

Dimensionality Reduction: Genetic Algorithm

Pilsung Kang

School of Industrial Management Engineering

Korea University

Genetic Algorithm

Siedlecki, W. Sklansky (1989)

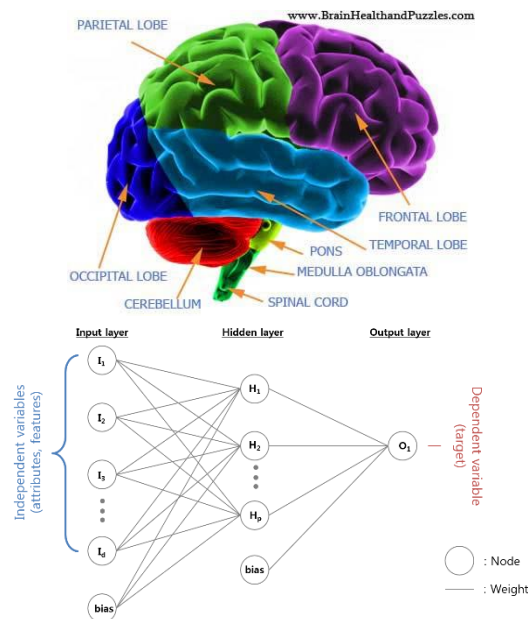
- Limitations of the previous variable selection methods
 - ✓ Exhaustive search: guarantee the optimal subset, but takes too long time (practically impossible for many tasks)
 - ✓ Local search (forward/backward/stepwise): efficient search but the search space is very limited, which leads to a low probability of finding the optimal solution
- Idea
 - ✓ Improve the performance of local searches with a little additional computational time!

Genetic Algorithm

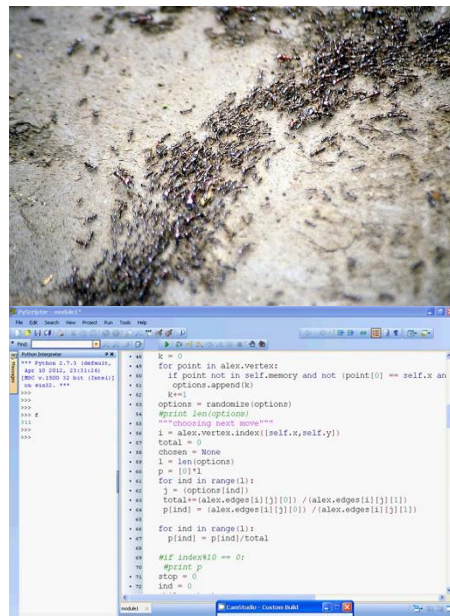
- Meta-Heuristic Approach

- ✓ Solve a complex problem by doing trials and errors **efficiently**
- ✓ Among the optimization algorithms, many of them mimic the way of a natural system works

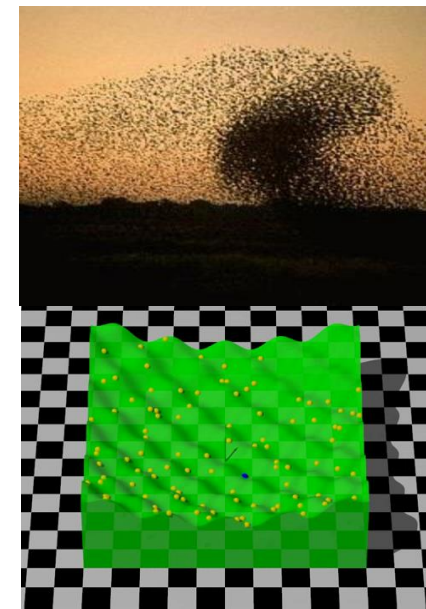
Artificial Neural Networks



Ant Colony Algorithm



Particle Swarm Optimization



Genetic Algorithm

- Meta-Heuristic Approach

AlphaStar: An Evolutionary Computation Perspective

Kai Arulkumaran
Imperial College London
London, United Kingdom
ka709@ic.ac.uk

Antoine Cully
Imperial College London
London, United Kingdom
a.cully@imperial.ac.uk

Julian Togelius
New York University
New York City, NY, United States
julian@togelius.com

ABSTRACT

In January 2019, DeepMind revealed AlphaStar to the world—the first artificial intelligence (AI) system to beat a professional player at the game of StarCraft II—representing a milestone in the progress of AI. AlphaStar draws on many areas of AI research, including deep learning, reinforcement learning, game theory, and evolutionary computation (EC). In this paper we analyze AlphaStar primarily through the lens of EC, presenting a new look at the system and relating it to many concepts in the field. We highlight some of its most interesting aspects—the use of Lamarckian evolution, competitive co-evolution, and quality diversity. In doing so, we hope to provide a bridge between the wider EC community and one of the most significant AI systems developed in recent times.

CCS CONCEPTS

• **Computing methodologies** → **Multi-agent reinforcement learning**; **Neural networks**; *Bio-inspired approaches*;

KEYWORDS

Lamarckian evolution, co-evolution, quality diversity

beat a grandmaster at StarCraft (SC), a real-time strategy game. Both the original game, and its sequel SC II, have several properties that make it considerably more challenging than even Go: real-time play, partial observability, no single dominant strategy, complex rules that make it hard to build a fast forward model, and a particularly large and varied action space.

