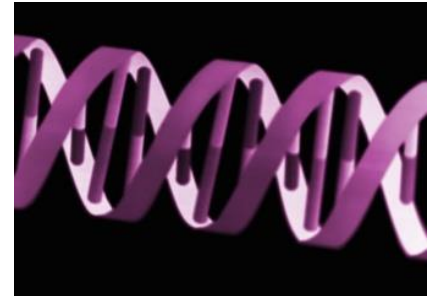


Pattern-based preservation of building blocks in genetic algorithms

Yoshitaka Kameya & Chativit Prayoonsri
(Tokyo Institute of Technology)

Background

- Genetic Algorithms
 - Building Block Hypothesis:
The building blocks are created and mixed effectively in the evolutionary process
 - But actually, building blocks are often destroyed by genetic operations soon after they are created
 - How to avoid this destruction?

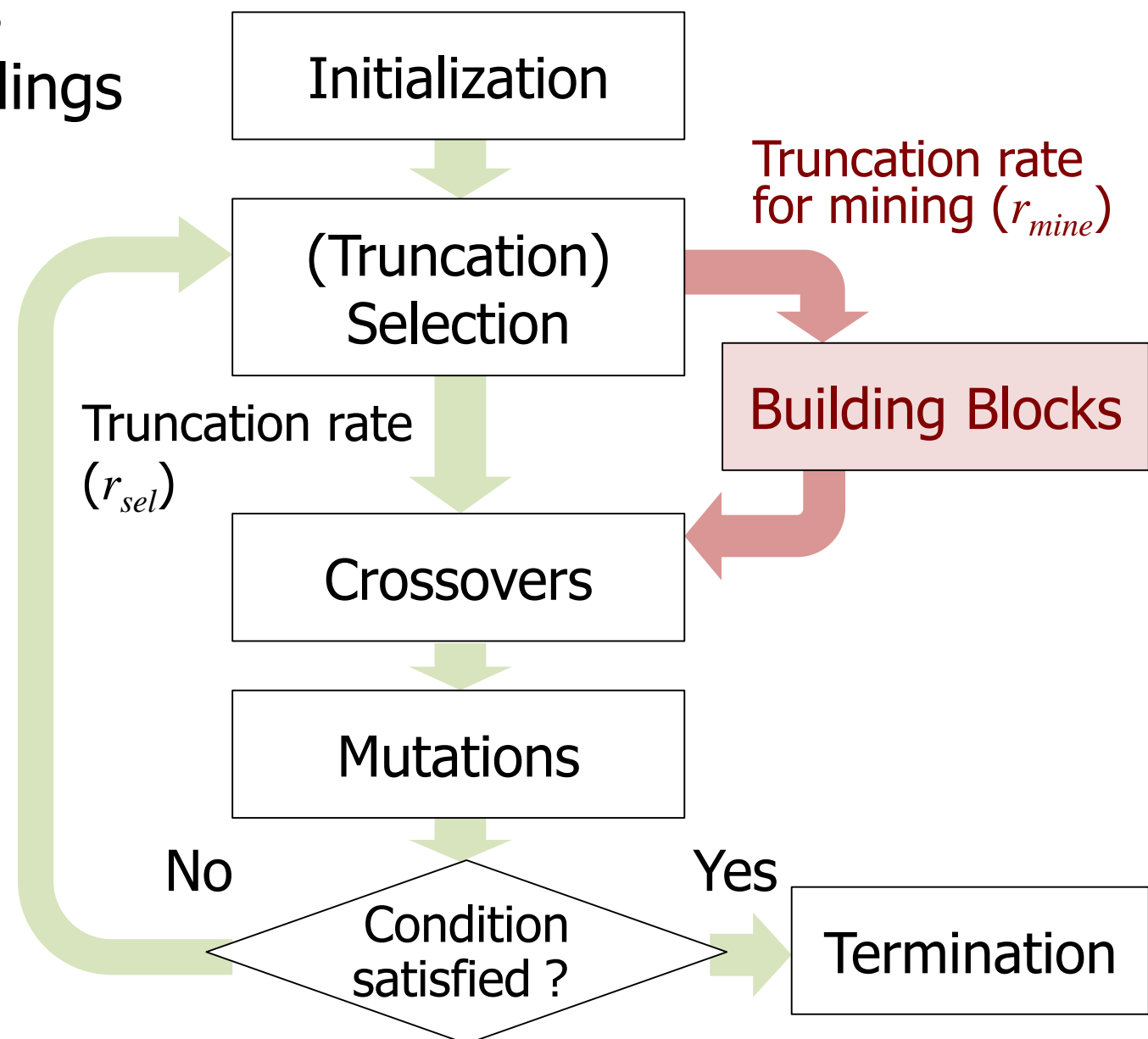


Background (cont'd)

