Interactive one-max problem allows to compare the performance of interactive and human-based genetic algorithms

Derrick Cheng, Alexander Kosorukoff

June 30, 2004

Introduction

- IGA and HBGA are two GAs using human interaction.
- IGA uses human evaluation.
- HBGA brings human innovation into computational process.
- Can HBGA be applied to IGA domain? Will it work better?

The question is not simple

- Human based innovation is better.
- Computational innovation (crossover, mutation) is faster.
- Design an experiment to answer this question.
- Measure how long does it take to achieve the same goal using IGA and HBGA.

Method

- Compare IGAs and HBGAs in two categories: Generational and Steady state, which end up in 4 different cases:
  - IGA-Generational
  - HBGA-Generational
  - IGA-Steady State
  - HBGA-Steady State

Experiment Setting

- The user is given a sequence of 4 algorithms.
- For each case, the task is to achieve the **white color**.
- Each session is logged in the file.
- Session ends when a color crosses the threshold.

RGB values

- Color is a combination of three positive decimal integers: Red, Green, Blue, each in the range of 0 to 255.
- Three RGB values are important for us:
  - Pure Black: B = (0, 0, 0)
  - Pure White: W = (255, 255, 255)
  - White Threshold: T = (245, 245, 245)

Progress Measures

- Use analog of fitness to measure our progress in achieving the **white** color.
- Progress measures are based on Manhattan (L1), Euclidean (L2), Sup (LS) distances.
- Progress measures are constructed according to the formula:

$$\mathbf{M(\mathit{x}) = D(B, W) - D(\mathit{x}, W)}$$

where M(x) = Progress Measure

x = RGB of current color

D = Distance (L1, L2 or LS)

B = RGB of pure black

W = RGB of pure white

D (distance)	M (progress measure)	Examples
	$\boxed{\mathrm{M}(x) = D(B,W) - D(x,W)}$	
L1	$\mathrm{M1}(x) = x_R + x_G + x_B$	$\mathrm{M1}(W) = 765$ $\mathrm{M1}(B) = 0$
L2	$\mathrm{M2}(x) = 255\sqrt{3} - \mathrm{L2}(x,W)$	$\mathrm{M2}(W) = 255\sqrt{3}$ $\mathrm{M2}(B) = 0$
LS	$\mathrm{MS}(x) = \min(x_R, x_G, x_B)$	$\mathrm{MS}(W) = 255$ $\mathrm{M}S(B) = 0$

Progress and termination criteria

- During the session, we log:

$$\mathrm{M1} = \max_{x \in P} \mathrm{M1}(x) \qquad \mathrm{M2} = \max_{x \in P} \mathrm{M2}(x)$$

$$\mathrm{MS} = \max_{x \in P} \mathrm{MS}(x)$$

- Terminate the session when at least one color x in population satisfies:
  MS $(x) \geq$ MS(T) = 245 where $T = 245$

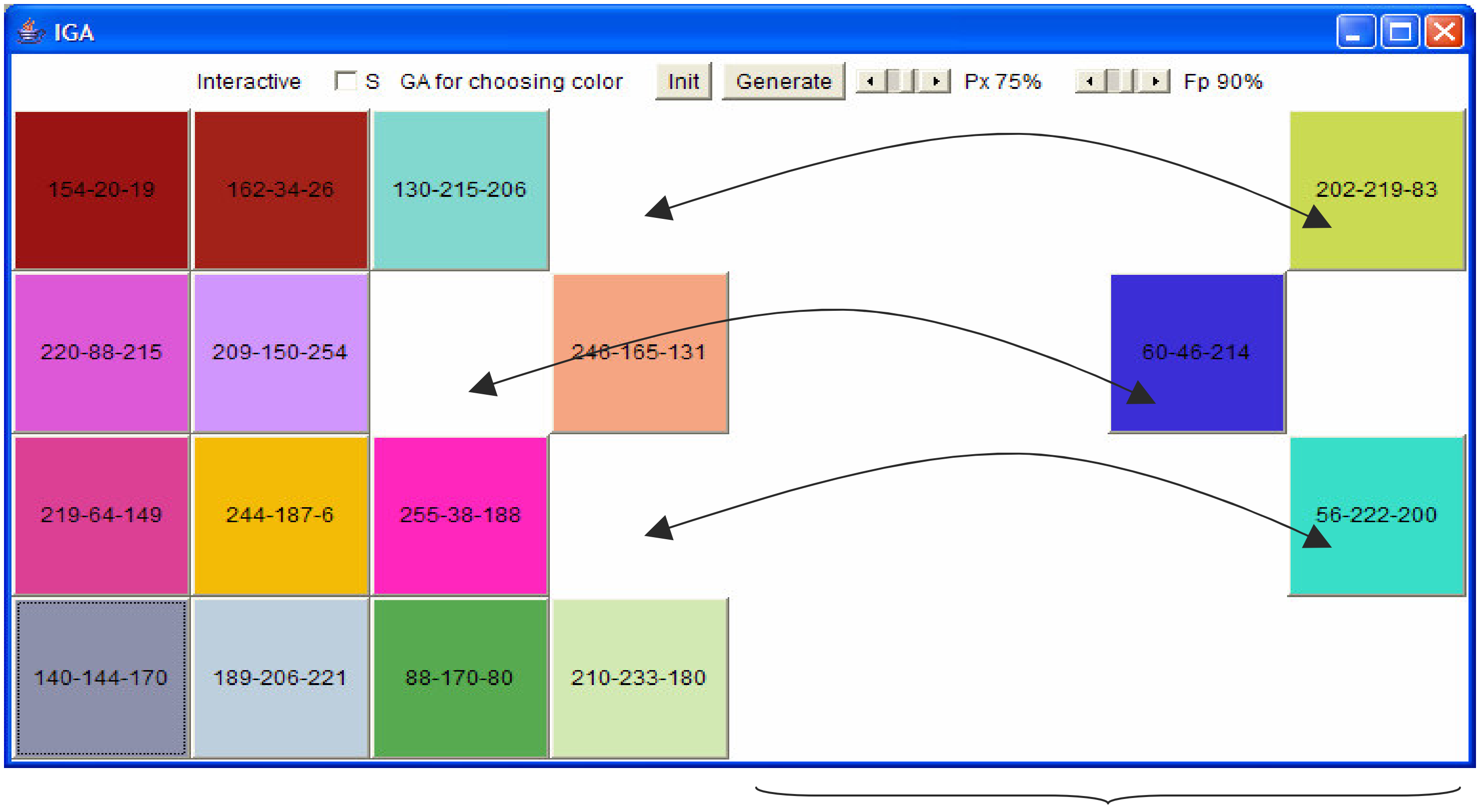
$$\mathrm{MS}(x) = \min(x_R, x_G, x_B)$$