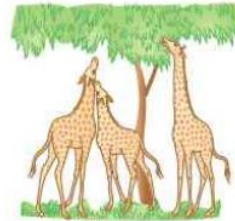
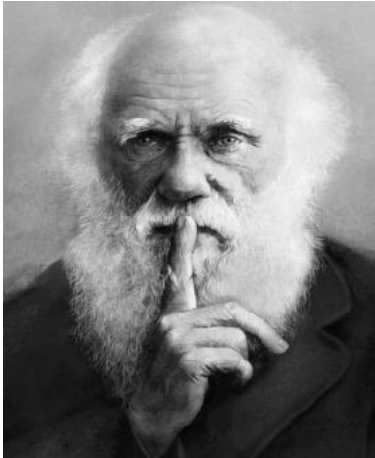
DeepMind recently took a considerable step towards this grand challenge with AlphaStar, a neural-network-based AI system that was able to beat a professional SC II player in December 2018 [20]. This system, like its predecessor AlphaGo, was initially trained using imitation learning to mimic human play, and then improved through a combination of reinforcement learning (RL) and self-play. At this point the algorithms diverge, as AlphaStar utilises population-based training (PBT) [9] to explicitly keep a population of agents that train against each other [8]. This part of the training process was built upon multi-agent RL and game-theoretic perspectives [2, 10], but the very notion of a population is central to evolutionary computation (EC), and hence we can examine AlphaStar through this lens as well¹.

2 COMPONENTS

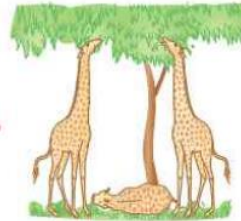
v3 [cs.NE] 14 Jul 2019

Genetic Algorithm

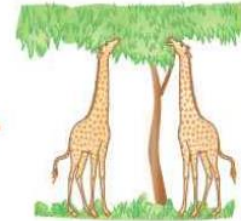
- 자연선택설 vs. 용불용설



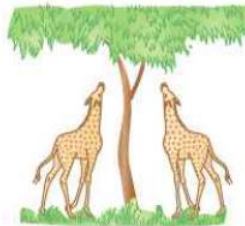
원래 기린은 목의 길이가 다양했다.



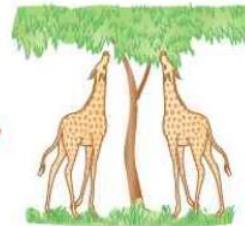
목이 긴 기린이 생존에 유리하여 살아 남았다.



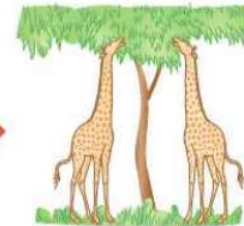
목이 긴 기린이 자손을 남기는 과정이 반복되어 기린의 목이 길어졌다.



원래 기린은 목이 짧았다.



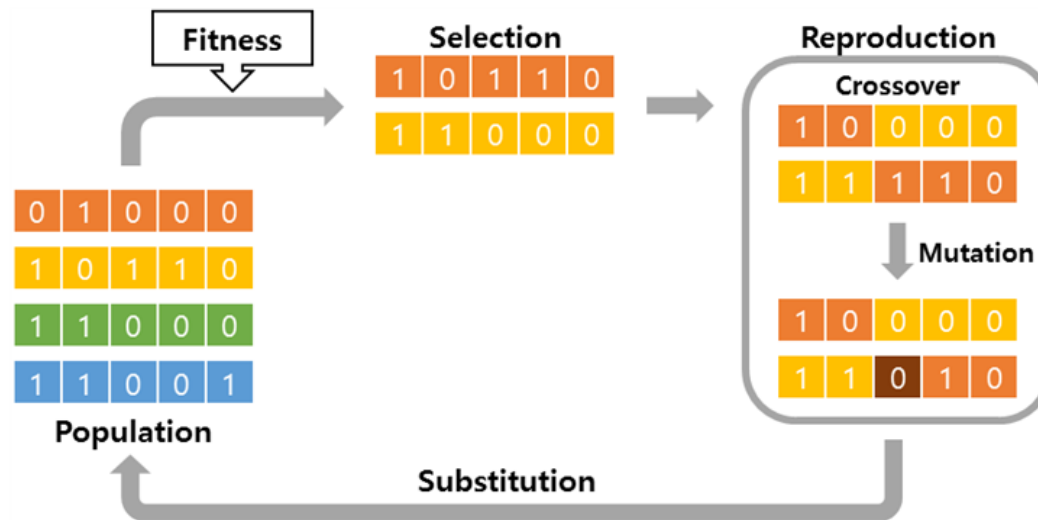
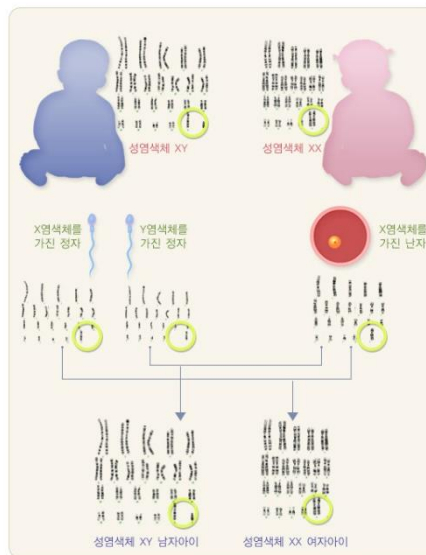
높은 곳의 잎을 먹기 위해 목을 자주 길게 뻗어 목이 점차 길어졌다.



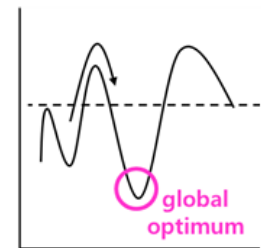
오늘날 기린은 긴 목을 가지게 되었다.

