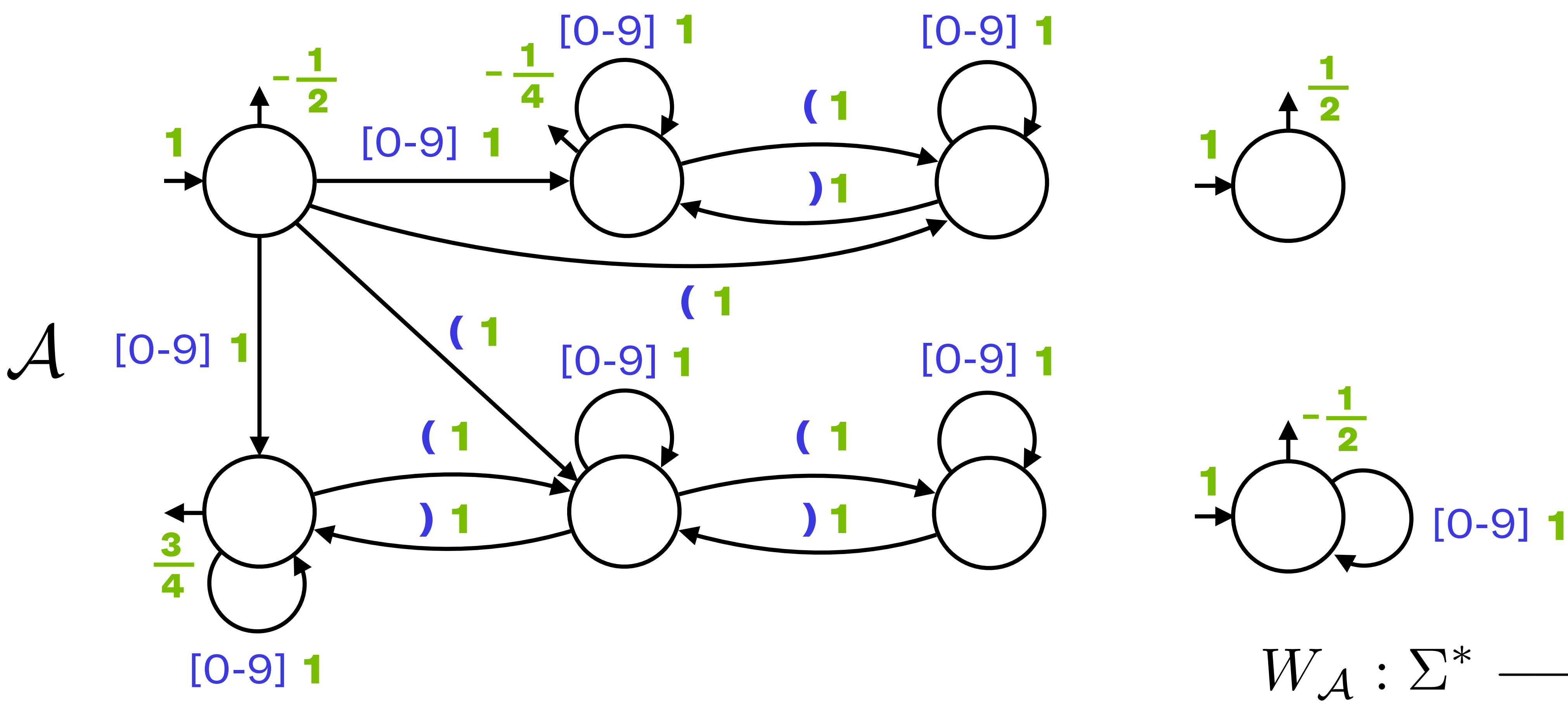
Weighted Finite-state automata: input alphabet $\Sigma = \{$ **(**, **)**, 0,...,9 $\}$



Weighted Automata

Transition system representation

Weighted Finite-state automata: input alphabet $\Sigma = \{$ (,), 0,…,9 $\}$



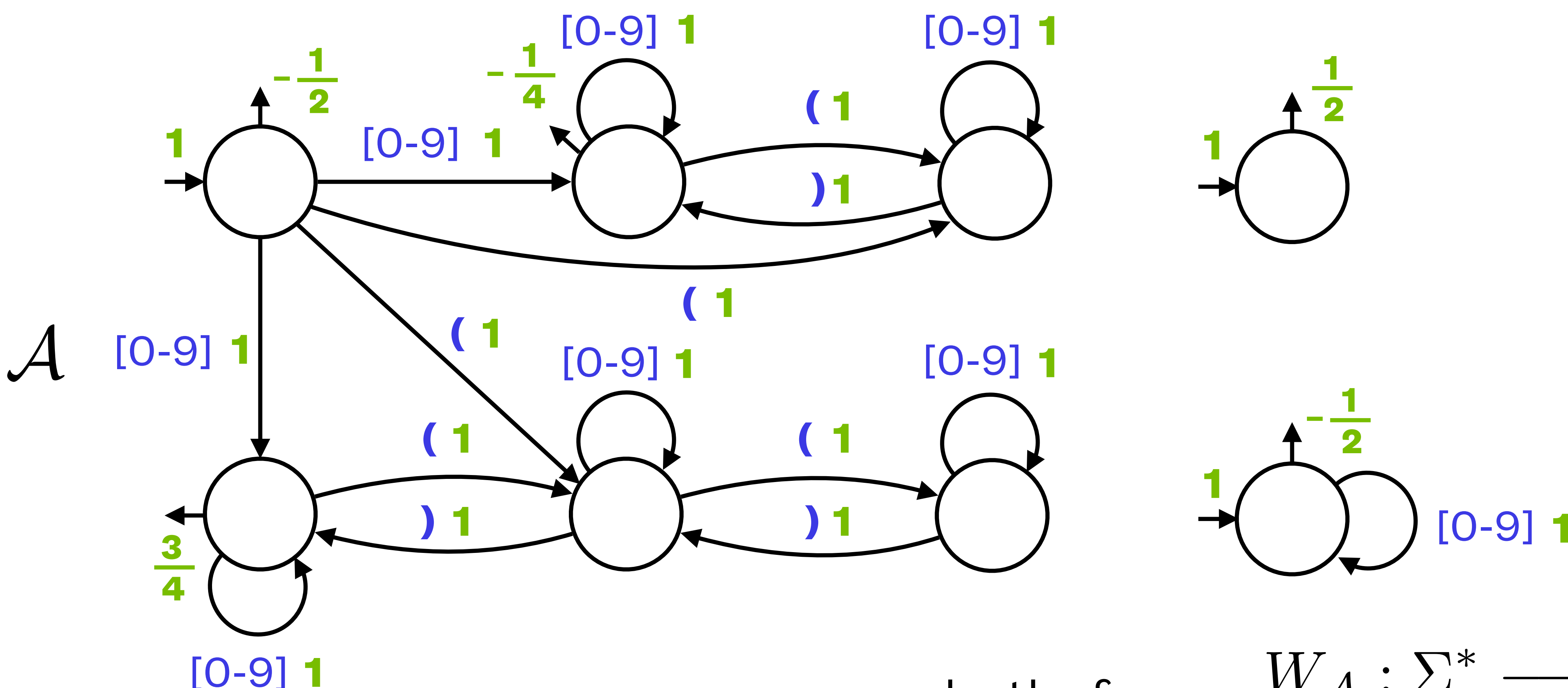
$W_{\mathcal{A}} : \Sigma^* \longrightarrow \mathbb{R}$

(3)	$\frac{1}{2}$
((47))(59)	$\frac{3}{4}$
(((5)))	0
9	0

Weighted Automata

Transition system representation

Weighted Finite-state automata: input alphabet $\Sigma = \{$ **(**, **)**, 0,...,9 $\}$



$$W_{\mathcal{A}} : \Sigma^* \longrightarrow \mathbb{R}$$

depth of w

$$W_{\mathcal{A}}(w) = \begin{cases} 1 - 2^{-D(w)} & \text{if } w \text{ is well-parenthesized and } D(w) \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

(3)	$\frac{1}{2}$
((47))(59)	$\frac{3}{4}$
(((5)))	0
9	0

Weighted Automata

Weighted Finite-state automata: input alphabet $\Sigma = \{\,(,\ ),\ 0,\ldots,9\,\}$

$$\mathcal{A} = (Q, \Sigma, \{M_a\}_{a \in \Sigma}, \mathrm{i}, \mathrm{f})$$

initial vector $\mathrm{i} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$

final vector $\mathrm{f} = \begin{pmatrix} -\frac{1}{2} & -\frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$

transition matrices:

$$M_{(} = \begin{matrix} 0&0&1&0&1&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{matrix} \qquad M_{)} = \begin{matrix} 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \end{matrix} \qquad M_{[0-9]} = \begin{matrix} 0&1&0&1&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1 \end{matrix}$$