Genetic Algorithm

- Standard binary representation:
  - Pure White: 11111111-11111111-11111111
- Standard parameters
  - Probability of crossover: 0.75
  - Probability of mutation: 1/24
- Fitness-proportional selection
  - Fitness of user-preferred colors: 0.9
  - Fitness of other colors: 0.1

IGA Interface (after initialization)

IGA Interface (during selection)

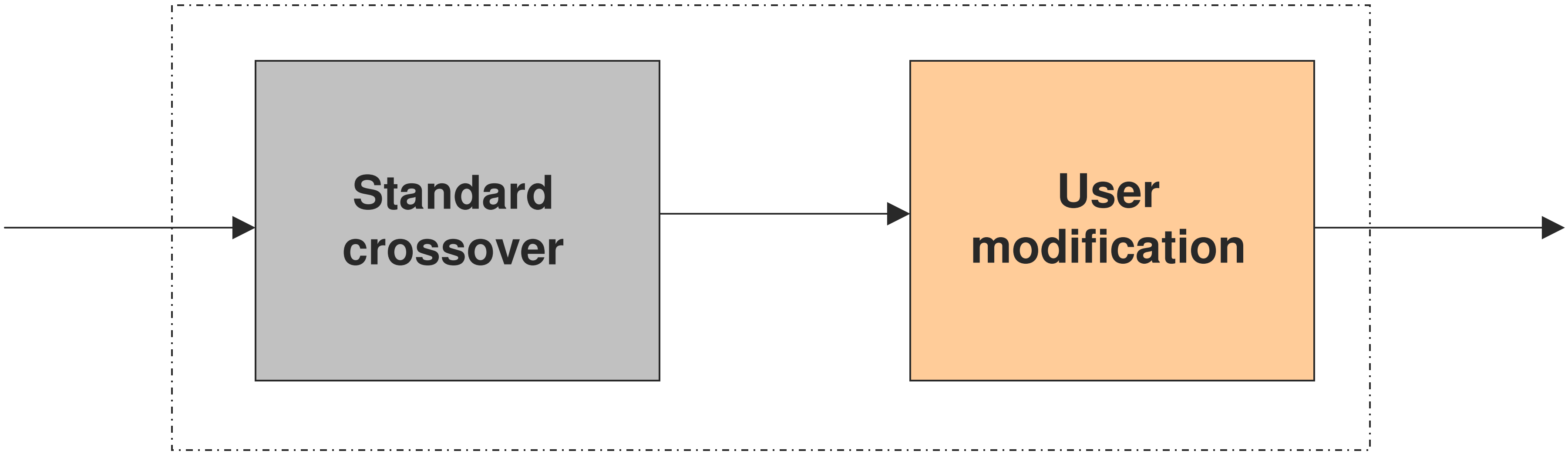


User preferred colors

Comparing IGA and HBGA

	Initialization	Selection	Crossover	Mutation
IGA	Computer	Human	Computer	Computer
HBGA	Computer	Human	**Human/ Computer**	**Human/ Computer**