Genetic Algorithm

- An Evolutionary Algorithm that mimics the Reproduction of Creatures
 - ✓ Find a superior solutions and preserve by repeating the reproduction process
 - Selection: Select a superior solution to improve the quality
 - Crossover: Search various alternatives based on the current solutions
 - Mutation: Give a chance to escape the local optima



(b)



(c)

Genetic Algorithm

- Genetic Algorithm for Feature Selection

Step1:
염색체 초기화 및 파라미터 설정 (Initiation)

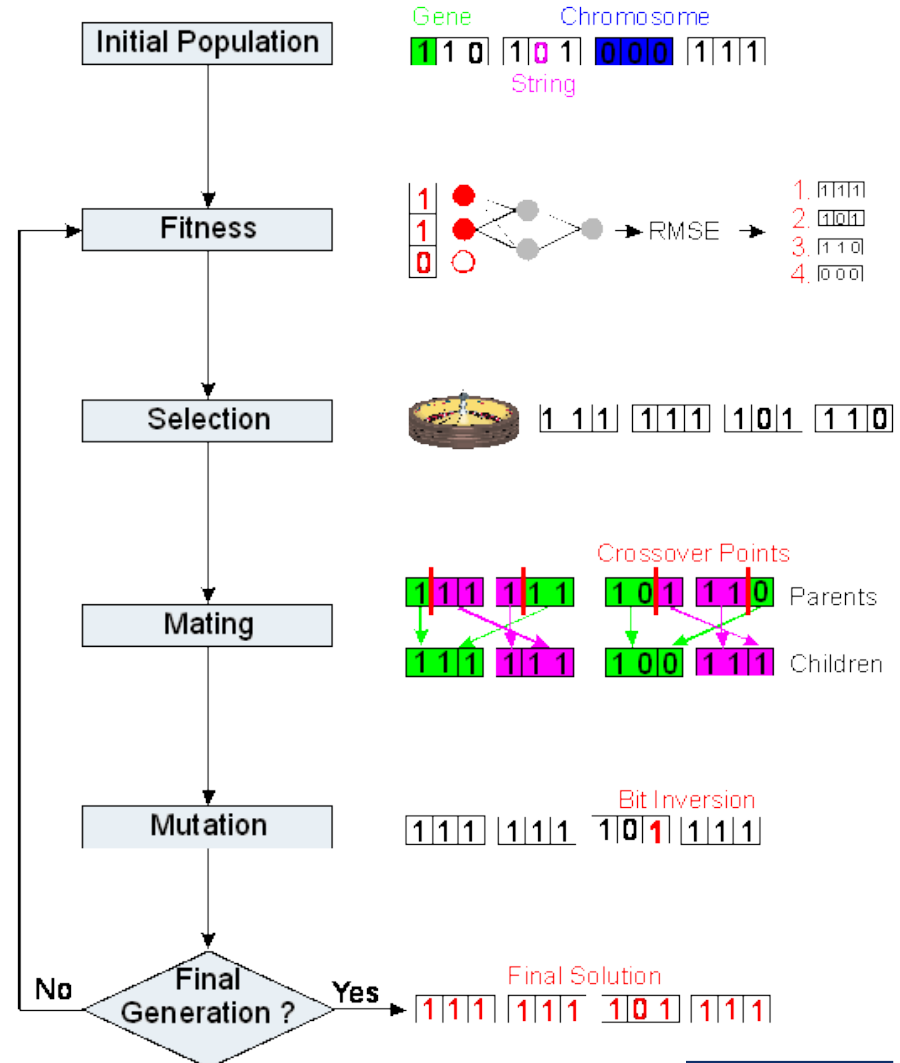
Step2:
각 염색체 선택 변수별 모델 학습

Step3:
각 염색체 적합도 평가 (Fitness evaluation)

Step4:
우수 염색체 선택 (Selection)

Step5:
다음 세대 염색체 생성 (Crossover & Mutation)

Step6:
최종 변수 집합 선택



GA Step I: Initialization

- Encoding Chromosomes

- ✓ Genetic algorithm can be used not only for variable selection, but for a wide range of optimization problems
- ✓ Encoding scheme can be different for different tasks
- ✓ Binary encoding is commonly used for variable selection

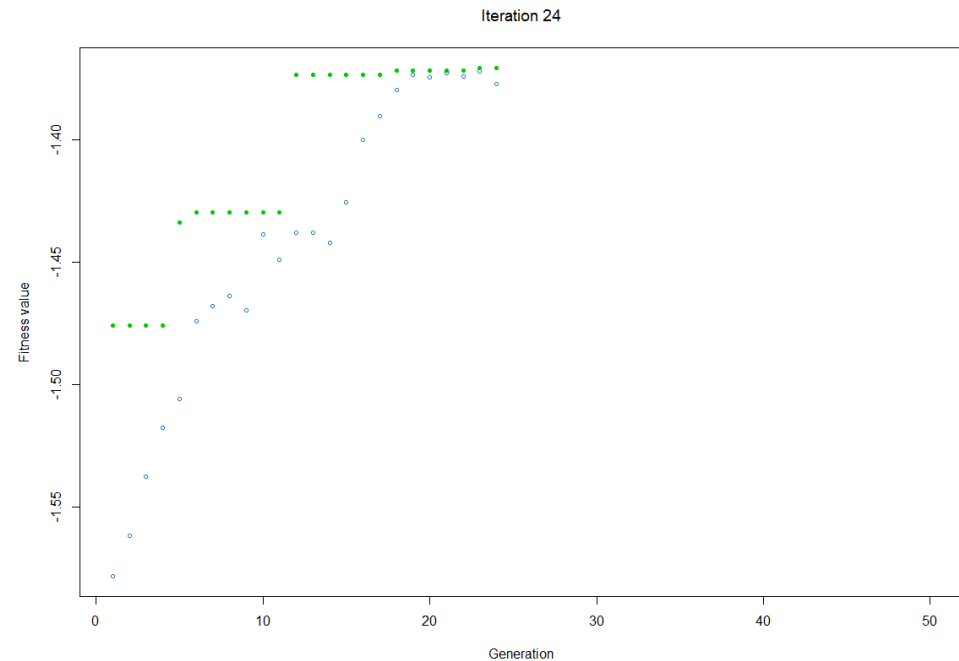
Chromosome				Gene					
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	...	x_d
1	0	0	1	0	1	1	0	...	1