Weighted Automata

Weighted Finite-state automata: input alphabet $\Sigma = \{$ **(**, **)**, 0,…,9 $\}$

$$\mathcal{A} = (Q, \Sigma, \{M_a\}_{a\in\Sigma}, \mathrm{i}, \mathrm{f})$$

Weight of a word $w = a_1\, a_2 \cdots a_n$ **:**

$$W_{\mathcal{A}}(w) = \mathrm{i}^\top \cdot \prod_{i=1}^{n} M_{a_i} \cdot \mathrm{f}$$

$$W_{\mathcal{A}}(\mathbf{(}3\mathbf{)}) =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \cdot \begin{matrix} 0\,0\,1\,0\,1\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0 \\ \vdots \\ \boxed{M_(} \\ \vdots \\ 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0 \end{matrix} \cdot \begin{matrix} 0\,1\,0\,1\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0 \\ \vdots \\ \boxed{M_{[0-9]}} \\ \vdots \\ 0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \end{matrix} \cdot \begin{matrix} 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0 \\ \vdots \\ \boxed{M_)} \\ \vdots \\ 0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0 \end{matrix} \cdot \begin{pmatrix} -\frac{1}{2} & -\frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

A Problem on Weighted Automata

Problem

Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$, compute a word $w_0 \in \operatorname*{argmax}_{w \in \Sigma^*} W_{\mathcal{A}}(w)$

Applications

A Problem on Weighted Automata

Problem

Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$, compute a word $w_0 \in \underset{w \in \Sigma^*}{\operatorname{argmax}} W_{\mathcal{A}}(w)$

Applications

- Compute a simple notion of **distance**

$$\operatorname{dist}(\mathcal{A}, \mathcal{B}) = \max_{w \in \Sigma^*} \{(\mathcal{A} \ominus \mathcal{B})(w), (\mathcal{B} \ominus \mathcal{A})(w)\}$$

WFA modelling a system

WFA for the specification of the system

A Problem on Weighted Automata

Problem

Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$, compute a word $w_0 \in \underset{w \in \Sigma^*}{\operatorname{argmax}} \, W_{\mathcal{A}}(w)$

Applications

- Compute a simple notion of **distance**

$$\operatorname{dist}(\mathcal{A}, \mathcal{B}) = \max_{w \in \Sigma^*} \{(\mathcal{A} \ominus \mathcal{B})(w), (\mathcal{B} \ominus \mathcal{A})(w)\}$$

WFA modelling a system

WFA for the specification of the system

  - Maximum “**error**” of $\mathcal{A}$ approximating $\mathcal{B}$

A Problem on Weighted Automata

Problem

Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$, compute a word $w_0 \in \underset{w \in \Sigma^*}{\operatorname{argmax}} W_{\mathcal{A}}(w)$

Applications

- Compute a simple notion of **distance**

$$\mathrm{dist}(\mathcal{A}, \mathcal{B}) = \max_{w \in \Sigma^*} \left\{ (\mathcal{A} \ominus \mathcal{B})(w), (\mathcal{B} \ominus \mathcal{A})(w) \right\}$$

WFA modelling a system

WFA for the specification of the system

  - Maximum "**error**" of $\mathcal{A}$ approximating $\mathcal{B}$
  - Identify **misclassified** input sequences in the RNN

A Problem on Weighted Automata

Problem

Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$, compute a word $w_0 \in \underset{w \in \Sigma^*}{\operatorname{argmax}}\, W_{\mathcal{A}}(w)$

A Problem on Weighted Automata

Problem

Bounded Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$ and $k \in \mathbb{N}$, compute $w_0 \in \operatorname*{argmax}_{w \in \Sigma^{\leq k}} W_{\mathcal{A}}(w)$

A Problem on Weighted Automata

Problem

Bounded Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$ and $k \in \mathbb{N}$, compute $w_0 \in \underset{w \in \Sigma^{\leq k}}{\operatorname{argmax}} W_{\mathcal{A}}(w)$

A Problem on Weighted Automata

Problem

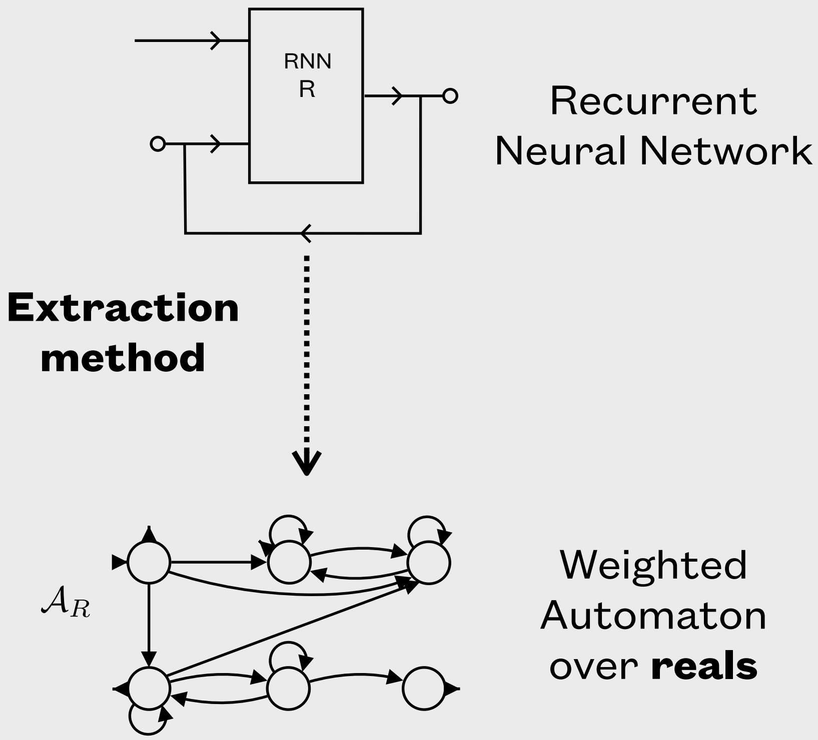
Bounded Weight Maximization Problem on Weighted Automata

Given a WFA $\mathcal{A}$ over $\mathbb{R}$ and $k \in \mathbb{N}$, compute $w_0 \in \underset{w \,\in\, \Sigma^{\leq k}}{\operatorname{argmax}} W_{\mathcal{A}}(w)$

- **Decidable!**... but NP-complete [Gutiérrez et al, GECCO 2020]

Overview

Context



[Okudono et al., AAAI 2020]