Human-Based Crossover



Human-Based Crossover

Standard Crossover

Parent 1: 00 | 000000

Parent 2: 11 | 111111

Child: 00 111111

Human-Based Crossover

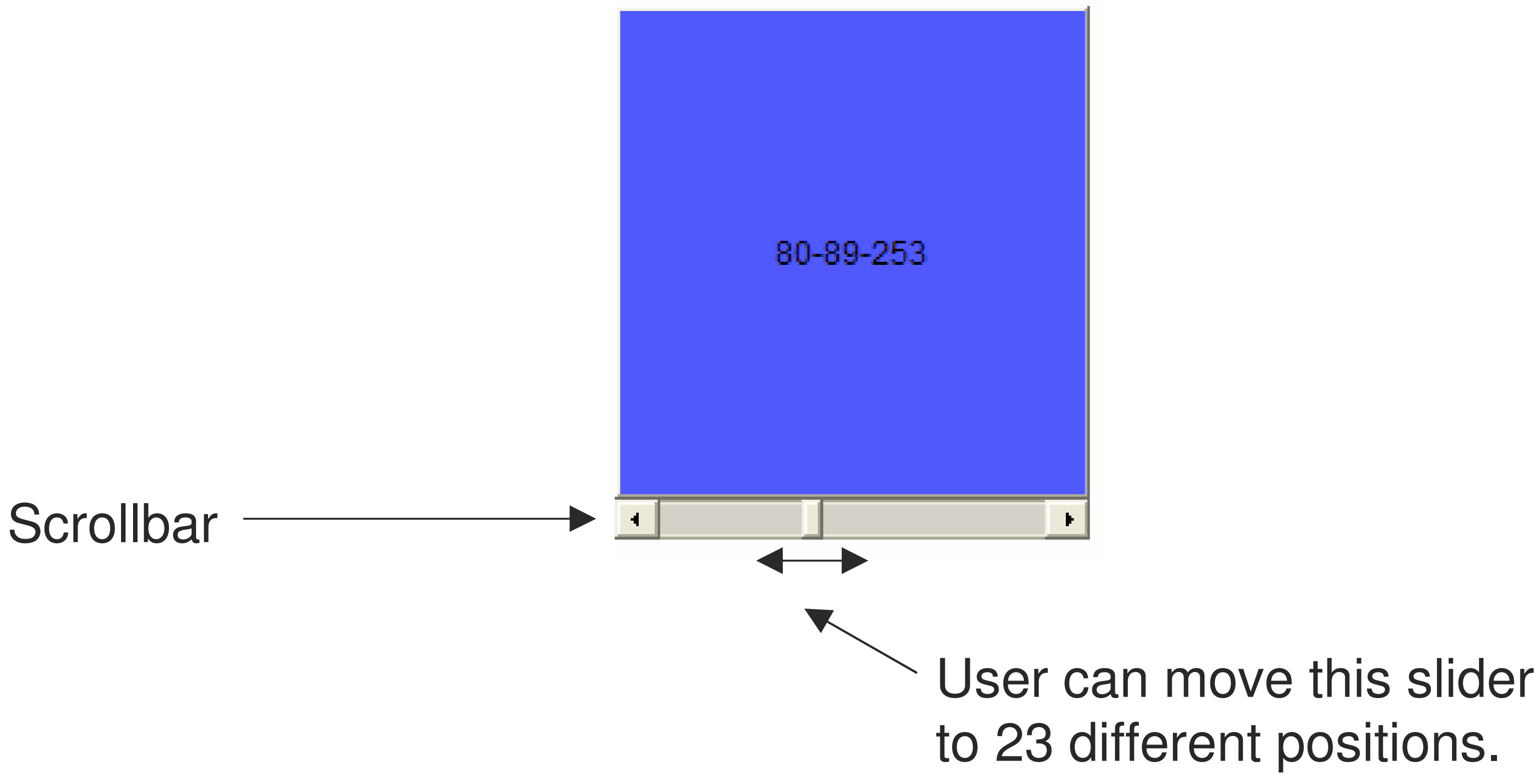
Child (old): 00 | 111111

→

Child (new): 00000 | 111

Let's move the crossover point to the right for 3 positions.

Interface (User-Modification)

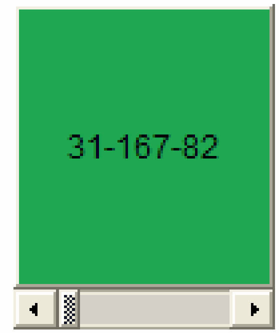


Human-Based Crossover example

Parent 1: 0 0000111 00100111 00110000

Parent 2: 0 0011111 10100111 01010010

Child: 0 0011111 10100111 01010010 (31-167-82)

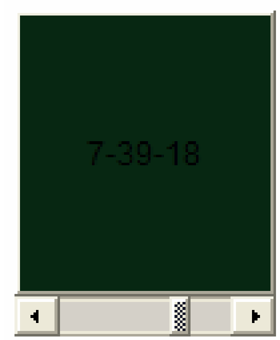


original position

Parent 1: 00000111 00100111 00 110000

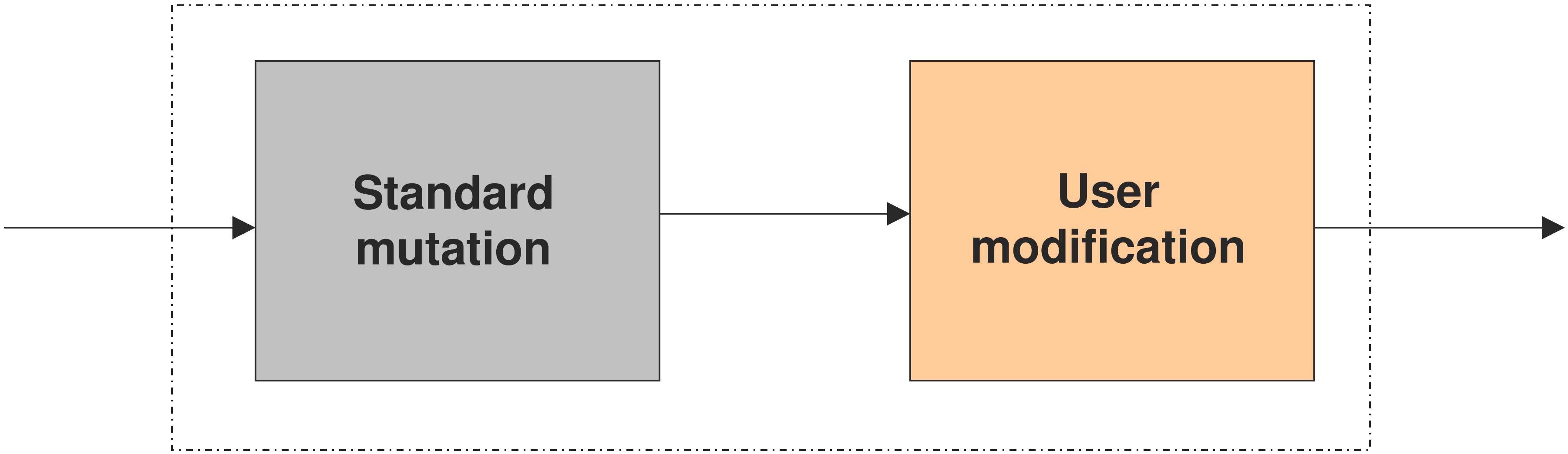
Parent 2: 00011111 10100111 01 010010

New Child: 00000111 00100111 01 010010 (7-39-18)



