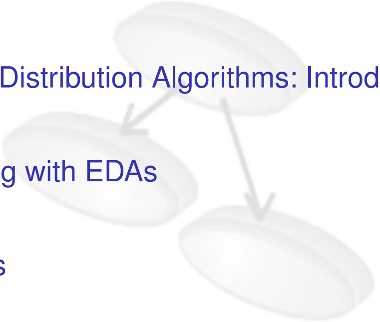
Sintetizando los Algoritmos Evolutivos: Algoritmos de Estimación de Distribuciones

Jose A. Lozano

Intelligent Systems Group
The University of the Basque Country

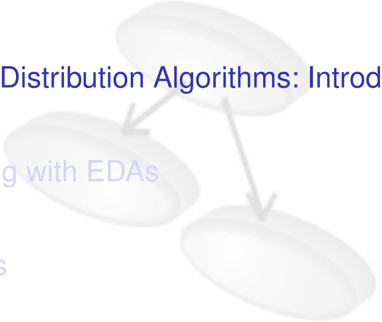
EVIA 2014
(A Coruña, September 3rd, 2014)

Outline of the presentation



1. Estimation of Distribution Algorithms: Introduction
2. Protein Folding with EDAs
3. Miscellaneous
4. Conclusions

Outline of the presentation



1 Estimation of Distribution Algorithms: Introduction

2 Protein Folding with EDAs

3 Miscellaneous

4 Conclusions

Evolutionary Computation

Genetic Algorithms (GAs)

- Based on mimicking natural evolution
- Keep a population of solutions at each step
- Use of reproduction operators: crossover and mutation
- Mathematically modeled as a Markov chain

Estimation of Distribution Algorithms (EDAs): The basics

Motivation: Drawbacks of Genetic Algorithms

- Too much heuristic algorithms
- Many parameters to set up
- Difficult to define good crossover and mutation operators
- Bad results in some trivial problems
- Lack of rigorous mathematical analysis

Basic Characteristics

- Delete the reproduction operators of GAs:
    - Learn a probability distribution from the selected individuals
    - Sample the probability distribution to obtain the new population

Estimation of Distribution Algorithms (EDAs): The basics

Motivation: Drawbacks of Genetic Algorithms

- Too much heuristic algorithms
- Many parameters to set up
- Difficult to define good crossover and mutation operators
- Bad results in some trivial problems
- Lack of rigorous mathematical analysis

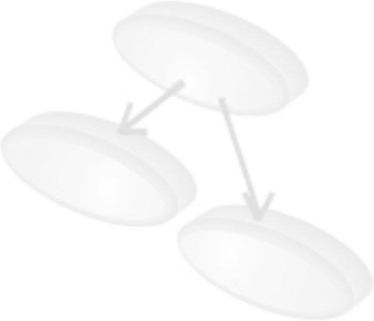
Basic Characteristics

- Delete the reproduction operators of GAs:
  - Learn a probability distribution from the selected individuals
  - Sample the probability distribution to obtain the new population