Problem

NP-COMPLETE

Bounded Weight Maximization
Problem on Weighted Automata

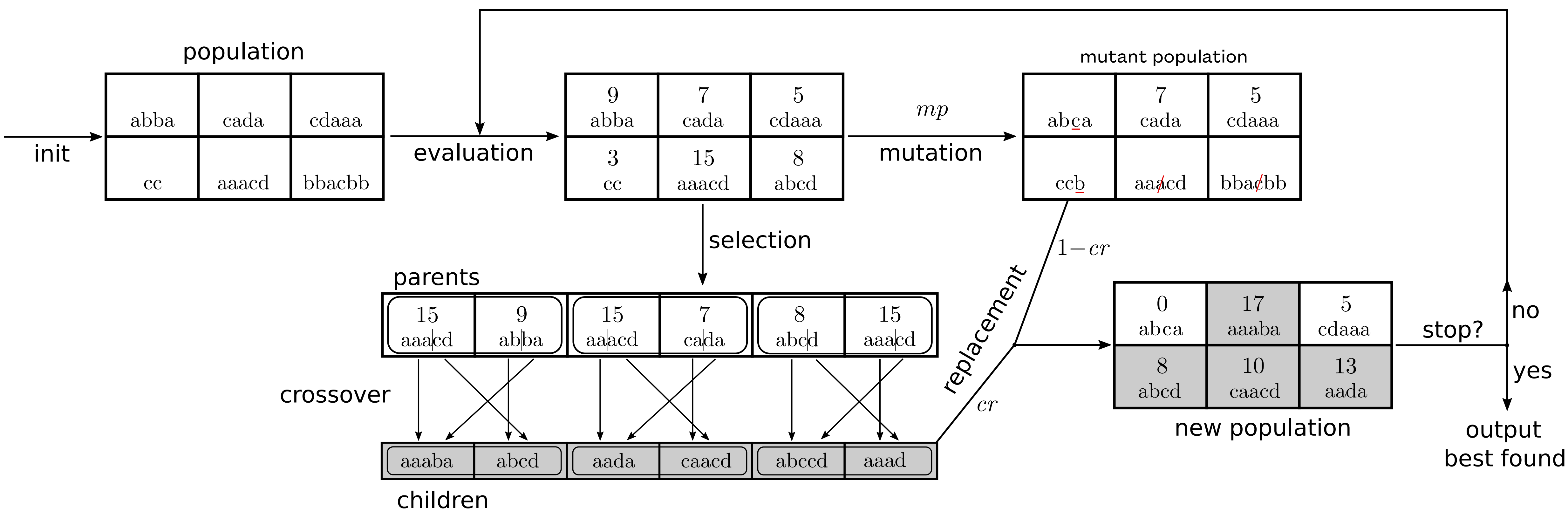
Find the word with the highest weight

Solution

Genetic Algorithm
for approximate
solutions

Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Genetic Representation

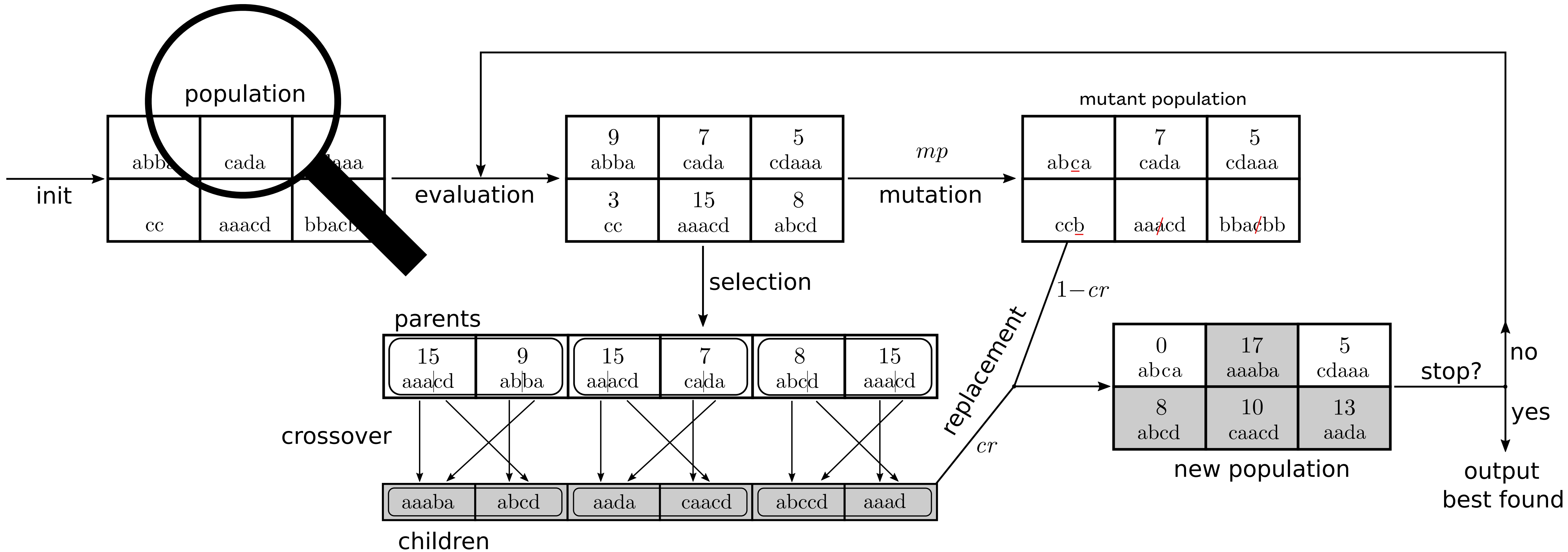
Input: WFA $\mathcal{A}$ (matrix representation) and k $\in \mathbb{N}$

Individuals: words in $\Sigma^{\leq k}$

Fitness Function: weight function $W_{\mathcal{A}}$

Genetic Algorithm for the BWM problem

Bounded **W**eight **M**aximization



Genetic Representation

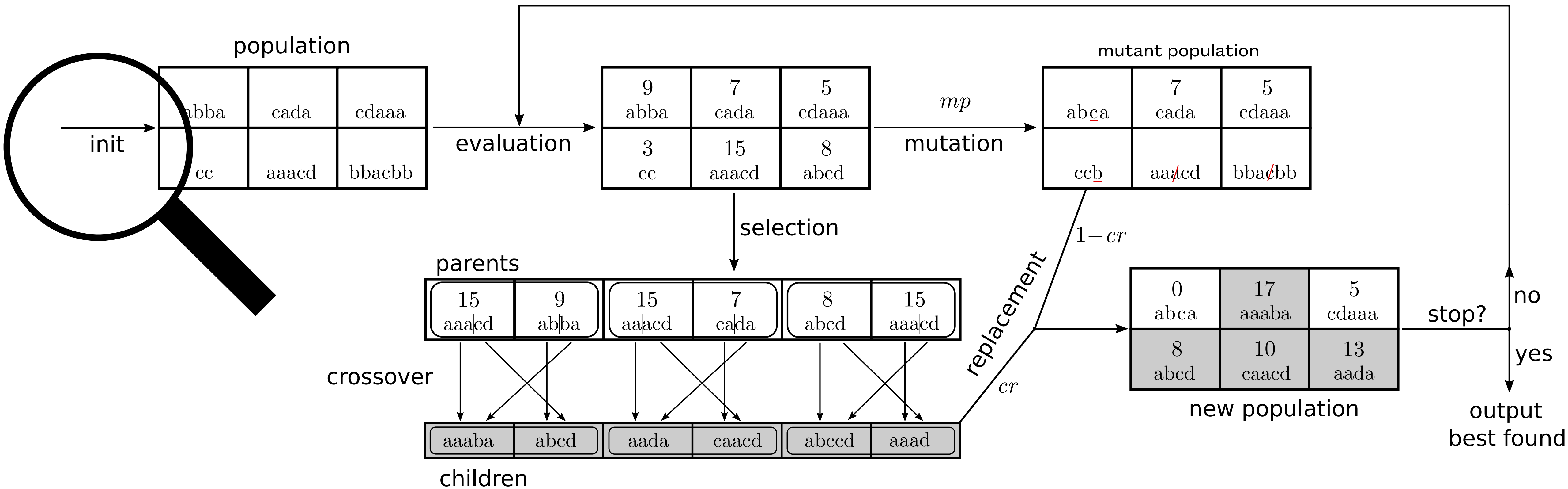
Input: WFA $\mathcal{A}$ (matrix representation) and k $\in \mathbb{N}$