new position

Human-Based Mutation



Human-Based Mutation

Standard Mutation	Parent:	00000000	XOR
	Mutation Mask:	00<u>1</u>00000	
	Mutant:	00100000	

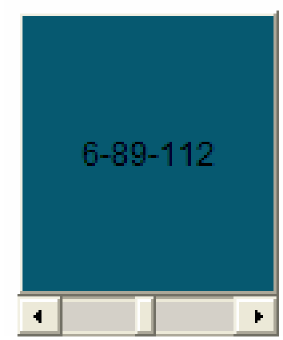
Human-Based Mutation	Mutation Mask: (old)	00<u>1</u>00000	User moves the slider (locus of mutation) to the right for 3 positions.
		→	
	Mutation Mask: (new)	00000<u>1</u>00	
	Mutant (new):	00000100	

Human-Based Mutation example

Parent:	00000110	11011000	01111001	XOR
Mutation Mask:	01000000	10000100	10000000	
Mutant:	01000110	01011100	11111001	(70-92-249)

original position

Parent:	00000110	11011000	01111001	XOR
New Mutation Mask:	00000000	10000001	00001001	
New Mutant:	00000110	01011001	01110000	(6-89-112)



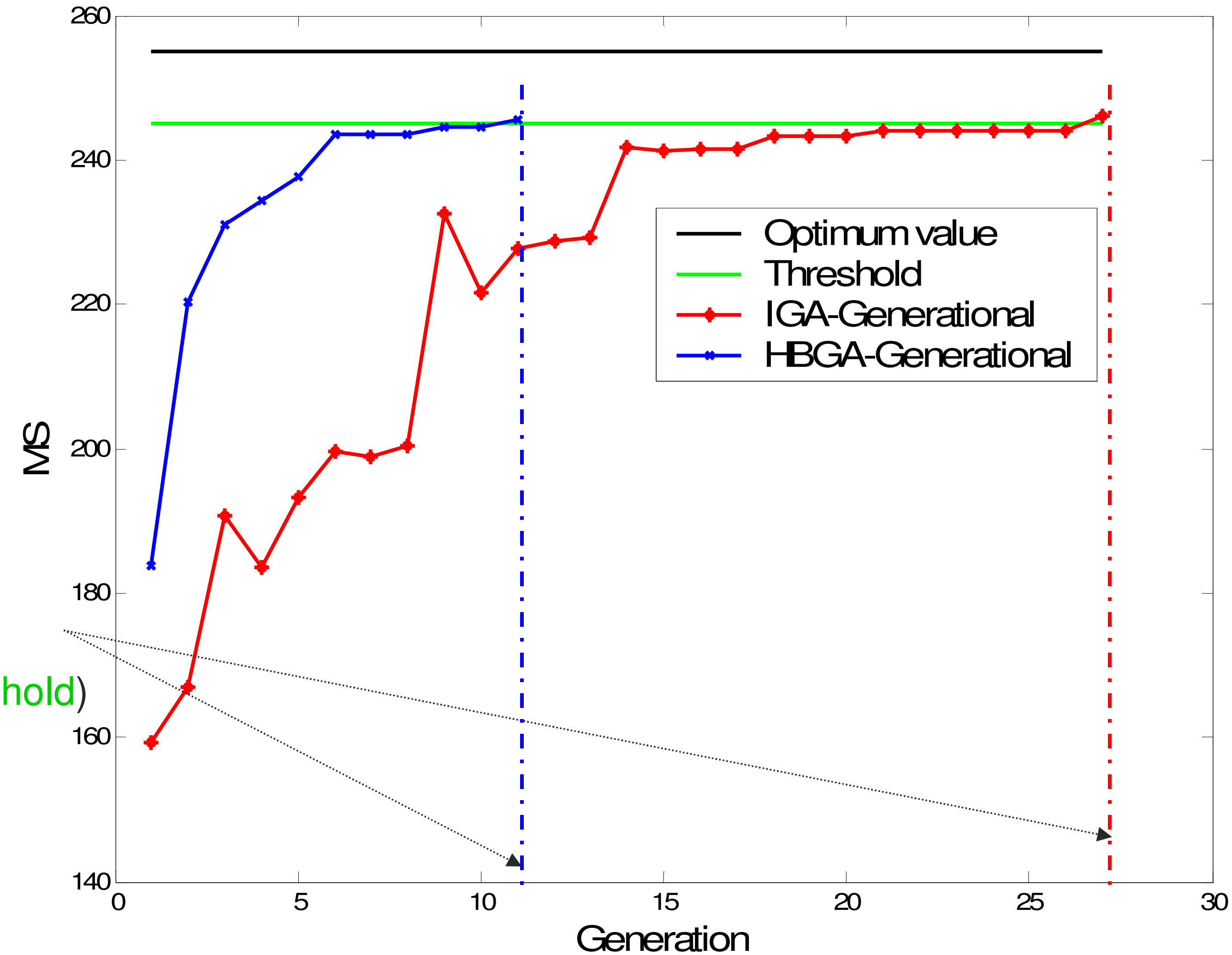
new position

Experimental Results

- Generational
  - MS (Minimum Component Metric)
  - M1 (Brightness)
  - M2 (Euclidean Distance)
- Steady State
  - MS (Minimum Component Metric)
  - M1 (Brightness)
  - M2 (Euclidean Distance)