Optimization of OneMax with EDAs

$$max\ h(\boldsymbol{x}) = \sum_{i=1}^{6} x_i$$

with $x_i = 0, 1$



Optimization of OneMax with EDAs

$$max\ h(\boldsymbol{x}) = \sum_{i=1}^{6} x_i$$

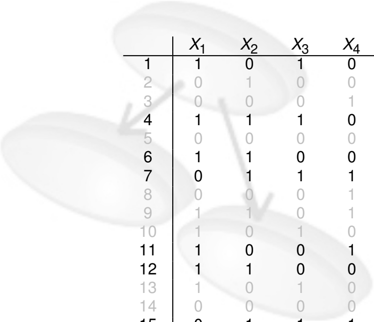
$$\text{with } x_i = 0, 1$$

	X_1	X_2	X_3	X_4	X_5	X_6	$h(\boldsymbol{x})$
1	1	0	1	0	1	0	3
2	0	1	0	0	1	0	2
3	0	0	0	1	0	0	1
4	1	1	1	0	0	1	4
5	0	0	0	0	0	1	1
6	1	1	0	0	1	1	4
7	0	1	1	1	1	1	5
8	0	0	0	1	0	0	1
9	1	1	0	1	0	0	3
10	1	0	1	0	0	0	2
11	1	0	0	1	1	1	4
12	1	1	0	0	0	1	3
13	1	0	1	0	0	0	2
14	0	0	0	0	1	1	2
15	0	1	1	1	1	1	5
16	0	0	0	1	0	0	1
17	1	1	1	1	1	0	5
18	0	1	0	1	1	0	3
19	1	0	1	1	1	1	5
20	1	0	1	1	0	0	3

Optimization of OneMax with EDAs

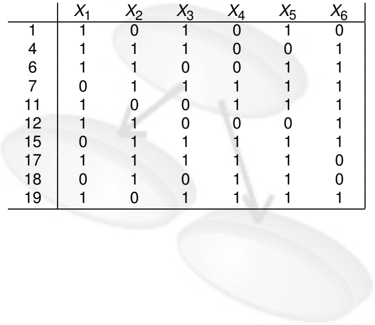
$$max\ h(\boldsymbol{x}) = \sum_{i=1}^{6} x_i$$

with $x_i = 0, 1$



	X_1	X_2	X_3	X_4	X_5	X_6	$h(\boldsymbol{x})$
1	1	0	1	0	1	0	3
2	0	1	0	0	1	0	2
3	0	0	0	1	0	0	1
4	1	1	1	0	0	1	4
5	0	0	0	0	0	1	1
6	1	1	0	0	1	1	4
7	0	1	1	1	1	1	5
8	0	0	0	1	0	0	1
9	1	1	0	1	0	0	3
10	1	0	1	0	0	0	2
11	1	0	0	1	1	1	4
12	1	1	0	0	0	1	3
13	1	0	1	0	0	0	2
14	0	0	0	0	1	1	2
15	0	1	1	1	1	1	5
16	0	0	0	1	0	0	1
17	1	1	1	1	1	0	5
18	0	1	0	1	1	0	3
19	1	0	1	1	1	1	5
20	1	0	1	1	0	0	3

Learning the probability distribution



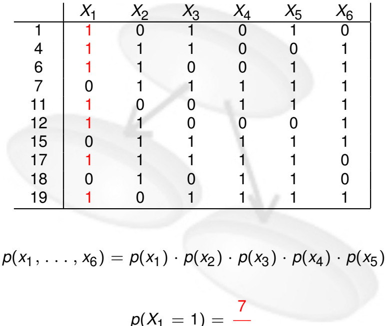
	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

Learning the probability distribution

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

Learning the probability distribution



	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10}$$

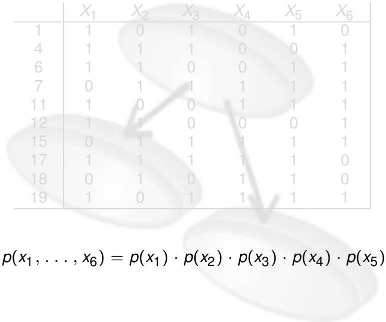
Learning the probability distribution

	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

Learning the probability distribution



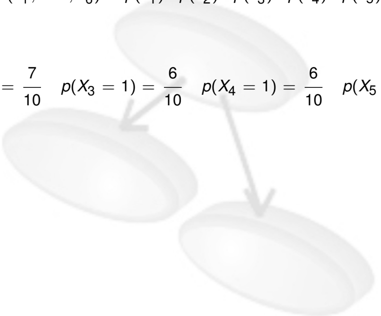
	X_1	X_2	X_3	X_4	X_5	X_6
1	1	0	1	0	1	0
4	1	1	1	0	0	1
6	1	1	0	0	1	1
7	0	1	1	1	1	1
11	1	0	0	1	1	1
12	1	1	0	0	0	1
15	0	1	1	1	1	1
17	1	1	1	1	1	0
18	0	1	0	1	1	0
19	1	0	1	1	1	1