Individuals: words in $\Sigma^{\leq k}$

Fitness Function: weight function $W_{\mathcal{A}}$

Genetic Algorithm for the BWM problem

Bounded Weight Maximization

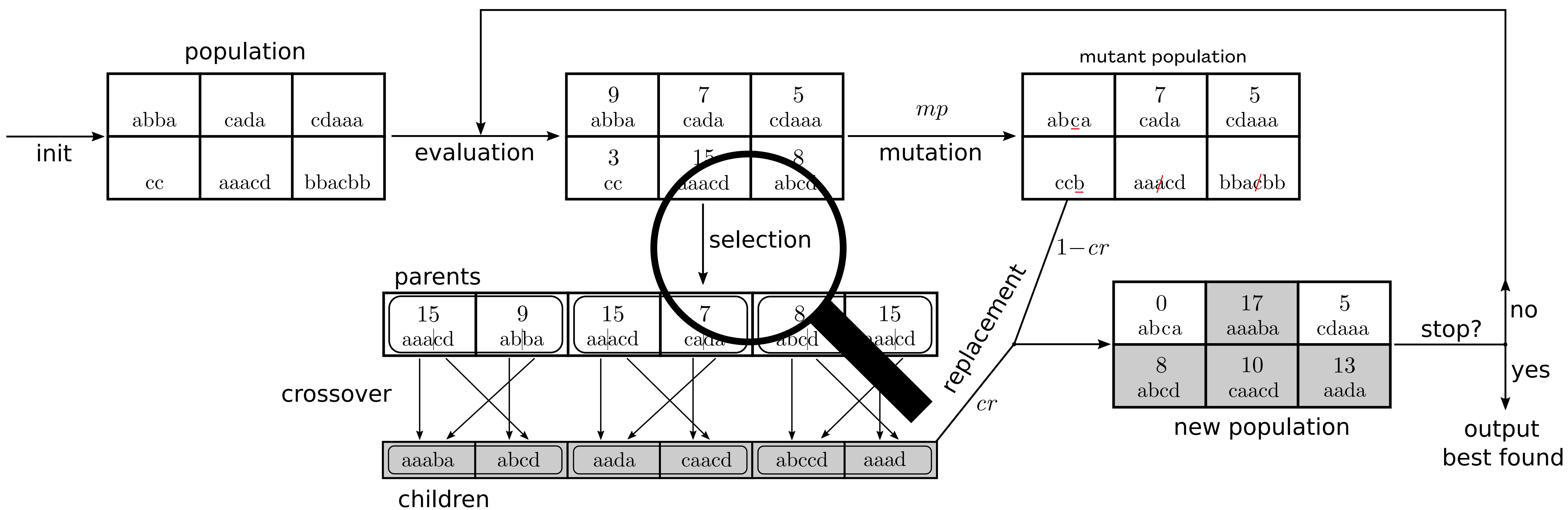


Initialization

N individuals of length ≤ k generated uniformly at random

Genetic Algorithm for the BWM problem

Bounded Weight Maximization



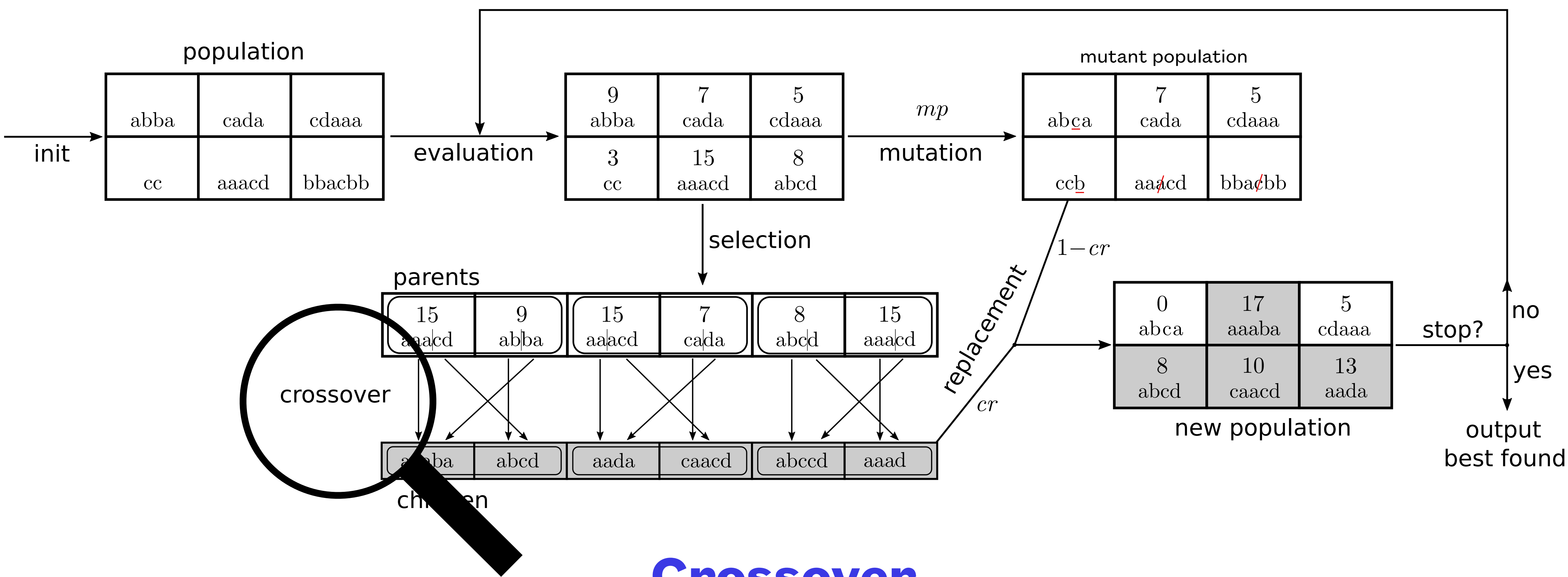
Selection

At the **crossover** and **replacement** steps

Follows a **fitness rank selection**

Genetic Algorithm for the BWM problem

Bounded Weight Maximization



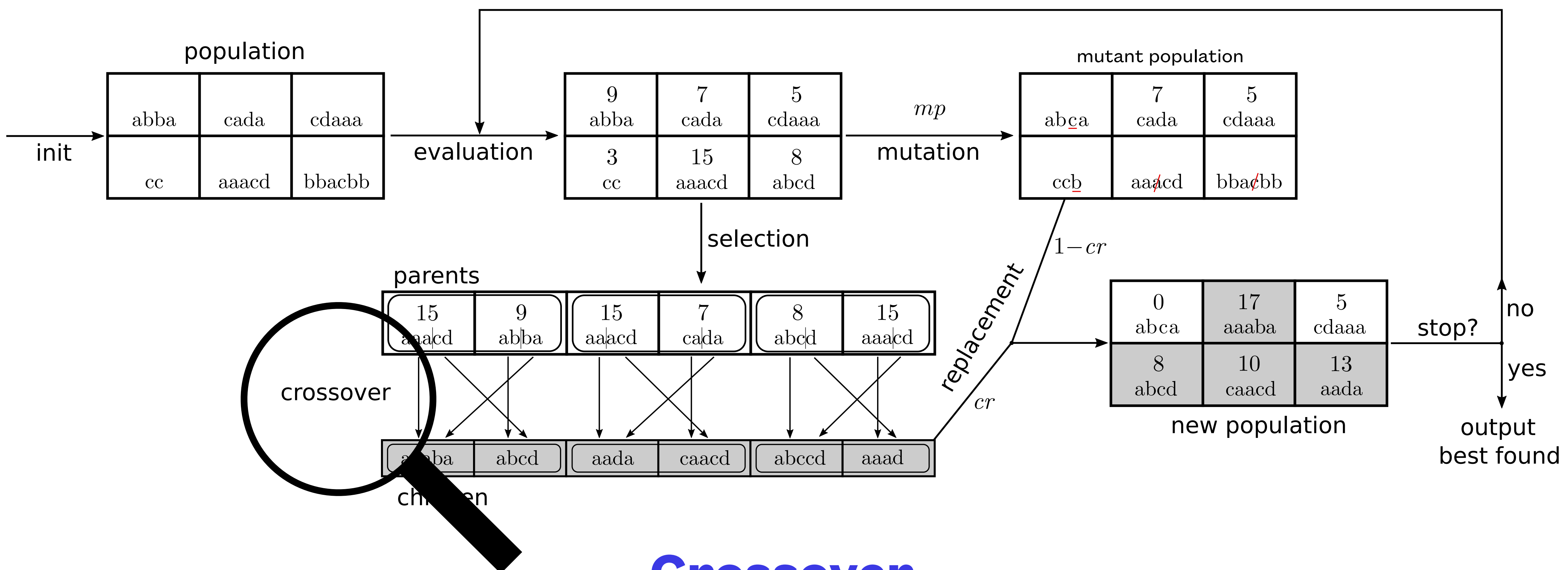
Crossover

Parents | a | a | b | b | b | c | c | c | | v | w | x | y | z |

Similar to single-point crossover

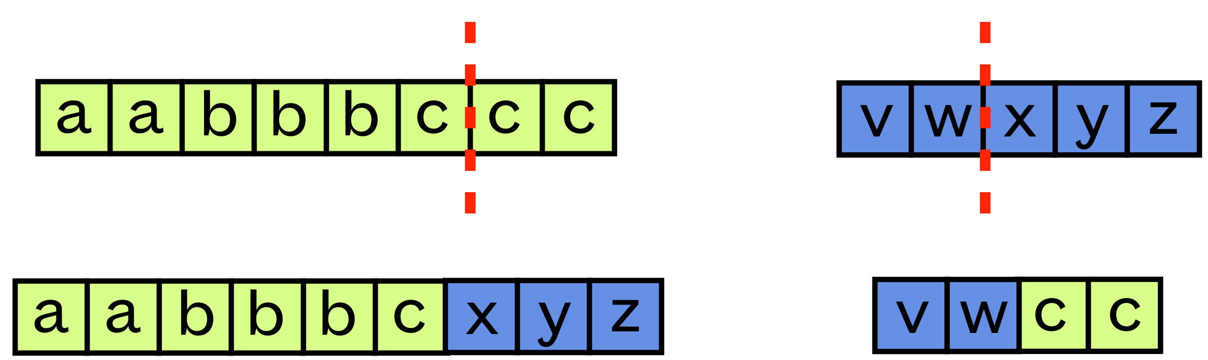
Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Crossover

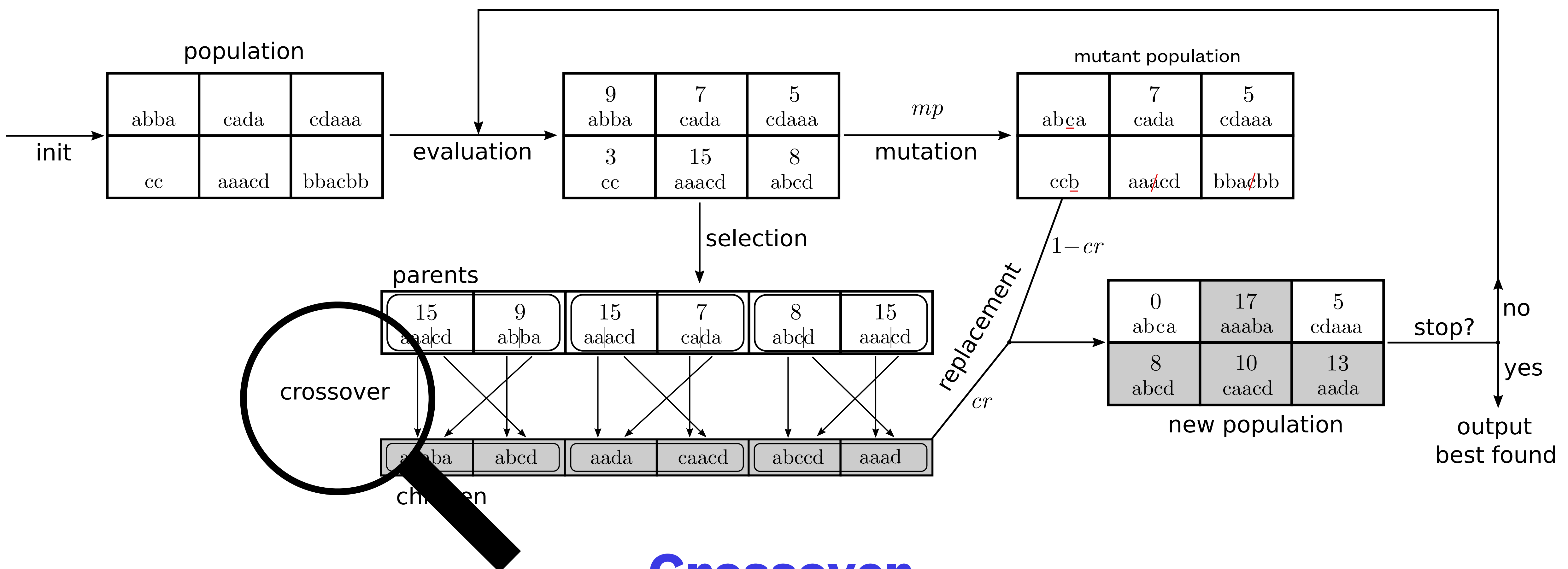
Parents



Similar to single-point crossover

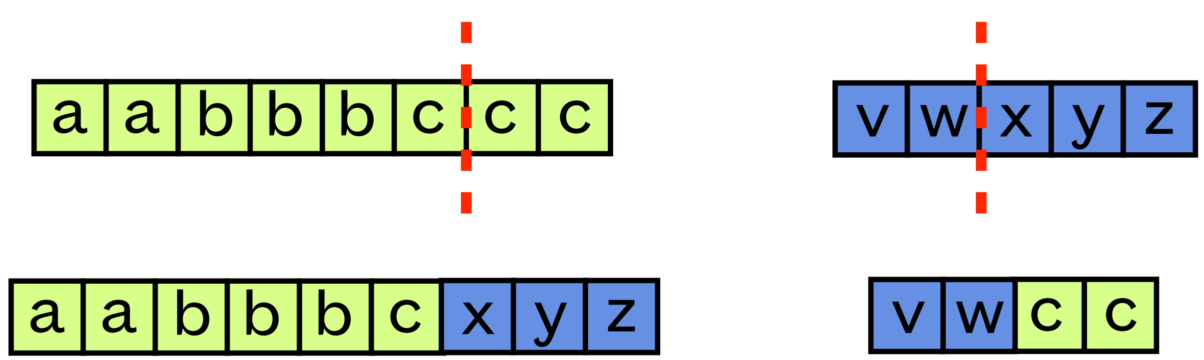
Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Crossover