1: Use the corresponding variable in the modeling

0: Do not use the variable

GA Step I: Initialization

- Parameter Initialization
 - ✓ The number of chromosome (population)
 - ✓ Fitness function
 - ✓ Crossover mechanism
 - ✓ The rate of mutation
 - ✓ Stopping criteria
 - minimum fitness improvement
 - maximum iterations, etc.



GA Step I: Initialization

- Example: Population Initialization

- ✓ Random number generation for each gene
- ✓ Convert the random numbers to binary values (cut-off = 0.5 in this example)

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 1	0.16	0.74	0.90	0.96	0.19	0.70	0.31	0.23	0.83	0.62
Chromosome 2	0.71	0.75	0.82	0.83	0.41	0.97	0.14	0.40	0.89	0.66
Chromosome 3	0.36	0.56	0.40	0.08	0.80	0.32	0.44	0.27	0.34	0.31
Chromosome 4	0.30	0.26	0.83	0.71	0.70	0.74	0.11	0.57	0.81	0.01
Chromosome 5	0.27	0.29	0.67	0.54	0.38	0.09	0.07	0.90	0.00	0.53
Chromosome 6	0.70	0.59	0.62	0.18	0.36	0.02	0.45	0.24	0.40	0.06
Chromosome 7	0.43	0.49	0.34	0.36	0.26	0.67	0.09	0.68	0.29	0.44
Chromosome 8	0.10	0.96	0.20	0.16	0.65	0.07	0.73	0.14	0.91	0.77



	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 1	0	1	1	1	0	1	0	0	1	1
Chromosome 2	1	1	1	1	0	1	0	0	1	1
Chromosome 3	0	1	0	0	1	0	0	0	0	0
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 5	0	0	1	1	0	0	0	1	0	1
Chromosome 6	1	1	1	0	0	0	0	0	0	0
Chromosome 7	0	0	0	0	0	1	0	1	0	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1

GA Step I: Initialization

- Example: Train the model

✓ Assume that the model is a multivariate linear regression (MLR)

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 1	0	1	1	1	0	1	0	0	1	1
Chromosome 2	1	1	1	1	0	1	0	0	1	1
Chromosome 3	0	1	0	0	1	0	0	0	0	0
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 5	0	0	1	1	0	0	0	1	0	1
Chromosome 6	1	1	1	0	0	0	0	0	0	0
Chromosome 7	0	0	0	0	0	1	0	1	0	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1

- For chromosome 1, fit the MLR model

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_2x_2 + \hat{\beta}_3x_3 + \hat{\beta}_4x_4 + \hat{\beta}_6x_6 + \hat{\beta}_9x_9 + \hat{\beta}_{10}x_{10}$$

- For chromosome 2, fit the MLR model

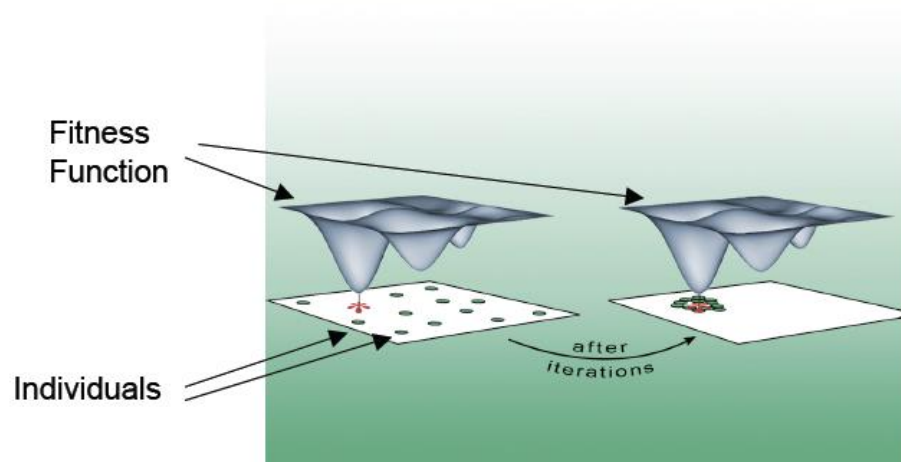
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1x_1 + \hat{\beta}_2x_2 + \hat{\beta}_3x_3 + \hat{\beta}_4x_4 + \hat{\beta}_6x_6 + \hat{\beta}_9x_9 + \hat{\beta}_{10}x_{10}$$

- And so on...