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

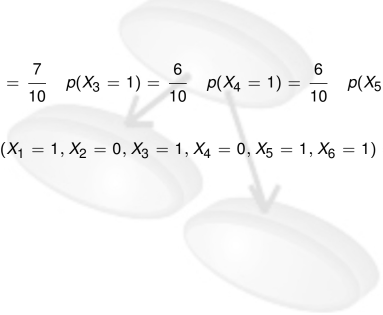
Learning the probability distribution

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$



$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

Learning the probability distribution



$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

Learning the probability distribution

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

$$p(X_1 = 1) \cdot p(X_2 = 0) \cdot p(X_3 = 1) \cdot p(X_4 = 0) \cdot p(X_5 = 1) \cdot p(X_6 = 1) =$$

Learning the probability distribution

$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

$$p(X_1 = 1) \cdot p(X_2 = 0) \cdot p(X_3 = 1) \cdot p(X_4 = 0) \cdot p(X_5 = 1) \cdot p(X_6 = 1) =$$

$$\downarrow$$

$$\frac{7}{10}$$

Learning the probability distribution

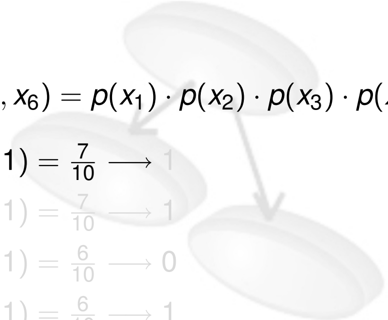
$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$p(X_1 = 1) = \frac{7}{10} \quad p(X_2 = 1) = \frac{7}{10} \quad p(X_3 = 1) = \frac{6}{10} \quad p(X_4 = 1) = \frac{6}{10} \quad p(X_5 = 1) = \frac{8}{10} \quad p(X_6 = 1) = \frac{7}{10}$$

$$p(X_1 = 1, X_2 = 0, X_3 = 1, X_4 = 0, X_5 = 1, X_6 = 1) =$$

$$\begin{array}{ccccccccccccc} p(X_1 = 1) & \cdot & p(X_2 = 0) & \cdot & p(X_3 = 1) & \cdot & p(X_4 = 0) & \cdot & p(X_5 = 1) & \cdot & p(X_6 = 1) & = \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\ \frac{7}{10} & \cdot & \frac{3}{10} & \cdot & \frac{6}{10} & \cdot & \frac{4}{10} & \cdot & \frac{8}{10} & \cdot & \frac{7}{10} & \end{array}$$

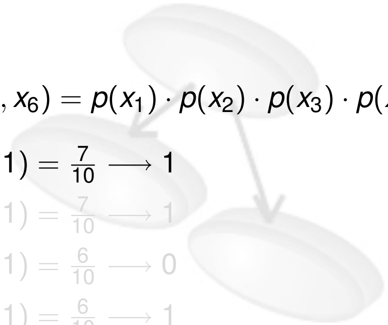
Obtaining the new population



$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

0.23	$p(X_1 = 1) = \frac{7}{10}$	$\longrightarrow 1$
0.45	$p(X_2 = 1) = \frac{7}{10}$	$\longrightarrow 1$
0.89	$p(X_3 = 1) = \frac{6}{10}$	$\longrightarrow 0$
0.12	$p(X_4 = 1) = \frac{6}{10}$	$\longrightarrow 1$
0.98	$p(X_5 = 1) = \frac{8}{10}$	$\longrightarrow 0$
0.54	$p(X_6 = 1) = \frac{7}{10}$	$\longrightarrow 1$