Parents

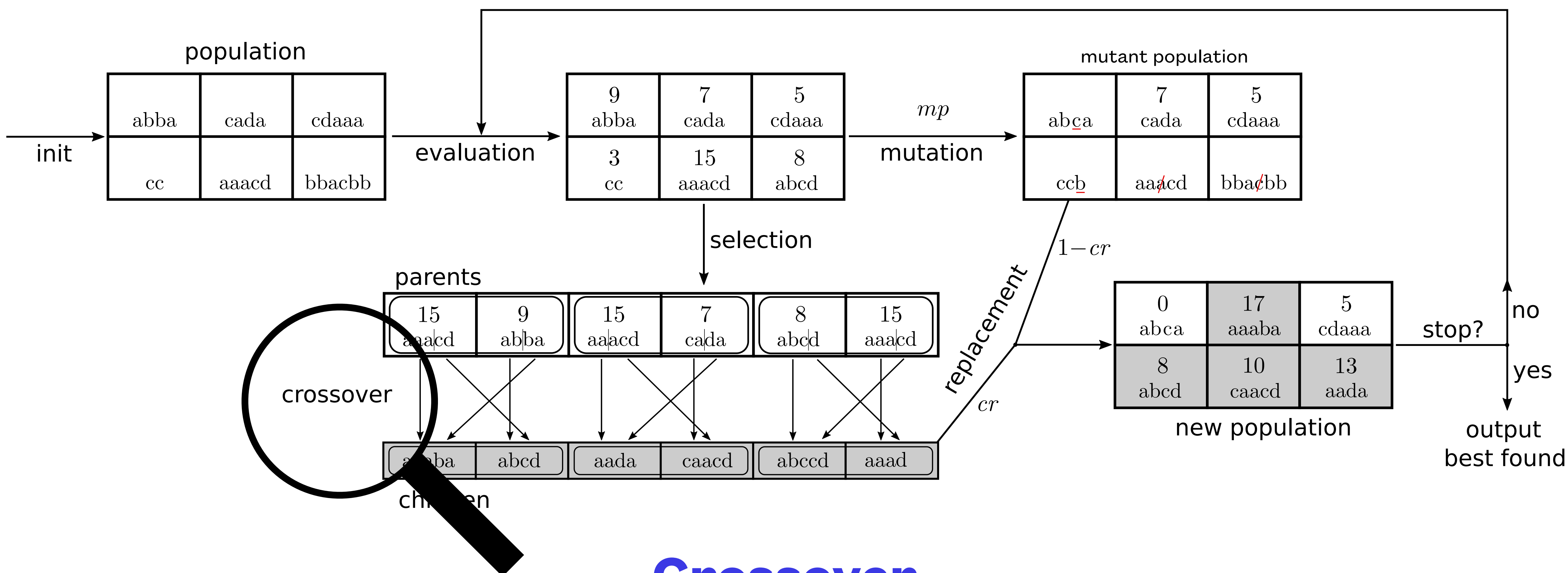


Similar to single-point crossover

E.g., k ≤ 7

Genetic Algorithm for the BWM problem

Bounded Weight Maximization

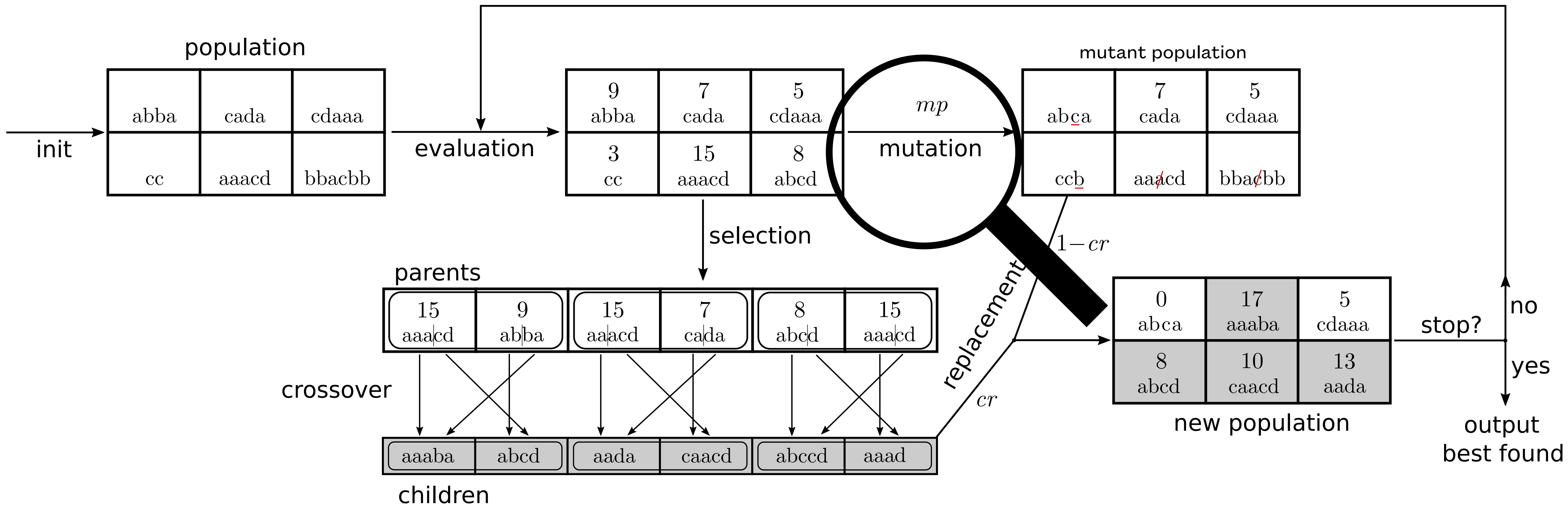


Crossover

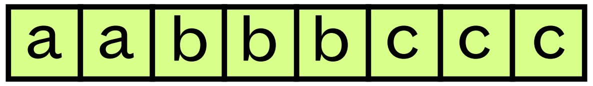
Parents	a a b b b c c c	v w x y z	Similar to single-point crossover
Children	a a b b b c x	v w c c	E.g., k ≤ 7