GA Step 3: Fitness Evaluation

- Fitness Function

- ✓ A criterion that determines which chromosomes are better than others
- ✓ In general, the higher the fitness value, the better the chromosomes
- ✓ Common criteria that are embedded in the fitness function
 - If two chromosomes have the same fitness value, the one with fewer variables is preferred
 - If two chromosomes use the same number of variables, the one with higher predictive performance is preferred
- ✓ In case of multiple linear regression
 - Adjusted R^2
 - Akaike information criterion (AIC)
 - Bayesian information criterion (BIC)



GA Step 3: Fitness Evaluation

- Example: Fitness Function

	Adj R^2	Rank	Weight
Chromosome 1	0.75	2	0.177
Chromosome 2	0.78	1	0.184
Chromosome 3	0.50	5	0.118
Chromosome 4	0.65	3	0.154
Chromosome 5	0.40	6	0.095
Chromosome 6	0.35	7	0.083
Chromosome 7	0.25	8	0.059
Chromosome 8	0.55	4	0.130

- ✓ Adjusted R^2 is used for the fitness function
- ✓ Each chromosome's weight is its Adj. R^2 divided by the sum of Adj. R^2 s of all chromosomes
 - Used for probabilistic chromosome selection

GA Step 4: Selection

- Selection

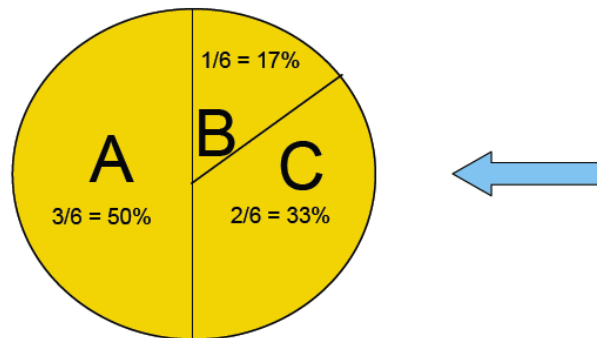
- ✓ Select superior chromosomes in the current population to reproduce the population of the next generation

- ✓ **Deterministic selection**

- Select only top N% of chromosomes
 - Bottom (100-N)% chromosomes are never selected

- ✓ **Probabilistic selection**

- Use the fitness value of each chromosome as the selection weight
 - All chromosomes can be selected with different probabilities



fitness(A) = 3

fitness(B) = 1

fitness(C) = 2

GA Step 4: Selection

- Example: Selection

- ✓ Case 1: Deterministic selection

	Adj R^2	Rank	Weight
Chromosome 1	0.75	2	0.177
Chromosome 2	0.78	1	0.184
Chromosome 3	0.50	5	0.118
Chromosome 4	0.65	3	0.154
Chromosome 5	0.40	6	0.095
Chromosome 6	0.35	7	0.083
Chromosome 7	0.25	8	0.059
Chromosome 8	0.55	4	0.130

- ✓ Only the chromosomes 1, 2, 4, and 8 can be a parent for the next generation

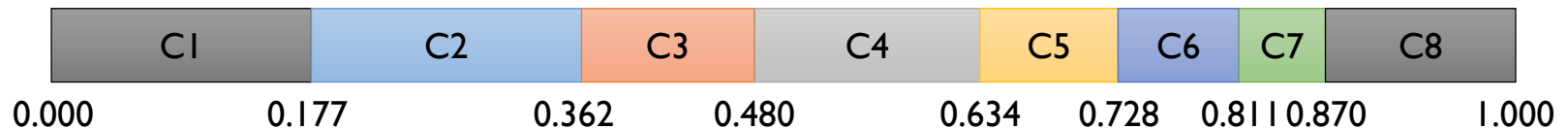
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 1	0	1	1	1	0	1	0	0	1	1
Chromosome 2	1	1	1	1	0	1	0	0	1	1
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1

GA Step 4: Selection

- Example: Selection

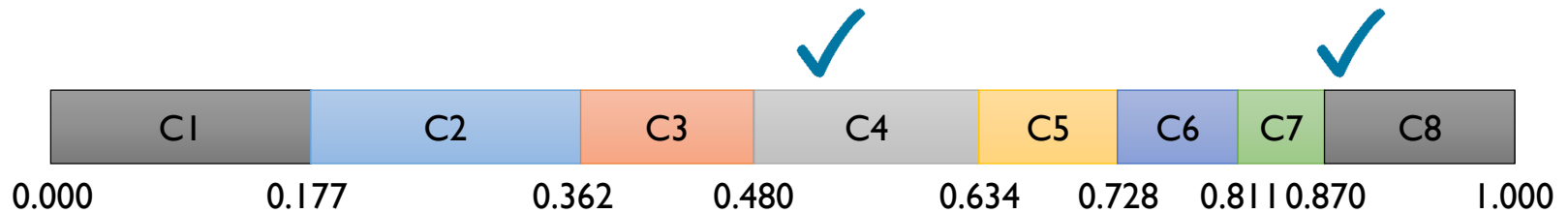
- ✓ Case 2: Probabilistic selection

- Cumulative weight



- For the first set of parents

- Random number initialization: (0.881, 0.499)



- Choose the chromosome 4 and 8

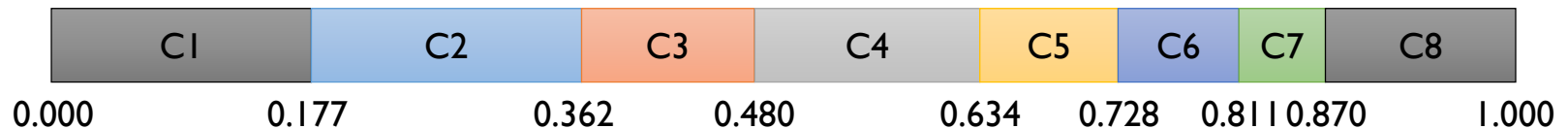
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1