Obtaining the new population



$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

0.23 $p(X_1 = 1) = \frac{7}{10} \longrightarrow 1$

0.45 $p(X_2 = 1) = \frac{7}{10} \longrightarrow 1$

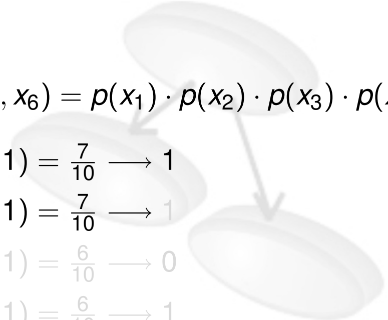
0.89 $p(X_3 = 1) = \frac{6}{10} \longrightarrow 0$

0.12 $p(X_4 = 1) = \frac{6}{10} \longrightarrow 1$

0.98 $p(X_5 = 1) = \frac{8}{10} \longrightarrow 0$

0.54 $p(X_6 = 1) = \frac{7}{10} \longrightarrow 1$

Obtaining the new population



$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

0.23 $\quad p(X_1 = 1) = \frac{7}{10} \longrightarrow 1$

0.45 $\quad p(X_2 = 1) = \frac{7}{10} \longrightarrow 1$

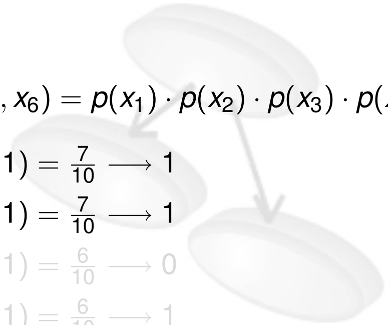
0.89 $\quad p(X_3 = 1) = \frac{6}{10} \longrightarrow 0$

0.12 $\quad p(X_4 = 1) = \frac{6}{10} \longrightarrow 1$

0.98 $\quad p(X_5 = 1) = \frac{8}{10} \longrightarrow 0$

0.54 $\quad p(X_6 = 1) = \frac{7}{10} \longrightarrow 1$

Obtaining the new population



$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

$$0.23 \qquad p(X_1 = 1) = \frac{7}{10} \longrightarrow 1$$

$$0.45 \qquad p(X_2 = 1) = \frac{7}{10} \longrightarrow 1$$

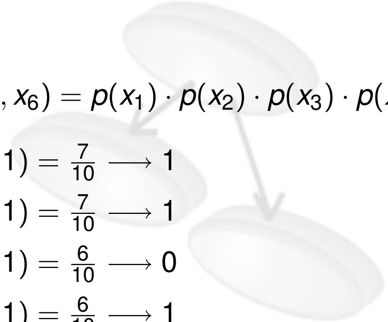
$$0.89 \qquad p(X_3 = 1) = \frac{6}{10} \longrightarrow 0$$

$$0.12 \qquad p(X_4 = 1) = \frac{6}{10} \longrightarrow 1$$

$$0.98 \qquad p(X_5 = 1) = \frac{8}{10} \longrightarrow 0$$

$$0.54 \qquad p(X_6 = 1) = \frac{7}{10} \longrightarrow 1$$

Obtaining the new population



$$p(\boldsymbol{x}) = p(x_1, \ldots, x_6) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \cdot p(x_6)$$

0.23 $\quad p(X_1 = 1) = \frac{7}{10} \longrightarrow 1$

0.45 $\quad p(X_2 = 1) = \frac{7}{10} \longrightarrow 1$

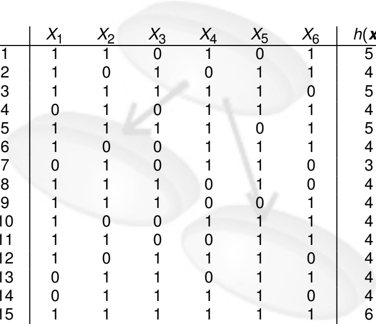
0.89 $\quad p(X_3 = 1) = \frac{6}{10} \longrightarrow 0$