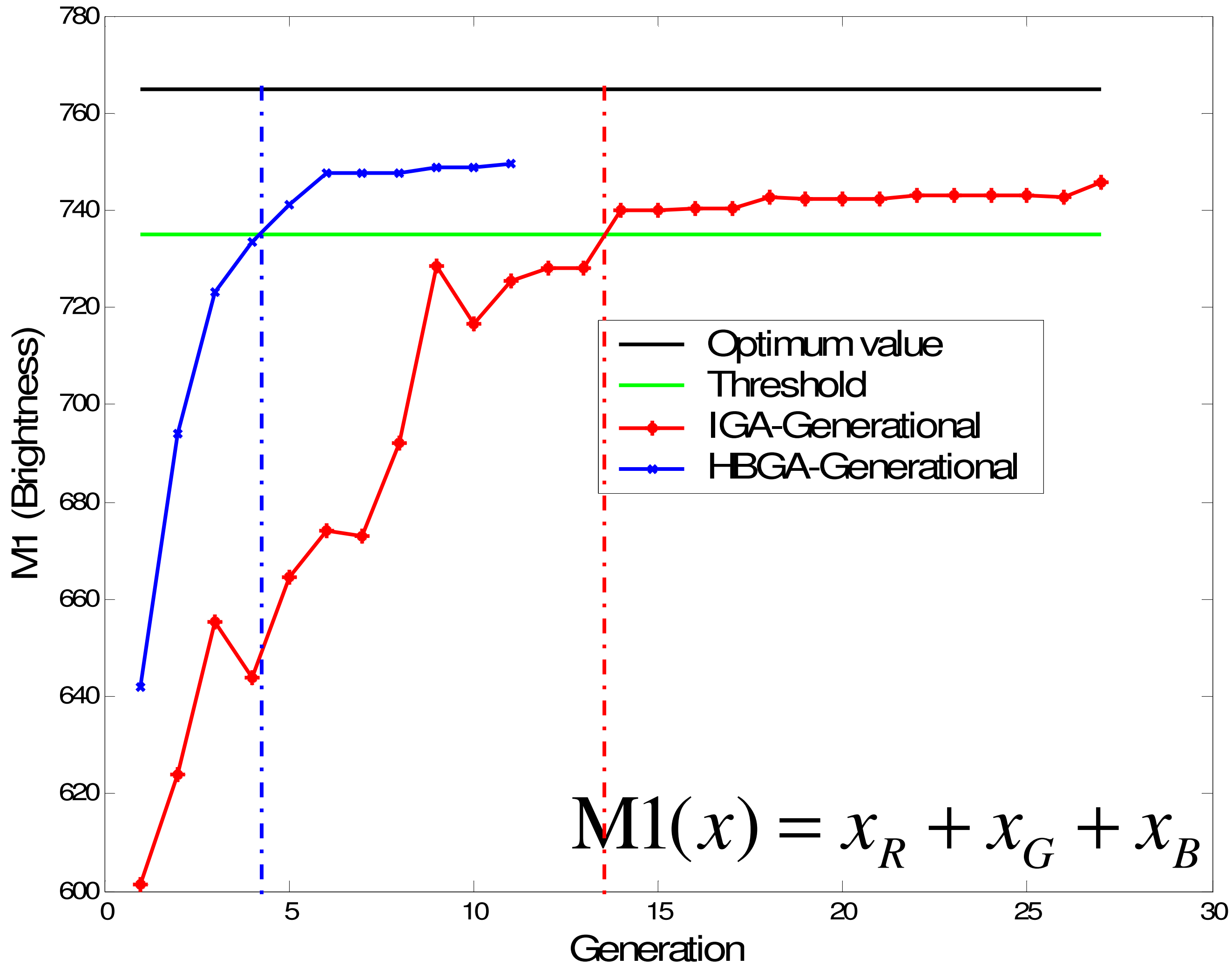
Minimum Component Metric (MS) of the brightest individual in the population for **Generational** type of IGA and HBGA

HBGA takes **fewer generations** to converge (crossing threshold) than IGA.