GA Step 4: Selection

- Example: Selection

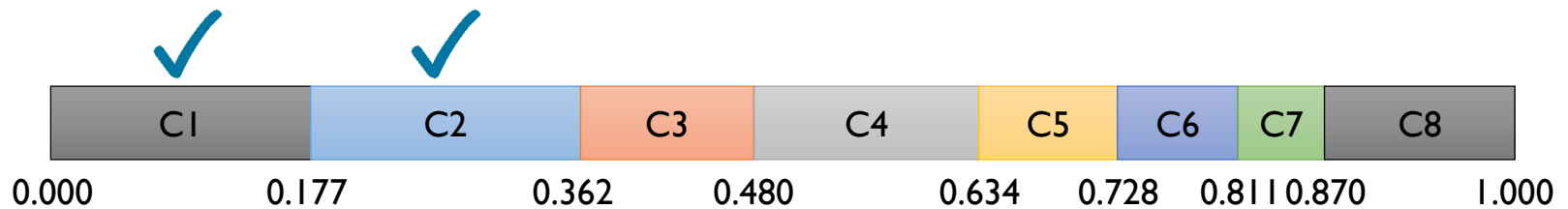
- ✓ Case 2: Probabilistic selection

- Cumulative weight



- For the second set of parents

- Random number initialization: (0.098, 0.252)



- Choose the chromosome 1 and 2

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 1	0	1	1	1	0	1	0	0	1	1
Chromosome 2	1	1	1	1	0	1	0	0	1	1

GA Step 5: Crossover & Mutation

- Crossover (Reproduction)

- ✓ Two child chromosomes are produced from two parent chromosomes
- ✓ The number of crossover points can vary from 1 to n (total number of genes)

- Example with 1 crossover point

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1



	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Child 1	0	1	0	0	1	1	0	1	1	0
Child 2	0	0	1	1	1	0	1	0	1	1

GA Step 5: Crossover & Mutation

- Crossover (Reproduction)

- ✓ Two child chromosomes are produced from two parent chromosomes
- ✓ The number of crossover points can vary from 1 to n (total number of genes)

- Example with 2 crossover point

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1



	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Child 1	0	1	0	1	1	1	0	0	1	1
Child 2	0	0	1	0	1	0	1	1	1	0

GA Step 5: Crossover & Mutation

- Crossover (Reproduction)

- ✓ Two child chromosomes are produced from two parent chromosomes
- ✓ The number of crossover points can vary from 1 to n (total number of genes)

- Example with 10 crossover point

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Chromosome 4	0	0	1	1	1	1	0	1	1	0
Chromosome 8	0	1	0	0	1	0	1	0	1	1
Random number	0.94	0.89	0.27	0.76	0.54	0.5	0.56	0.08	0	0.46



	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Child 1	0	1	1	0	1	0	1	1	1	0
Child 2	0	0	0	1	1	1	0	0	1	1

GA Step 5: Crossover & Mutation

- Mutation

- ✓ Genetic operator used to maintain diversity from one generation of a population of chromosomes to the next
- ✓ Alters one or more gene values in a chromosome from its initial state, which result in entirely new gene values being added to the gene pool
- ✓ By mutation, the current solution can have a chance to escape from the local optima
- ✓ A too mutation rate can increase the time to converge (0.01 can be a good choice)

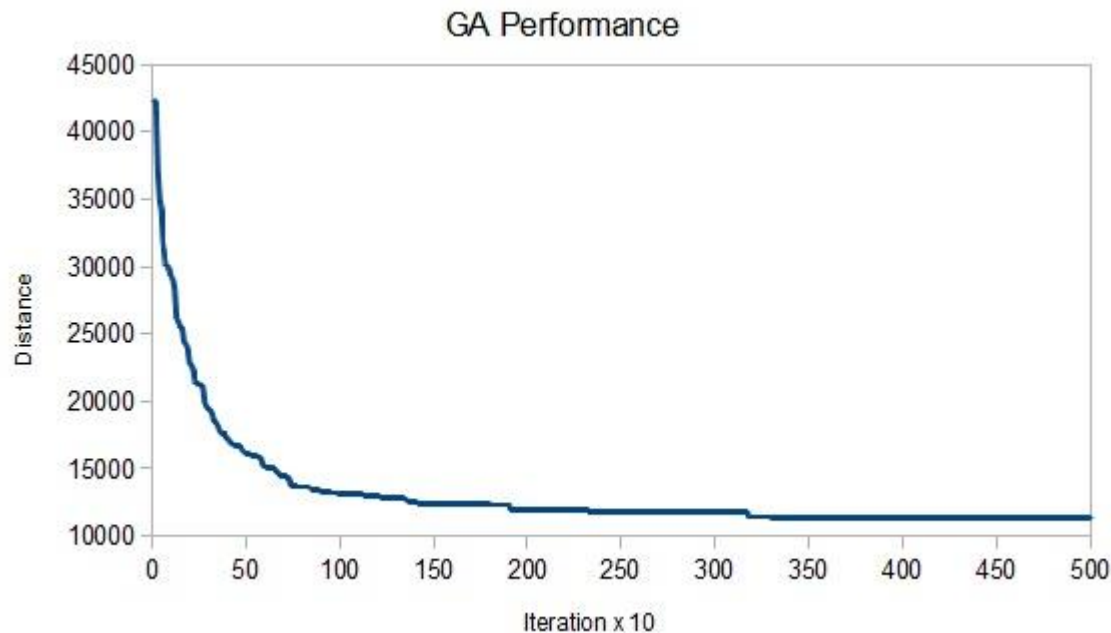
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Child 1	0	1	1	0	1	0	1	1	1	0
Random number	0.43	0.35	0.71	0.54	0.62	0.73	0.71	0.92	0.95	0.91
Child 2	0	0	0	1	1	1	0	0	1	1
Random number	0.91	0.03	0.22	0.96	0.32	0.73	0.43	0.32	0.01	0.04