0.12 $\quad p(X_4 = 1) = \frac{6}{10} \longrightarrow 1$

0.98 $\quad p(X_5 = 1) = \frac{8}{10} \longrightarrow 0$

0.54 $\quad p(X_6 = 1) = \frac{7}{10} \longrightarrow 1$

Obtaining the new population



	X_1	X_2	X_3	X_4	X_5	X_6	$h(\boldsymbol{x})$
1	1	1	0	1	0	1	5
2	1	0	1	0	1	1	4
3	1	1	1	1	1	0	5
4	0	1	0	1	1	1	4
5	1	1	1	1	0	1	5
6	1	0	0	1	1	1	4
7	0	1	0	1	1	0	3
8	1	1	1	0	1	0	4
9	1	1	1	0	0	1	4
10	1	0	0	1	1	1	4
11	1	1	0	0	1	1	4
12	1	0	1	1	1	0	4
13	0	1	1	0	1	1	4
14	0	1	1	1	1	0	4
15	1	1	1	1	1	1	6
16	0	1	1	0	1	1	4
17	1	1	1	1	1	0	5
18	0	1	0	0	1	0	2
19	0	0	1	1	0	1	3
20	1	1	0	1	1	1	5

Pseudocode for EDAs

```
Obtain an initial population of individuals $D_0$
Repeat until a stopping criterion is met
    Select from $D_i$ a subset of individuals $D_i^S$
    Learn a probability distribution $p_i(\boldsymbol{x})$ from $D_i^S$
    Sample $p_i(\boldsymbol{x})$ to obtain $D_{i+1/2}$
    Create the new population $D_{i+1}$ from $D_i$ and $D_{i+1/2}$
```

What components should be set in EDAs?

- Population size
- Selection operator
- Probabilistic model
- Learning algorithm
- Sampling algorithm

What components should be set in EDAs?

- Population size
- Selection operator
- Probabilistic model
- Learning algorithm
- Sampling algorithm

What components should be set in EDAs?

- Population size
- Selection operator
- Probabilistic model
- Learning algorithm
- Sampling algorithm

What components should be set in EDAs?

- Population size
- Selection operator
- Probabilistic model
- Learning algorithm
- Sampling algorithm

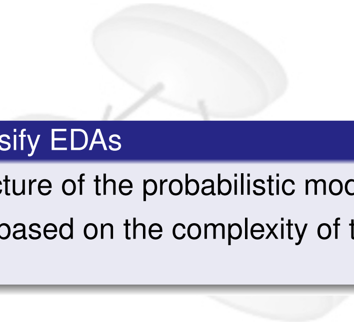
What components should be set in EDAs?

- Population size
- Selection operator
- Probabilistic model
- Learning algorithm
- Sampling algorithm

What components should be set in EDAs?

- Population size
- Selection operator
- Probabilistic model
- Learning algorithm
- Sampling algorithm

Classification of EDAs



Two criteria to classify EDAs

- Keep the structure of the probabilistic model fixed
- Classification based on the complexity of the model structure

EDAs with fixed structure

- The structure of the probabilistic model keeps fixed during the search
- At each generation only parametric learning is carried out
- The fixed probabilistic model tries to mimic the structure of the function to optimize
- Most of the work has been done in the optimization of additive decomposable functions

EDAs that change the structure

Classification based on the model structure

- Modify the structure ((in)dependences) of the probability model at each iteration:
  1. Univariate model: all the variables are independent (UMDA, PBIL, CGA,...)(*)
  2. Bivariate model: consider second-order statistics (TREE, MIMIC)
  3. Unrestricted model: use of Bayesian networks (EBNA, LFDA, BOA)

EDAs that change the structure

Classification based on the model structure

- Modify the structure ((in)dependences) of the probability model at each iteration:
    1. Univariate model: all the variables are independent (UMDA, PBIL, CGA,...)(*)
    2. Bivariate model: consider second-order statistics (TREE, MIMIC)
    3. Unrestricted model: use of Bayesian networks (EBNA, LFDA, BOA)

UPV - EHU

EDAs that change the structure

Classification based on the model structure

- Modify the structure ((in)dependences) of the probability model at each iteration:
  1. Univariate model: all the variables are independent (UMDA, PBIL, CGA,...)(*)
  2. Bivariate model: consider second-order statistics (TREE, MIMIC)
  3. Unrestricted model: use of Bayesian networks (EBNA, LFDA, BOA)