Genetic Algorithm for the BWM problem

Bounded Weight Maximization

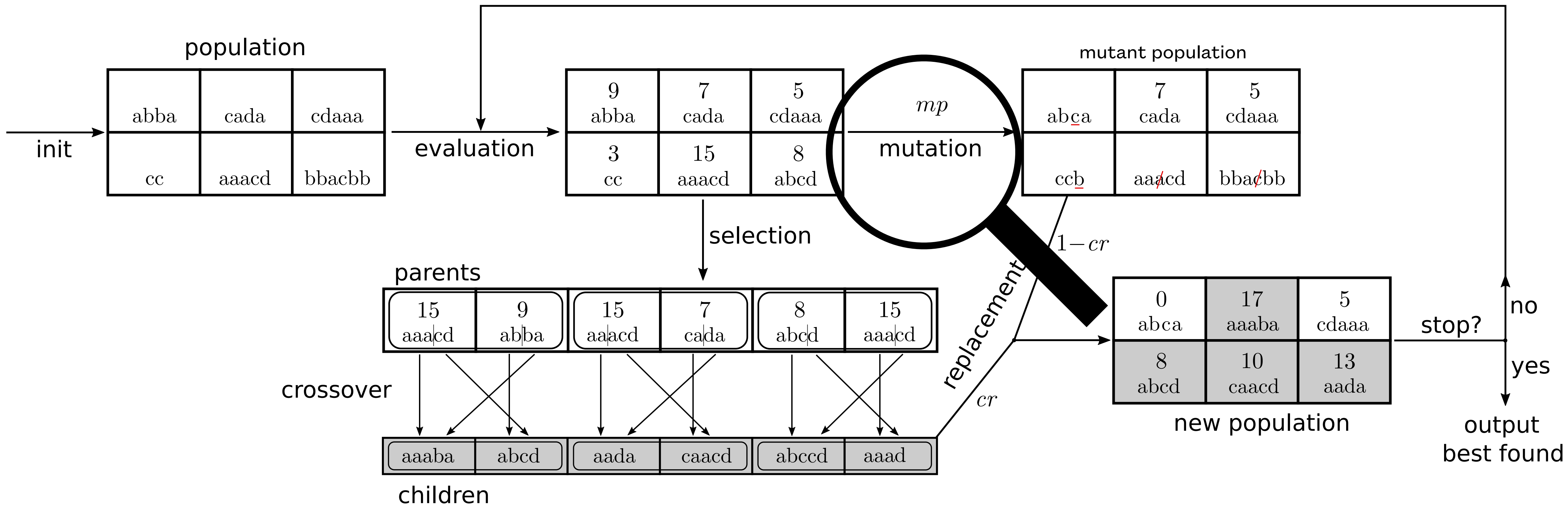


Mutation



Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Mutation

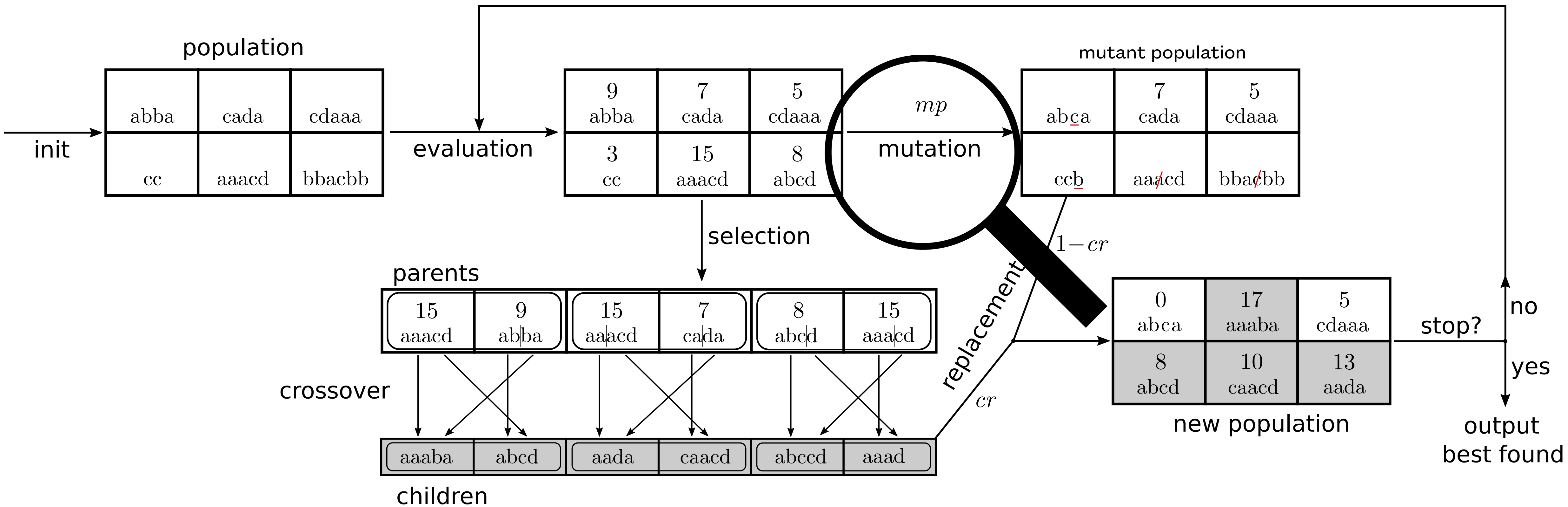
a	a	b	b	b	c	c	c

Sample indices to be mutated using an **exponential distribution**

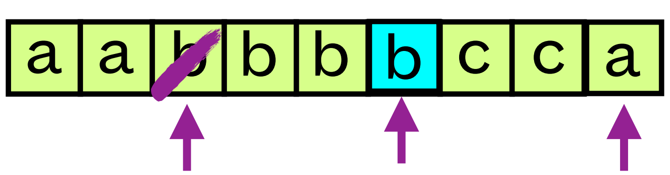
- Deletion
- Replacement
- Insertion

Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Mutation

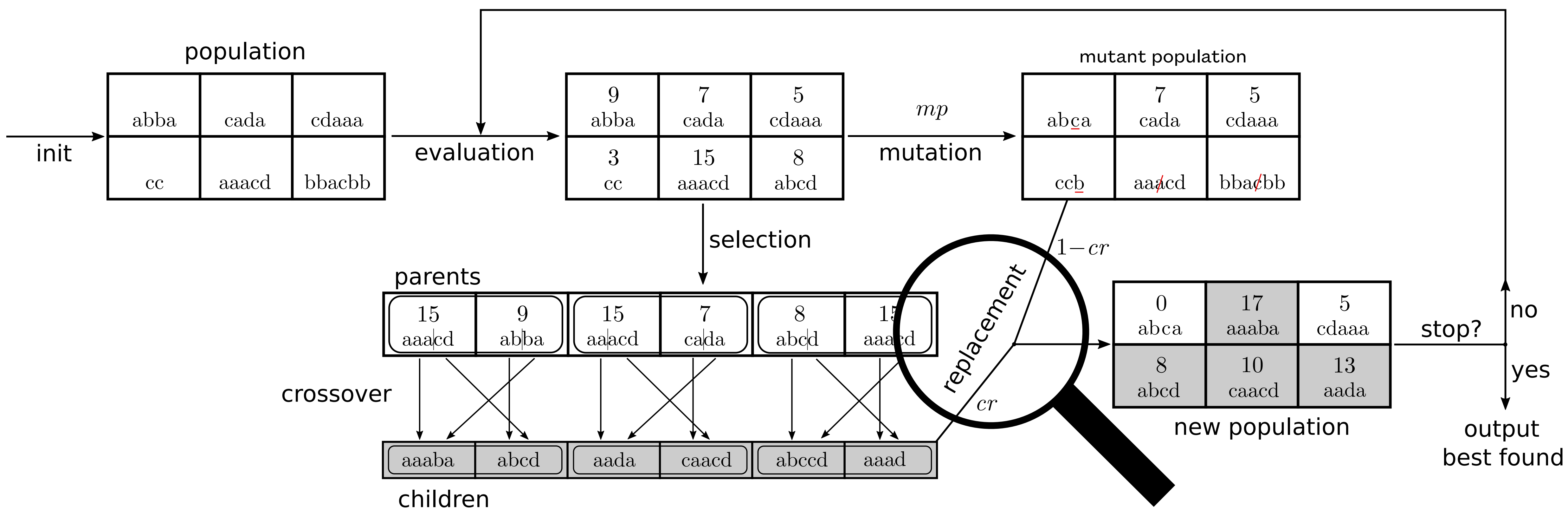


Sample indices to be mutated using an **exponential distribution**

- Deletion
- Replacement
- Insertion

Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Replacement

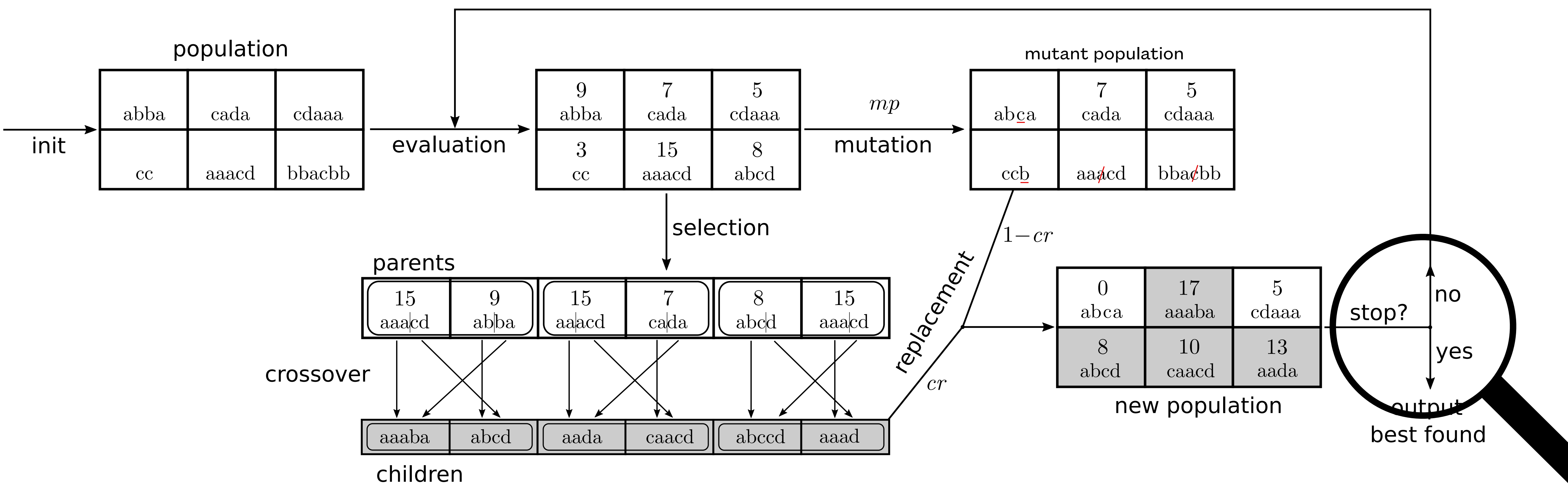
cr: children rate

Select N • cr individuals from the children population

Select N • (1 - cr) individuals from the mutant population

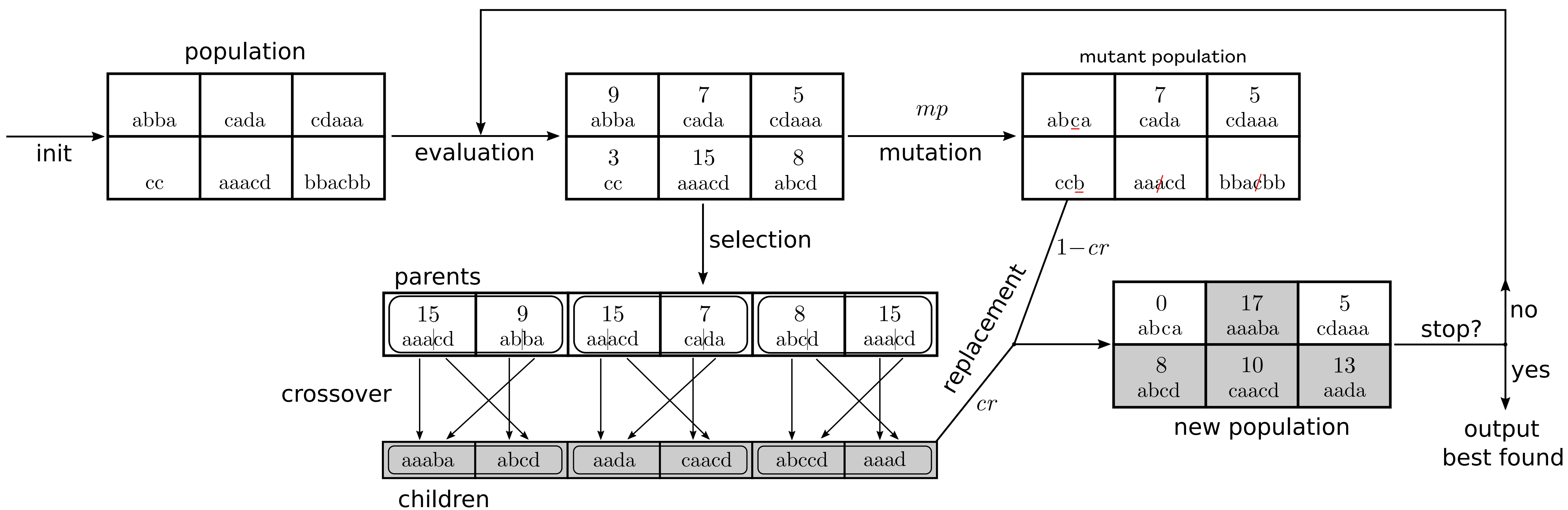
Genetic Algorithm for the BWM problem

Bounded Weight Maximization



Genetic Algorithm for the BWM problem

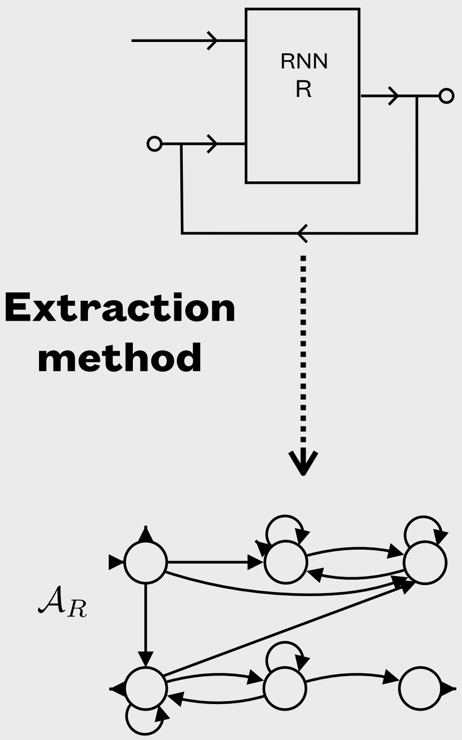
Bounded Weight Maximization



Overview

Code at github.com/elenagutiv/ga-wfas

Context



Recurrent Neural Network

Extraction method

Weighted Automaton over **reals**

[Okudono et al., 2020]

Problem

NP-COMPLETE

Bounded Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