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
Child 1	0	1	1	0	1	0	1	1	1	0
Child 2	0	0	0	1	1	1	0	0	0	1



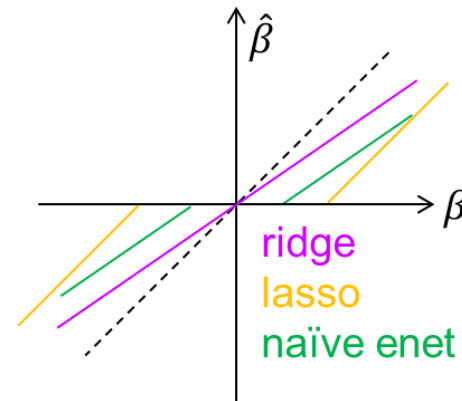
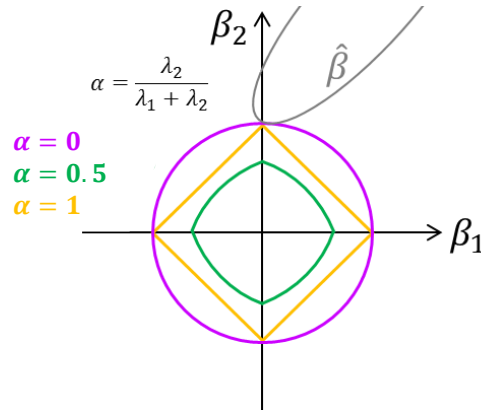
GA Step 5: Find the Best Solution

- Find the best variable subset
 - ✓ Select the chromosome with the highest fitness value after the stopping criteria are satisfied.
 - ✓ Generally, significant fitness improvement occurs in the early stages, which becomes marginal after some generations



Empirical Study

- Compare four variable selection and three shrinkage methods in linear regression
 - ✓ Variable selection: forward selection, backward elimination, stepwise selection, genetic algorithms
 - ✓ Shrinkage methods: ridge regression, lasso regression, elastic net



Ridge	$\hat{\beta} = \min_{\beta} \mathbf{Y} - \mathbf{X}\beta ^2 + \lambda_1 \beta ^2$	shrinkage
Lasso	$\hat{\beta} = \min_{\beta} \mathbf{Y} - \mathbf{X}\beta ^2 + \lambda_2 \beta ^1$	shrinkage, variable selection
Elastic net	$\hat{\beta} = \min_{\beta} \mathbf{Y} - \mathbf{X}\beta ^2 + \lambda_2 \beta ^1 + \lambda_1 \beta ^2$	shrinkage, variable selection, grouping effect