EDAs that change the structure

Classification based on the model structure

- Modify the structure ((in)dependences) of the probability model at each iteration:
    1. Univariate model: all the variables are independent (UMDA, PBIL, CGA,...)(*)
    2. Bivariate model: consider second-order statistics (TREE, MIMIC)
    3. Unrestricted model: use of Bayesian networks (EBNA, LFDA, BOA)

EDAs that change the structure

Classification based on the model structure

- Modify the structure ((in)dependences) of the probability model at each iteration:
    1. Univariate model: all the variables are independent (UMDA, PBIL, CGA,...)(*)
    2. Bivariate model: consider second-order statistics (TREE, MIMIC)
    3. Unrestricted model: use of Bayesian networks (EBNA, LFDA, BOA)

No dependencies

Univariate Marginal Distribution Algorithm (UMDA)

- Probabilistic model: $p_l(\boldsymbol{x}) = \prod_{i=1}^{n} p_l(x_i)$
- Parameter learning: maximum likelihood (frequency counts)

Probabilistic Based Incremental Learning (PBIL)

- Probabilistic model: same as UMDA
- Parameter learning: maximum likelihood (frequency counts)
- Rule to modify the probabilistic model:

$$p_{l+1}(\boldsymbol{x}) = (1-\alpha)p_l(\mathbf{x}) + \alpha \frac{1}{N} \sum_{k=1}^{N} x_{k:M}^{l}$$

Bivariate model

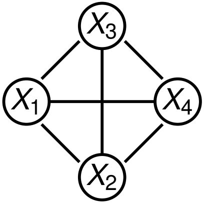
TREE

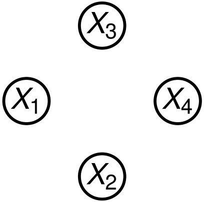
- Probabilistic model: $p(\boldsymbol{x}) = \prod_{i=1}^{n} p(x_i|x_{j(i)})$
- Structural learning: Chow and Liu algorithm
- Parametric learning: Maximum likelihood
- Sampling: Probabilistic logic sampling

Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

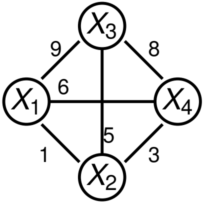


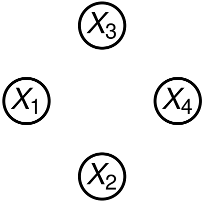


Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$



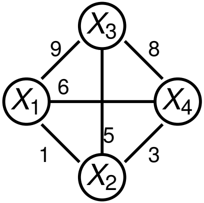


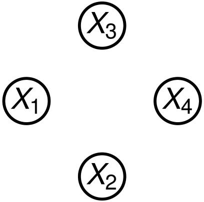
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree



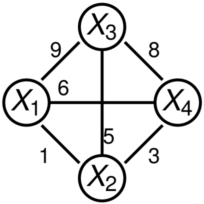


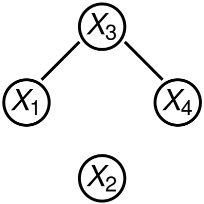
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree



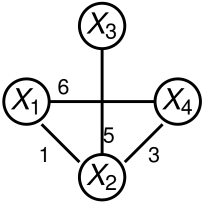


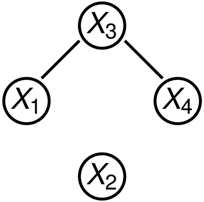
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree





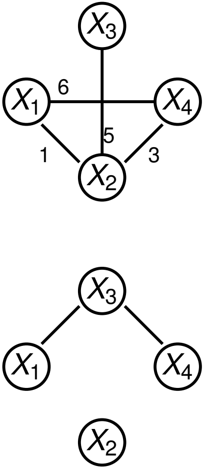
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree

iii) Repeat until there are $n - 1$ edges in the tree:
Assign the edge with the highest MI value to the tree if it does not form a loop, otherwise delete it