- Linkage Learning
 - Linkage adaptation: messy GA, LLGA, ...
 - Perturbation-based methods: LINC, LIMD, D5, ILI, ...
 - Model building methods: eCGA, BOA, DSMGA, ...
- Disadvantages in previous LL:
 - BBs are only protected in locus-wise (coarse-grained)
 - Only a few methods can handle permutations or variable-length chromosomes
 - Repair operator
 - Sampling under the constraint for permutation

Our proposal

- GAP (genetic algorithms with patterns)
 - ... Inherits the notion of “genetic engineering,” originally proposed by Gero & Kazakov (1995, 2001)
 - ... Uses frequent pattern mining techniques
 - ... Can deal with permutations as well as fixed-position encodings
 - ... Protects BBs in allele-wise (fine-grained)
 - ... Requires no extra fitness evaluations
 - ... Is understandable from a biological point of view
 - ... Makes possible a post-analysis of the evolution process



Chromosome encodings and patterns

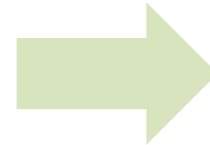
- Highly-fit individuals = Database of sequences
- Translation:
 - Fixed-position encodings: a sequence of locus-allele pairs
 - Chromosome $\langle 1, 0, 1, 1, 0 \rangle$
→ Sequence $\langle (1, 1), (2, 0), (3, 1), (4, 1), (5, 0) \rangle$
 - Permutations: as-is
 - Chromosome $\langle a, c, b \rangle$ → Sequence $\langle a, c, b \rangle$
- Patterns = Subsequence with gaps:
 - One subsequence of $\langle (1, 1), (2, 0), (3, 1), (4, 1), (5, 0) \rangle$ is $\langle (2, 0), (5, 0) \rangle$
 - Subsequences of $\langle a, c, b \rangle$ are $\langle a \rangle, \langle b \rangle, \langle c \rangle, \langle a, b \rangle, \langle a, c \rangle, \langle c, b \rangle, \langle a, c, b \rangle$

Mining BBs

- We find patterns that occur no less than σ_{\min} (minimum support) in highly-fit individuals
- Fixed-position encodings:

Minimum support $\sigma_{\min} = 4$

<11101>
<11001>
<11100>
<01011>
<11101>
<01001>



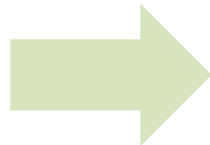
of occurrences

6	#1###	4	11#0#
5	#1#0#	4	#1#01
5	#1##1	4	11###
5	###0#	4	1##0#
5	####1	4	###01
		4	1####

- Permutation encodings:

$\sigma_{\min} = 4$

<a b c d e f>
<d a b e c f>
<a c b e f d>
<d c e b a f>
<a c b d e f>

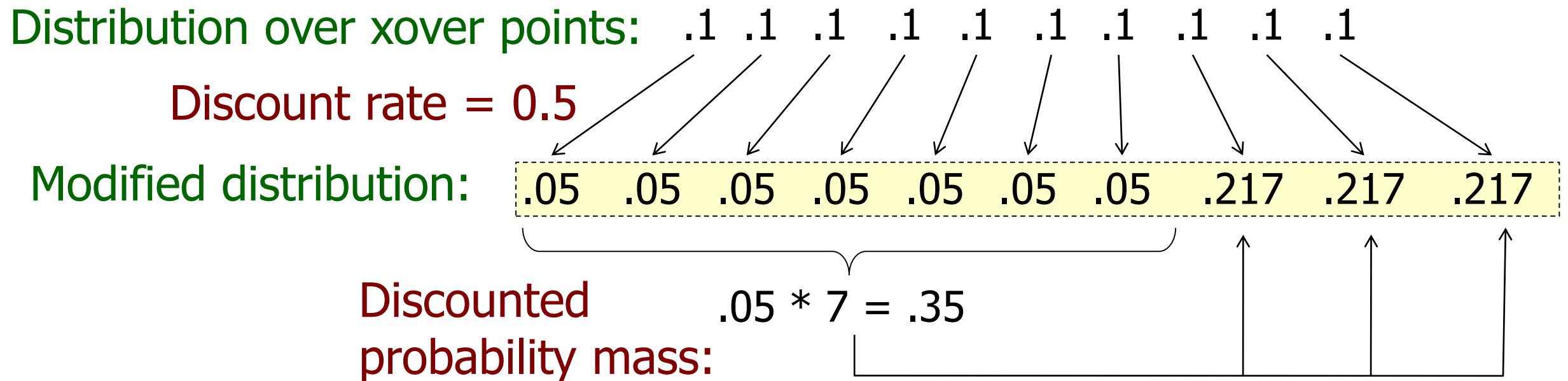
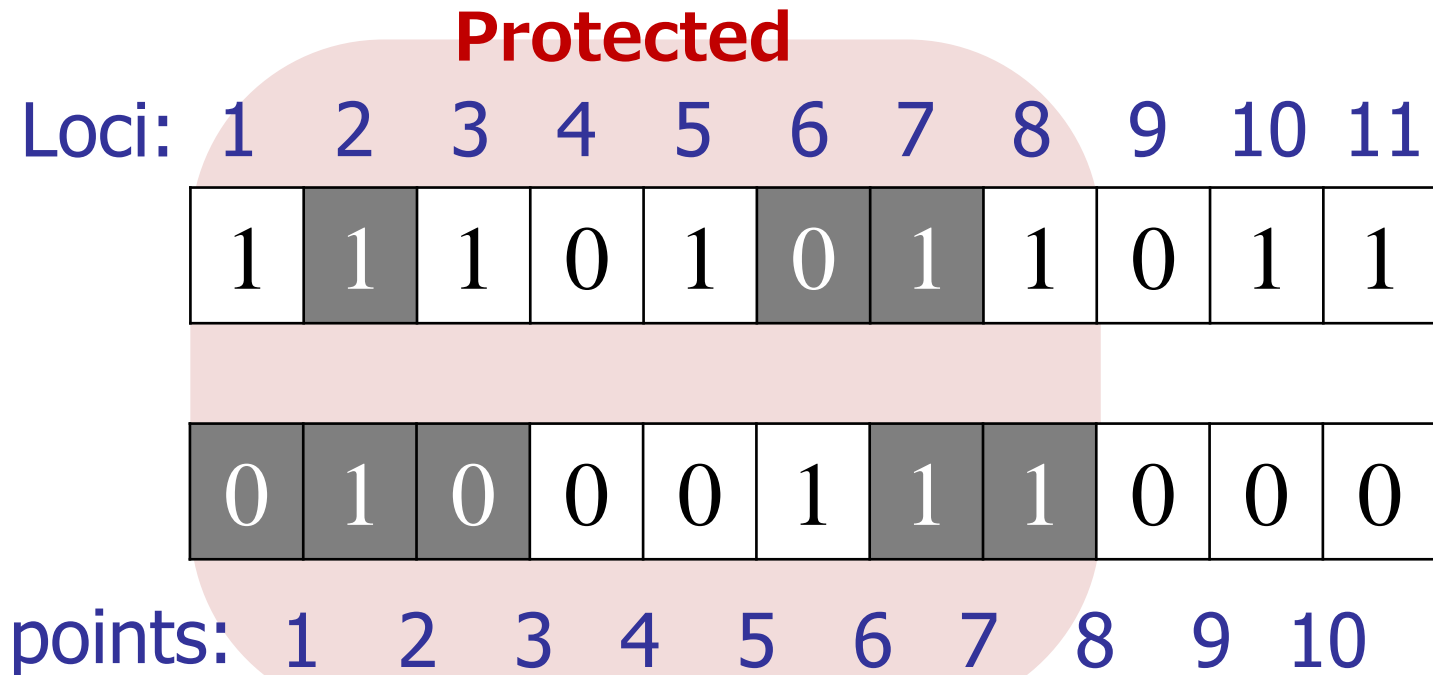


5	<a f>	4	<a b e f>	4	<a b>
5	<b f>	4	<a b e>	4	<a c>
5	<c f>	4	<a b f>	4	<a e>
5	<e f>	4	<a c f>	4	<b e>
5	<a>	4	<a e f>	4	<c e>
5		4	<b e f>	4	<d e>
5	<c>	4	<c e f>	4	<d f>
5	<d>	4	<d e f>		
5	<e>				
5	<f>				