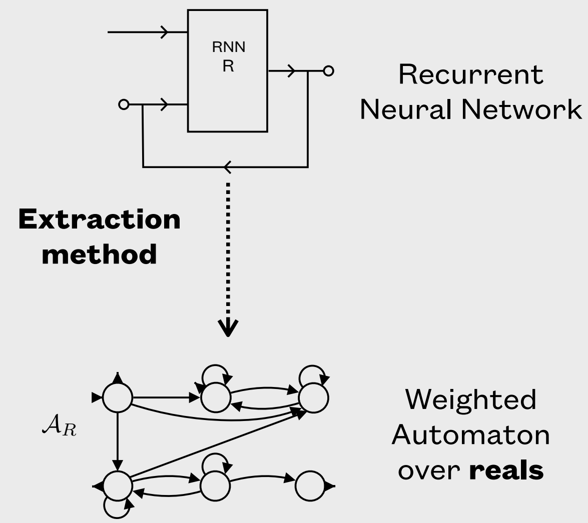
Solution

Genetic Algorithm for approximate solutions

Overview

Code at github.com/elenagutiv/ga-wfas

Context



[Okudono et al., 2020]

Problem

NP-COMPLETE

Bounded Weight Maximization
Problem on Weighted Automata

Find the word with the highest weight

Solution

Genetic Algorithm
for approximate
solutions

① **What is a lower bound on the gain of using memoization?**

② **Is our GA-based meta-heuristic a good fit for the BWMP?**

Empirical evaluation of the algorithm

Experimental Setting: ① **What is a lower bound on the gain of using memoization?**

② **Is our GA-based meta-heuristic a good fit for the BWMP?**

Benchmarks:

- 12 WFAs obtained from RNNs
  - Using the **extraction method** [Okudono et al., 2020]
  - RNNs trained with a set of input-output pairs $(w, W_{\mathcal{A}}(w))$
  - $w \in \Sigma^{\leq 20}, \ |\Sigma| \in \{4, 6, 10\}$
  - $W_{\mathcal{A}} \in [0,1] \cap \mathbb{R}$ is the weight function of a WFA $\mathcal{A}$ **randomly** generated
- 6-25 states

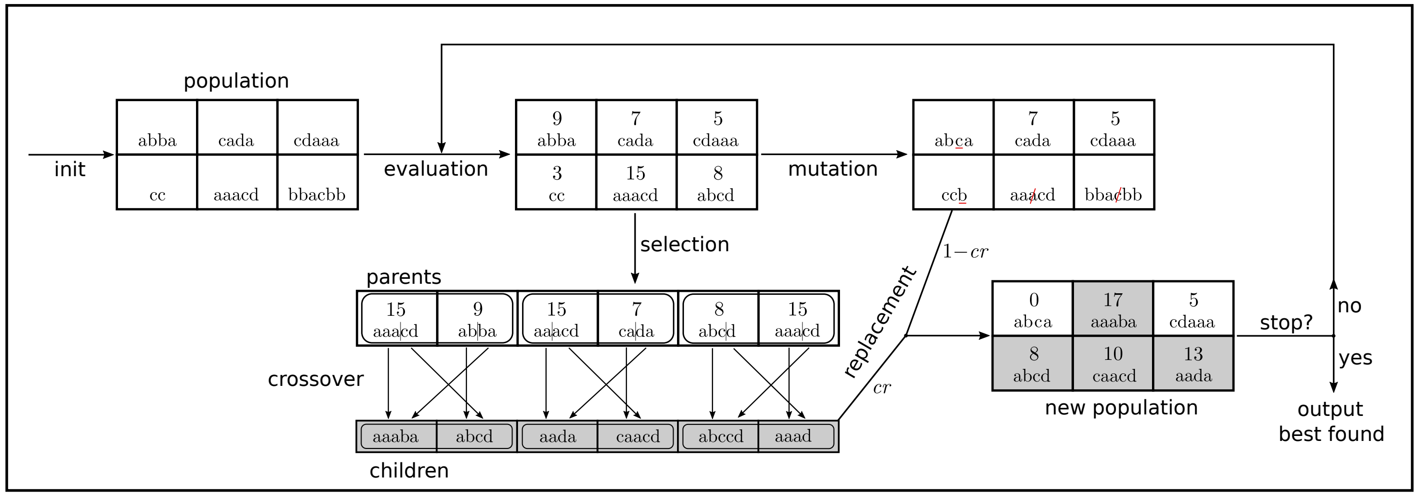
Empirical evaluation of the algorithm

Experimental Setting: ① **What is a lower bound on the gain of using memoization?**

② **Is our GA-based meta-heuristic a good fit for the BWMP?**

Parameters:

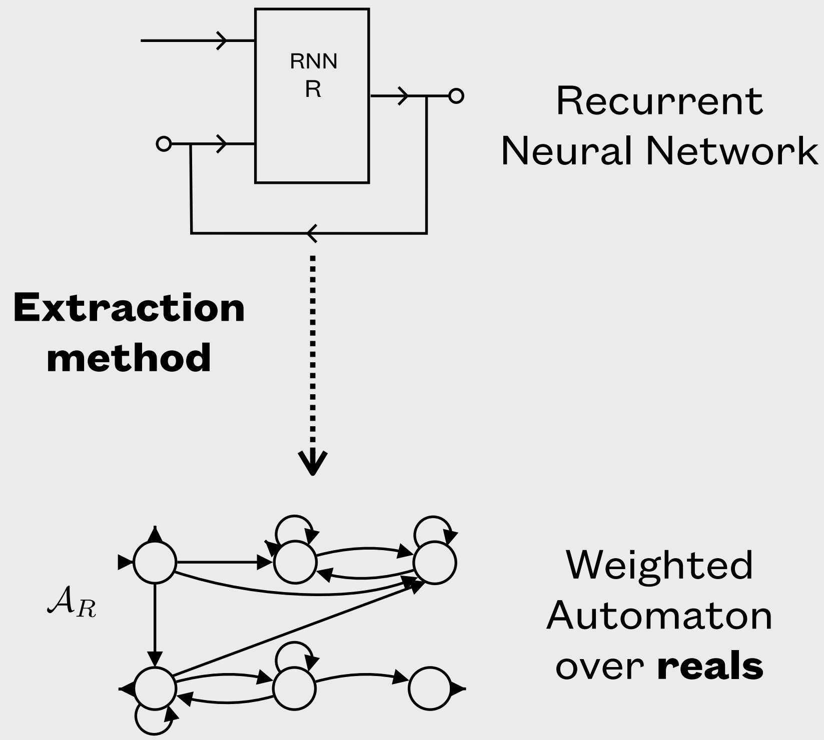
Parameter Values					
k (length bound)	N Size of population	cr children rate	mp mutation probability	λ (exponential distribution)	timeout
20	200	0.8	0.1	0.1	120 s



Overview

Code at github.com/elenagutiv/ga-wfas

Context



[Okudono et al., 2020]

Problem

NP-COMPLETE

Bounded Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

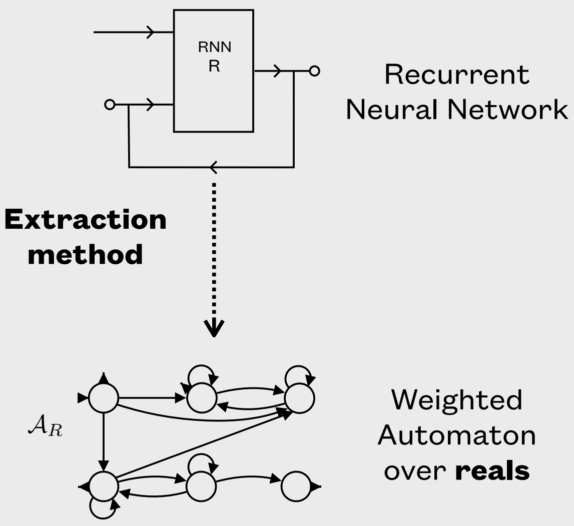
Solution

Genetic Algorithm for approximate solutions

Overview

Code at github.com/elenagutiv/ga-wfas

Context



[Okudono et al., 2020]

Problem NP-COMPLETE

Bounded Weight Maximization **Problem** on Weighted Automata

Find the word with the highest weight

Solution

Genetic Algorithm for approximate solutions

① **What is a lower bound on the gain of using memoization?** **~ 4x more words analyzed per second**

② **Is our GA-based meta-heuristic a good fit for the BWMP?**

Empirical evaluation of the algorithm

② **Is our GA-based meta-heuristic a good fit for the BWMP?**

Parameter Values					
k	Size of population	children rate	mutation probability	λ	timeout
20	200	0.8	0.1	0.1	120 s

Empirical evaluation of the algorithm

② **Is our GA-based meta-heuristic a good fit for the BWMP?**

Parameter Values					
k	Size of population	children rate	mutation probability	λ	timeout
20	200	0.8	0.1	0.1	120 s

Comparison with Random Search

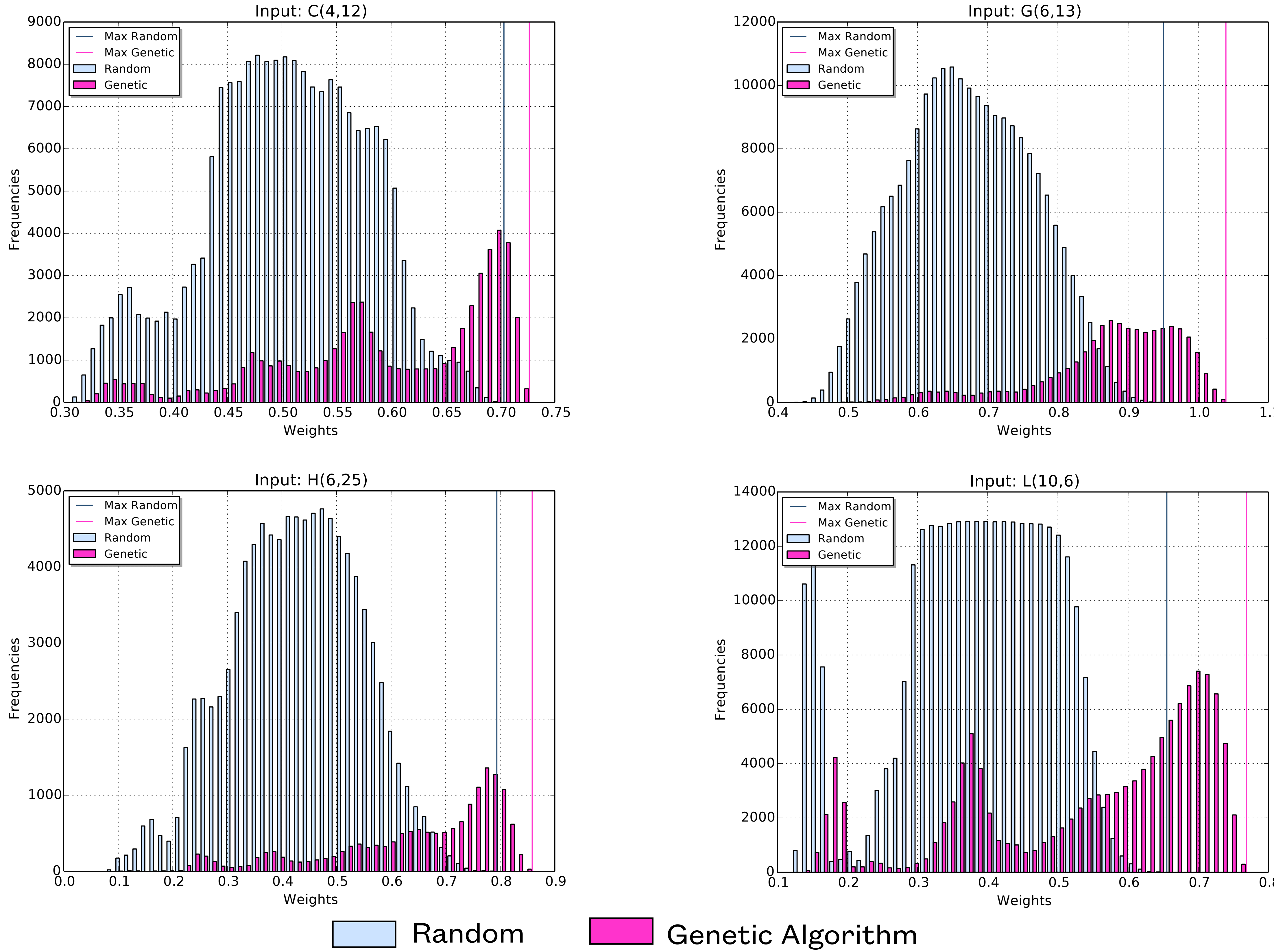
	Random search		Genetic algorithm	
	Weights observed	w/s	Weights observed	w/s
A(4,7)	0.77 ± 0.19 (**1.02**)	14.21k	0.96 ± 0.13 (**1.07**)	1.5k
B(4,11)	0.78 ± 0.17 (**0.98**)	6.22k	0.93 ± 0.12 (**1.03**)	0.76k
C(4,12)	0.5 ± 0.15 (**0.71**)	5.77k	0.59 ± 0.2 (**0.73**)	0.68k
D(4,15)	0.4 ± 0.24 (**0.73**)	3.55k	0.67 ± 0.23 (**0.84**)	0.14k
E(6,9)	0.67 ± 0.2 (**1.04**)	8.65k	0.92 ± 0.22 (**1.17**)	0.95k
F(6,12)	0.46 ± 0.15 (**0.74**)	5.24k	0.63 ± 0.22 (**0.86**)	0.62k
G(6,13)	0.67 ± 0.18 (**0.96**)	4.14k	0.87 ± 0.2 (**1.04**)	0.49k
H(6,25)	0.43 ± 0.23 (**0.79**)	1.21k	0.65 ± 0.31 (**0.86**)	0.14k
I(10,6)	0.47 ± 0.22 (**0.78**)	14.3k	0.76 ± 0.17 (**0.92**)	1.42k
J(10,6)	0.5 ± 0.23 (**0.82**)	13.38k	0.73 ± 0.28 (**0.91**)	1.44k
K(10,6)	0.43 ± 0.16 (**0.66**)	14.26k	0.61 ± 0.19 (**0.77**)	1.5k
L(10,6)	0.38 ± 0.22 (**0.66**)	14.3k	0.55 ± 0.3 (**0.76**)	1.51k

maximum

95% confidence interval

average of 10 executions

② Is our GA-based meta-heuristic a good fit for the BWMP?



Conclusions

- **Genetic**-based meta-heuristic to approximate solutions for the BWM problem
- Benefits from **memoization** of partial executions
- Suitable alternative for WAs from **automata-learning techniques**
- Application for “light-weight” verification of RNNs against weighted **regular** specifications