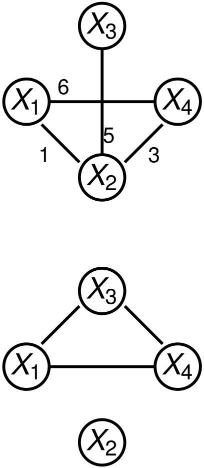
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree

iii) Repeat until there are $n - 1$ edges in the tree:
Assign the edge with the highest MI value to the tree if it does not form a loop, otherwise delete it



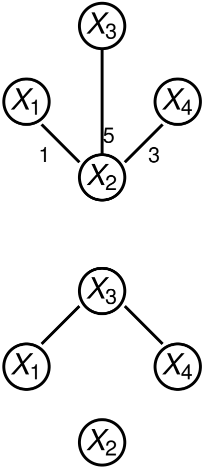
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree

iii) Repeat until there are $n-1$ edges in the tree:
Assign the edge with the highest MI value to the tree if it does not form a loop, otherwise delete it



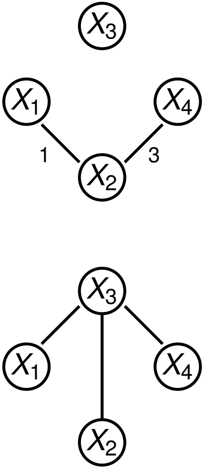
Chow and Liu algorithm

i) For each X_i, X_j compute:

$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree

iii) Repeat until there are $n-1$ edges in the tree:
Assign the edge with the highest MI value to the tree if it does not form a loop, otherwise delete it



Chow and Liu algorithm

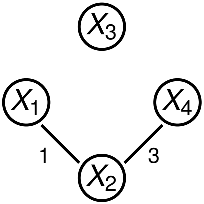
i) For each X_i, X_j compute:

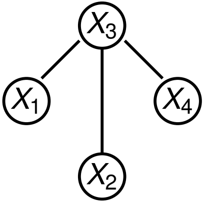
$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree

iii) Repeat until there are $n-1$ edges in the tree:
Assign the edge with the highest MI value to the tree if it does not form a loop, otherwise delete it

iv) Give orientation to the edges by choosing a node as the root





Chow and Liu algorithm

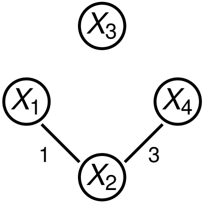
i) For each X_i, X_j compute:

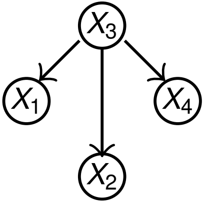
$$MI(X_i, X_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

ii) Assign the two edges with the highest MI to the tree

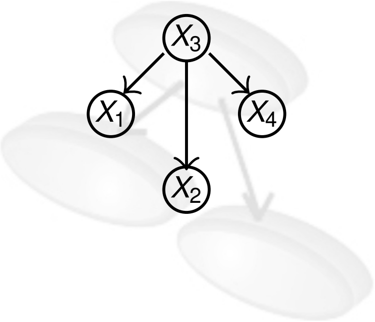
iii) Repeat until there are $n - 1$ edges in the tree:
Assign the edge with the highest MI value to the tree if it does not form a loop, otherwise delete it