We further use advanced mining techniques:
Top-K & closed pattern mining

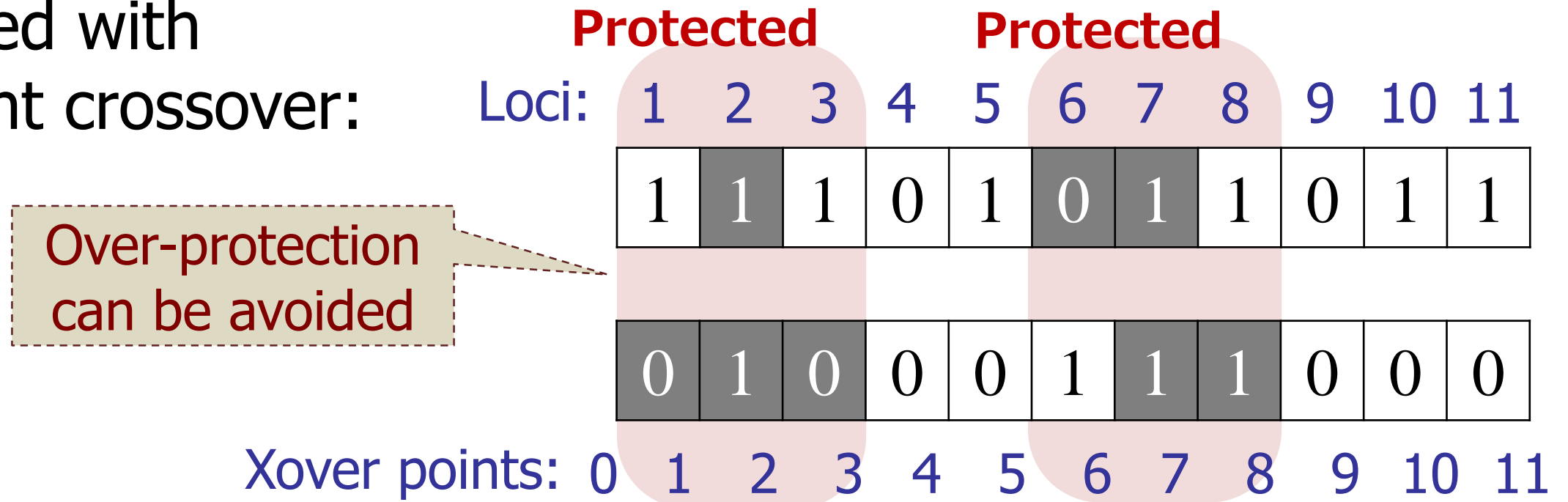
Protection of BBs

- Combined with single-point crossover:



Protection of BBs (cont'd)

- Combined with two-point crossover:



- Combined with edge recombination:

Parents :

1	2	3	4	5	6
---	---	---	---	---	---

2	4	3	1	5	6
---	---	---	---	---	---



Offspring :

2	3	4	5	1	6
---	---	---	---	---	---

City γ	Edge list $M[\gamma]$
1	{2, 3, 5, 6}
2	{1, 3, 4, 6}
3	{1, 2, 4}
4	{2, 3, 5}
5	{1, 4, 6}
6	{1, 2, 5}

γ	Frequent edge list $M^*[\gamma]$
1	\emptyset
2	{4}
3	\emptyset
4	{2,5}
5	{4}
6	\emptyset

Experiments: Traveling Salesman Problem

- Grötschels120 (from TSBLIB)
 - Averaged fitness of the best individuals (100 runs, Optimum = 6,942)

G120	SGA				GAP			
Pop	ER		PX		ER		PX	
200	9434.5	± 34.8	9742.3	± 69.1	9542.3	± 38.5	9730.2	± 85.4
500	8411.0	± 29.1	8075.4	± 34.2	8432.3	± 23.1	8071.6	± 33.6
1000	8258.1	± 28.9	7733.1	± 29.2	8254.1	± 27.9	7694.2	± 23.7
2000	8148.2	± 34.8	7579.0	± 21.1	8103.7	± 21.6	7570.5	± 21.9
5000	8012.6	± 47.7	7489.2	± 18.2	7805.3	± 17.4	7477.2	± 65.3

- Averaged # of generations (100 runs)

G120	SGA				GAP			
Pop	ER		PX		ER		PX	
200	557.0	± 8.7	488.2	± 10.9	544.3	± 8.9	513.5	± 21.2
500	874.7	± 20.1	528.2	± 17.8	856.9	± 19.8	490.5	± 9.2
1000	1039.9	± 20.8	574.2	± 31.1	928.9	± 20.4	553.6	± 22.9
2000	1275.0	± 29.4	649.7	± 40.0	1044.3	± 18.8	580.5	± 29.4
5000	1561.8	± 32.4	827.0	± 57.4	1183.0	± 22.6	856.0	± 61.0

Future work

- Implementational improvement of the mining algorithm

Execution time in the current implementation:

Royal Road (tight, 2PTX)

Pop	Truncation rate for mining (best)	Exec. time (msec / generation)			
		GA	Mining	Protection	Eval
64	0.20	0.38	3.51	1.18	0.03
128	0.05	1.09	1.77	1.82	0.07
256	0.20	2.31	90.63	6.74	0.17
512	0.10	4.76	137.26	13.06	0.33

Base algorithm : LCM [Uno et al. 2004]

Grötschels120 (Edge recomb.)

Pop	Truncation rate for mining (best)	Exec. time (msec / generation)			
		GA	Mining	Protection	Eval
200	0.05	9.78	12.25	1.83	0.15
500	0.01	24.43	4.92	1.98	0.32
1000	0.10	48.42	82.75	23.44	0.60
2000	0.10	98.02	164.39	51.56	1.20
5000	0.01	222.01	30.83	122.34	3.02

Base algorithm : BIDE [Wang et al. 2004]

- Emerging patterns [Dong et al. 1999]
 - Occur frequently in good individuals but infrequently in bad individuals
- Management of a BB library through generations