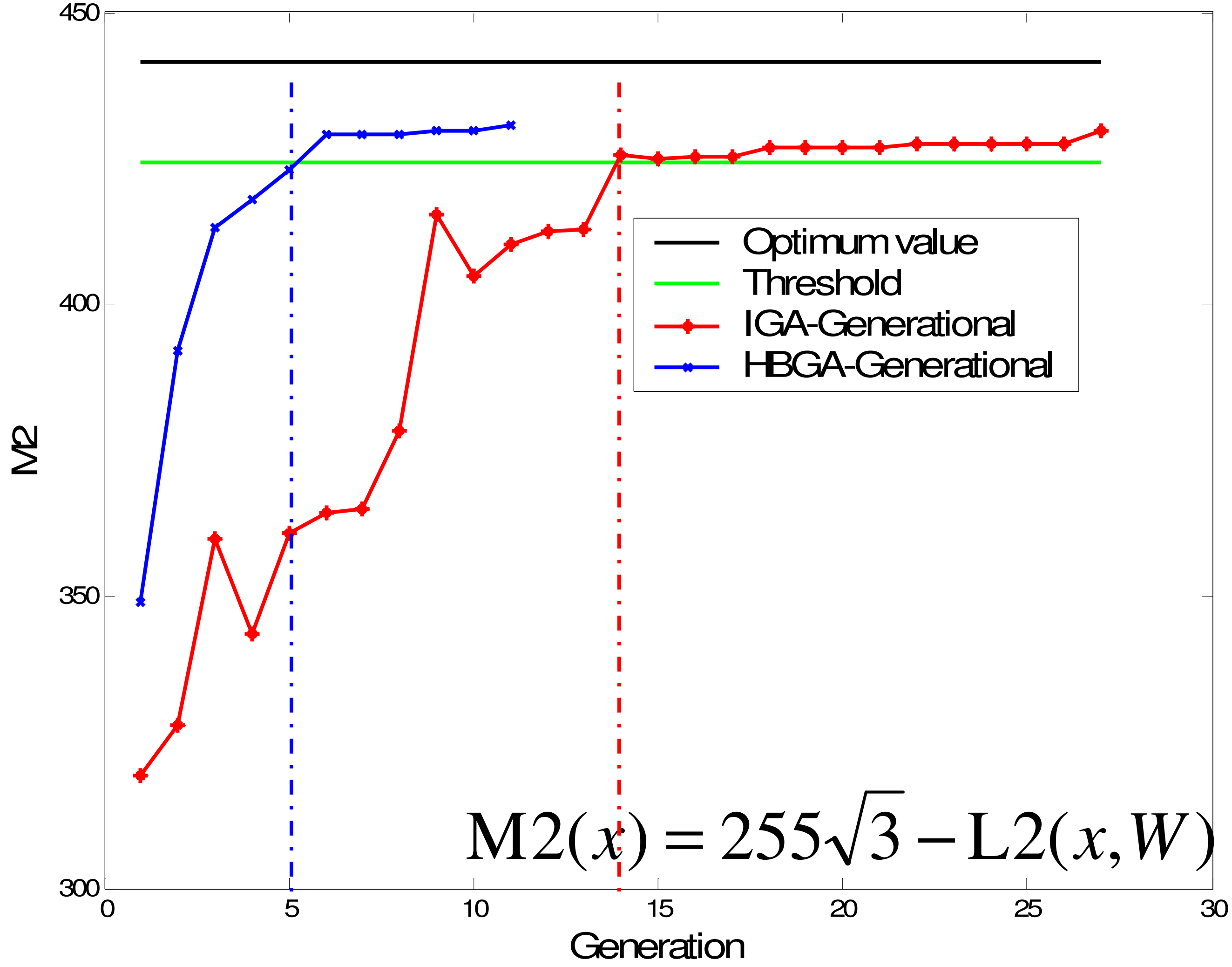


$$MS(x) = \min(x_R, x_G, x_B)$$

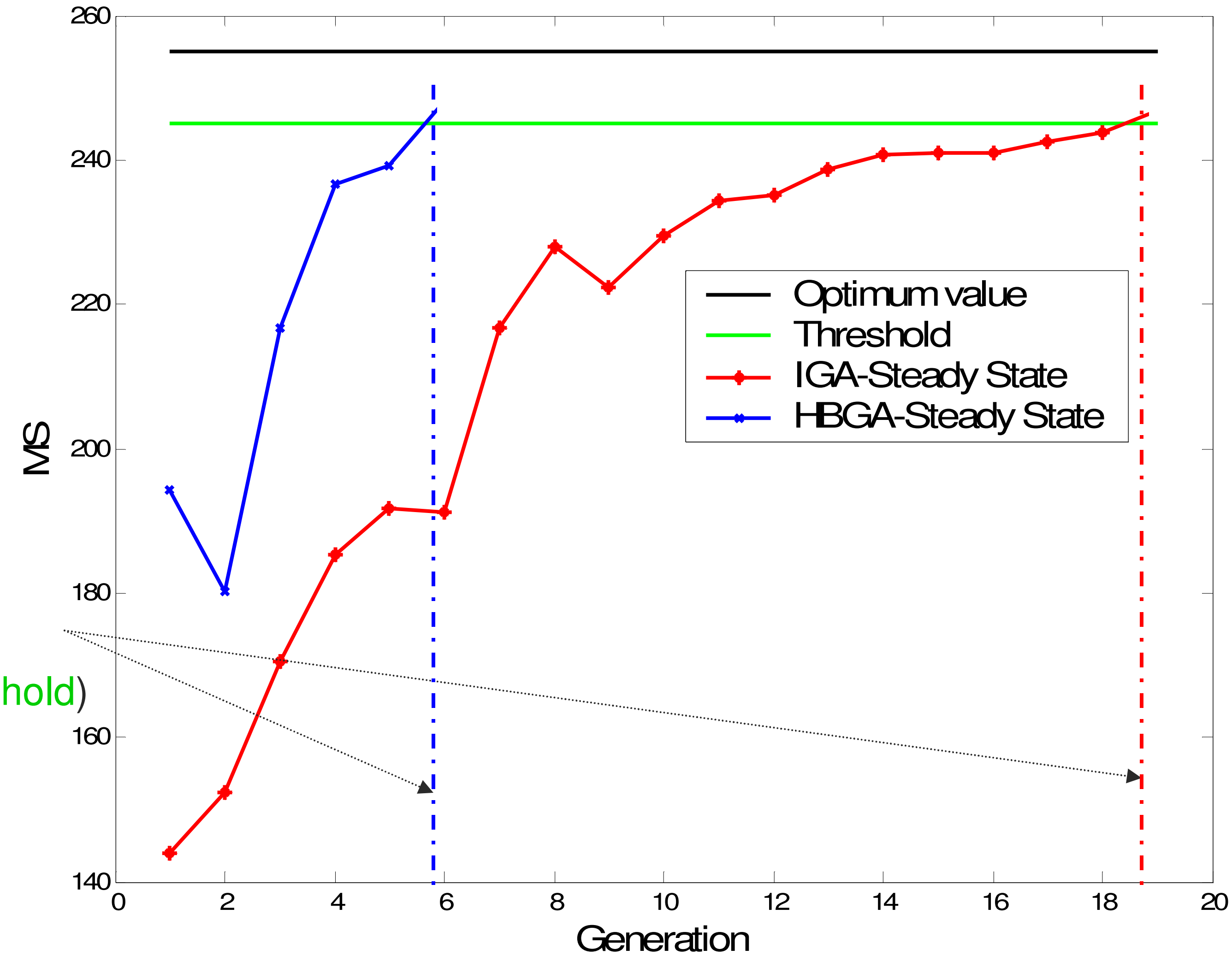
Brightness (M1) of the brightest individual in the population for **Generational** type of IGA and HBGA