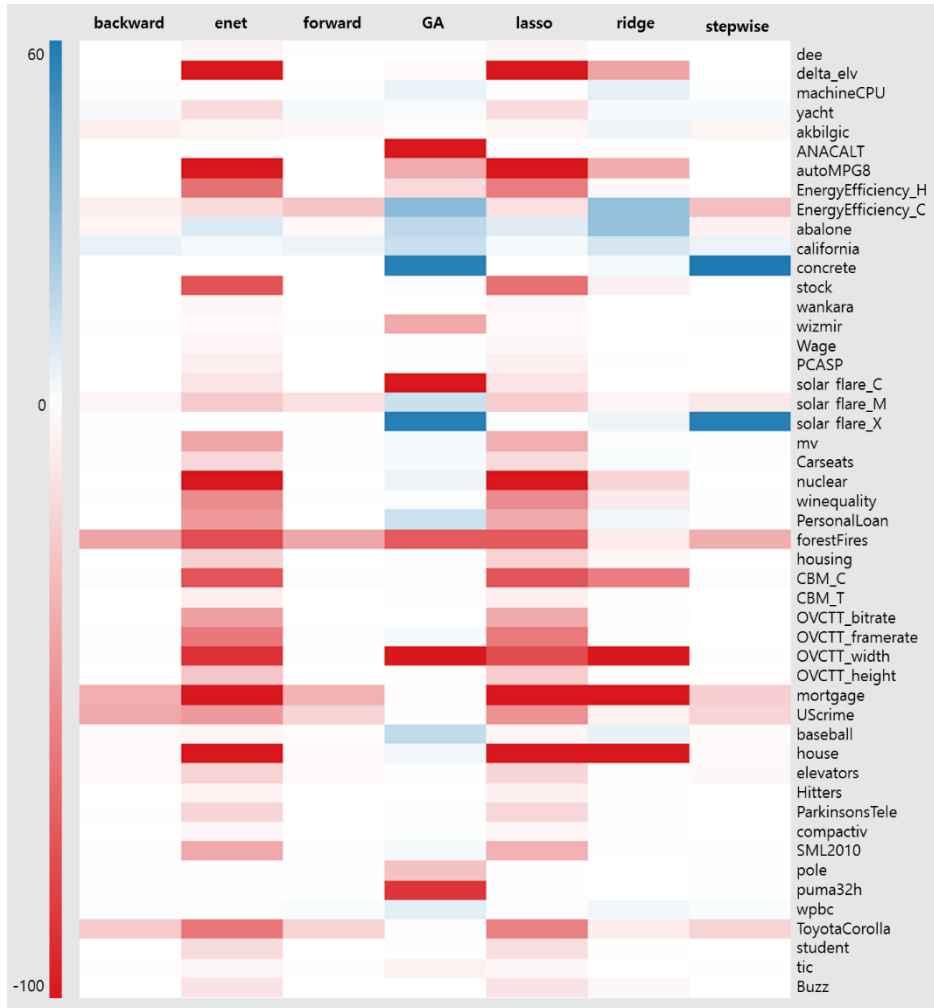
Empirical Study

- Data sets: 49 benchmark datasets

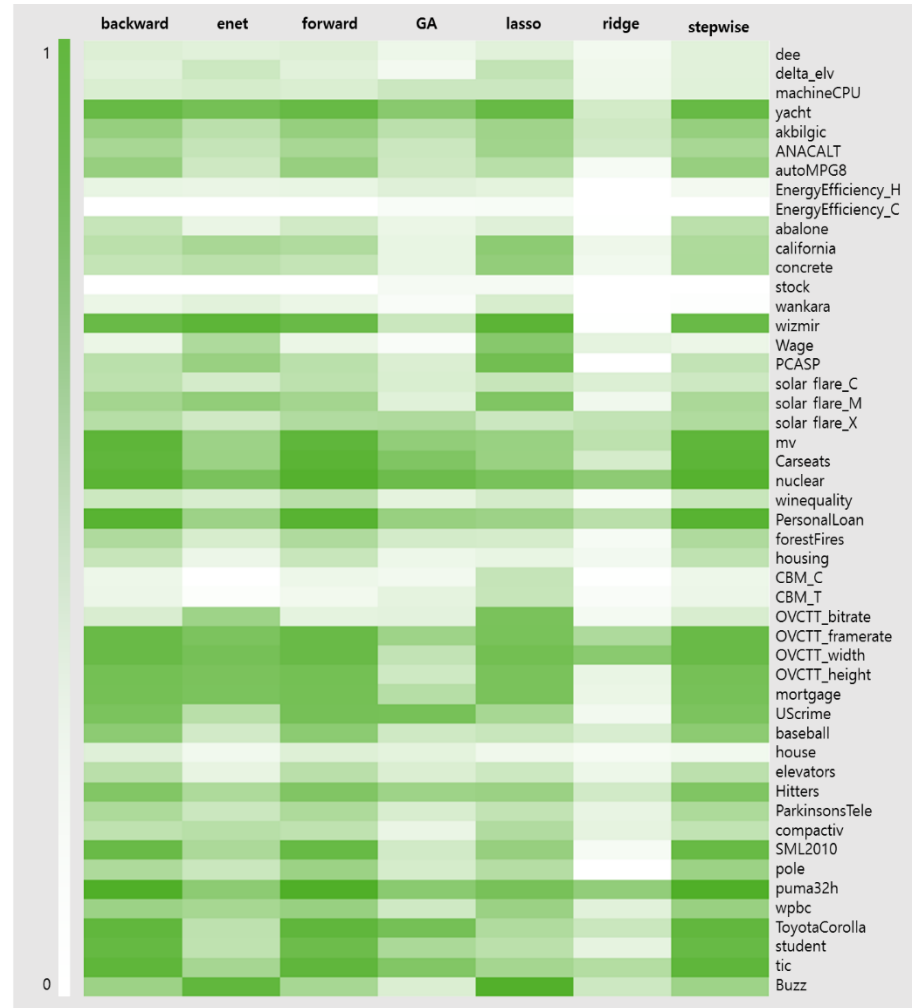
Dataset	source	records	variables	Dataset	source	records	variables
abalone	KEEL	4,177	9	OVCTT_bitrate	UCI	68,784	16
akbilgic	UCI	536	8	OVCTT_framerate	UCI	68,784	16
ANACALT	KEEL	4,052	8	OVCTT_height	UCI	68,784	16
autoMPG8	KEEL	392	8	OVCTT_width	UCI	68,784	16
baseball	KEEL	336	17	ParkinsonsTele	UCI	5,875	21
Buzz	UCI	28,179	95	PersonalLoan	etc.	2,500	13
california	KEEL	20,640	9	PCASP	UCI	45,730	10
Carseats	R	400	11	pole	KEEL	14,998	27
CBM_C	UCI	11,934	15	puma32h	KEEL	4,124	33
CBM_T	UCI	11,934	15	SML2010	UCI	4,137	24
compactiv	KEEL	8,192	22	solar flare_C	UCI	323	11
concrete	KEEL	1,030	9	solar flare_M	UCI	323	11
dee	KEEL	365	7	solar flare_X	UCI	323	11
delta_elv	KEEL	9,517	7	stock	KEEL	950	10
elevators	KEEL	16,599	19	student	UCI	382	51
EnergyEfficiency_C	UCI	768	9	tic	KEEL	9,822	86
EnergyEfficiency_H	UCI	768	9	ToyotaCorolla	etc.	1,436	34
forestFires	KEEL	517	13	UScrime	R	47	16
Hitters	R	263	20	Wage	R	3,000	10
house	KEEL	22,784	17	wankara	KEEL	1,609	10
housing	UCI	506	14	winequality	UCI	6,497	12
machineCPU	KEEL	209	7	wizmir	KEEL	1,461	10
mortgage	KEEL	1,049	16	wdbc	UCI	194	34
mv	KEEL	40,768	11	yacht	UCI	308	7
nuclear	R	32	11				

Empirical Study

Error Rate Improvement



Variable Reduction Ratio



Empirical Study

- Rankings in terms of
 - ✓ (1) Error rate improvement
 - ✓ (2) Variable reduction rate
 - ✓ (3) Computational efficiency

Variable selection technique	Error rate improvement	Variable reduction rate	Computational efficiency
Forward	5	4	1
Backward	4	3	2
Stepwise	3	2	6
GA	1	6	7
Ridge	2	7	5
LASSO	7	1	3
Enet	6	5	4



References

Research Papers

- Siedlecki, W. Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10(5): 335-347

Other materials

- Figure in the title page: <https://pastmike.com/what-is-a-genetic-algorithm/>