Euclidean metric (M2) of the brightest individual in the population for **Generational** type of IGA and HBGA



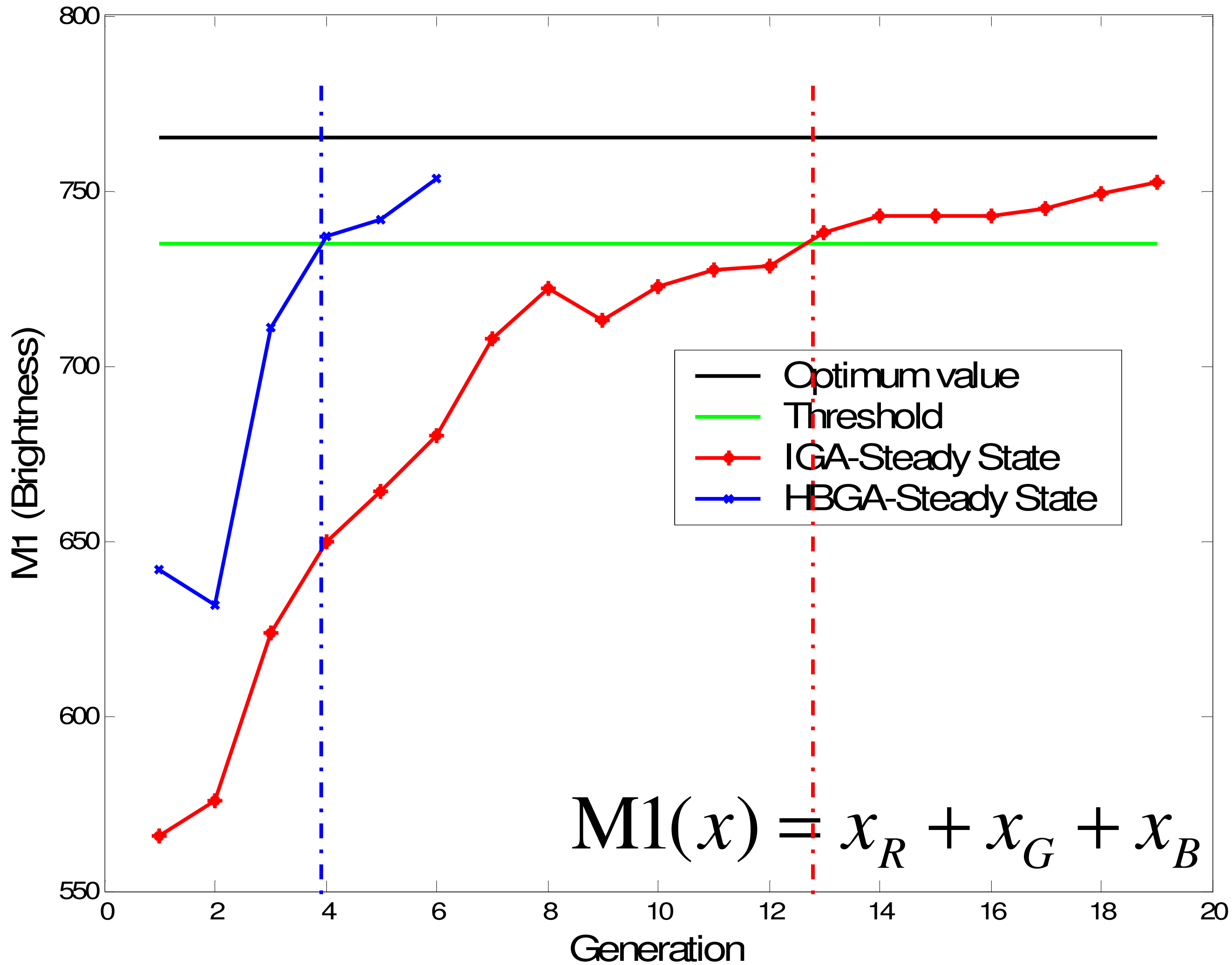
Minimum Component Metric (MS) of the brightest individual in the population for **Steady State** type of IGA and HBGA



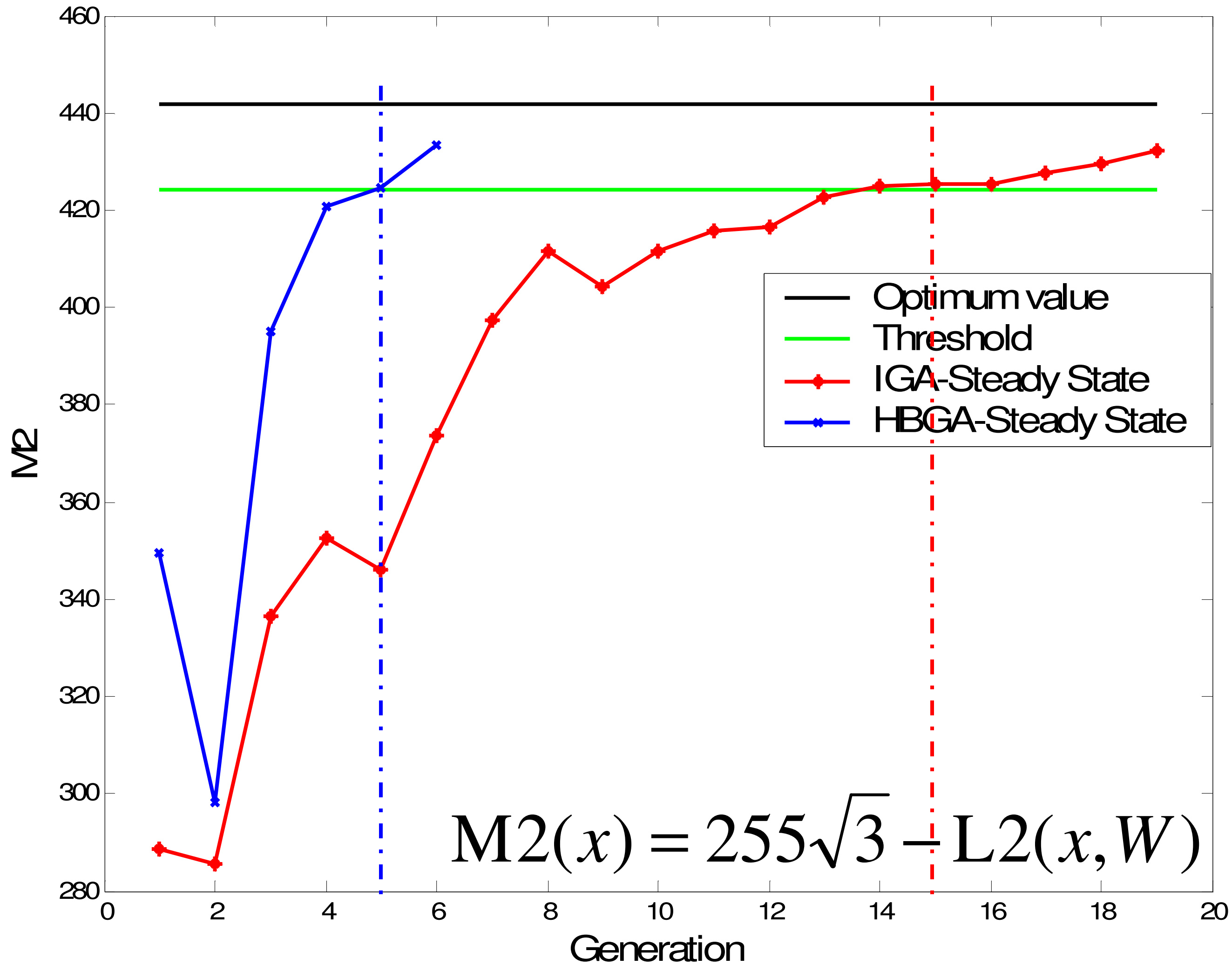
HBGA
takes fewer
generations
to converge
(crossing threshold)
than IGA.

$$MS(x) = \min(x_R, x_G, x_B)$$

Brightness (M1) of the brightest individual in the population for **Steady State** type of IGA and HBGA



Euclidean metric (M2) of the brightest individual in the population for **Steady State** type of IGA and HBGA



Performance comparison (Number of generation to converge)

	Generational			Steady State		
Progress Metric	HBGA	IGA	HBGA Speedup $\left(\frac{\text{IGA}}{\text{HBGA}}\right)$	HBGA	IGA	HBGA Speedup $\left(\frac{\text{IGA}}{\text{HBGA}}\right)$
M1(Brightness)	4	13	3.25	4	13	3.25
M2	6	14	2.33	5	14	2.8
MS	11	27	2.45	6	19	3.17

- The algorithms using human-based innovation operators require **2-3 times less generations** to converge.

Performance comparison (Time)

	Generational			Steady State		
	HBGA	IGA	HBGA Speedup $\left(\frac{\text{HBGA}}{\text{IGA}}\right)$	HBGA	IGA	HBGA Speedup $\left(\frac{\text{HBGA}}{\text{IGA}}\right)$
Generation time (s)	9.2	5.5	-	14.2	6.5	-
Number of Generations	11	27	-	6	19	-
Time to converge (s)	103.8	147.6	1.42	92.3	125.7	1.36

- The algorithms using human-based innovation operators show a **time-to-converge speedup of 36-42%.**

Conclusions

- HBGA requires **2-3 times less generation** than IGA to achieve the same goal.
- HBGA shows a **time-to-converge speedup of 36-42%**.
- **Using human-based innovation operators is advantageous** even when computational innovation operators are available.

Questions?