iv) Give orientation to the edges by choosing a node as the root

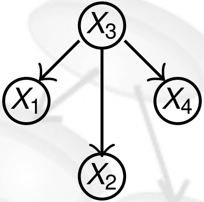




Chow and Liu algorithm



Chow and Liu algorithm

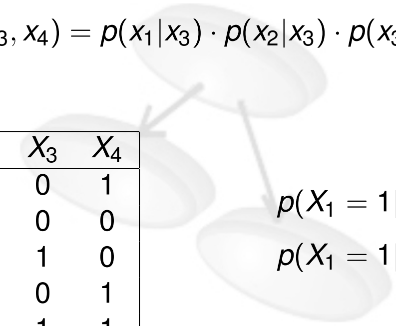


$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

UPV - EHU

Parametric learning in TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$



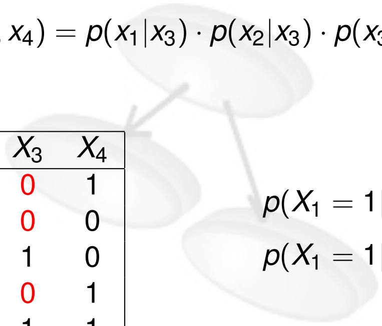
X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0)$$

$$p(X_1 = 1|X_3 = 1)$$

Parametric learning in TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$



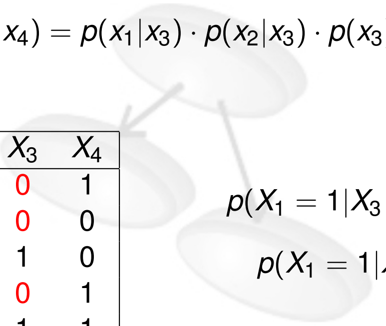
X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0)$$

$$p(X_1 = 1|X_3 = 1)$$

Parametric learning in TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$



X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2}{3}$$

$$p(X_1 = 1|X_3 = 1)$$

Parametric learning in TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

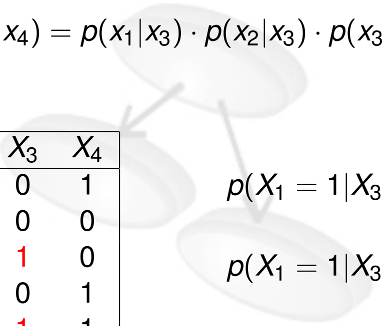
X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1 | X_3 = 0) = \frac{2}{3}$$

$$p(X_1 = 1 | X_3 = 1)$$

Parametric learning in TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$



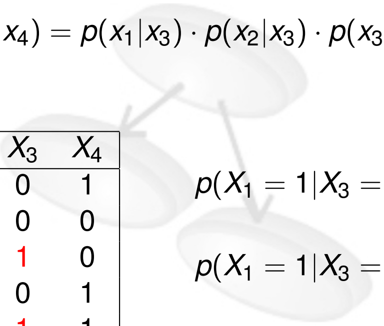
X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2}{3}$$

$$p(X_1 = 1|X_3 = 1) = \frac{0}{2}$$

Parametric learning in TREE

$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$



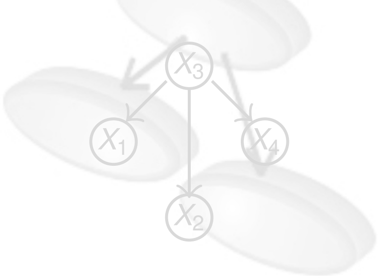
X_1	X_2	X_3	X_4
1	0	0	1
1	1	0	0
0	1	1	0
0	0	0	1
0	1	1	1

$$p(X_1 = 1|X_3 = 0) = \frac{2+\epsilon}{3+2\epsilon}$$

$$p(X_1 = 1|X_3 = 1) = \frac{0+\epsilon}{2+2\epsilon}$$

Sampling in TREE

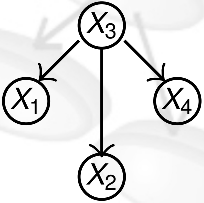
- Order the variables in such a way that parents variables come before children variables



- In our case: X_3, X_1, X_2, X_4
- Variables are sampled in this order

Sampling in TREE

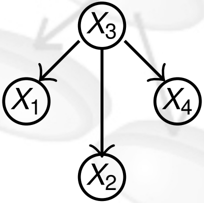
- Order the variables in such a way that parents variables come before children variables



- In our case: X_3, X_1, X_2, X_4
- Variables are sampled in this order

Sampling in TREE

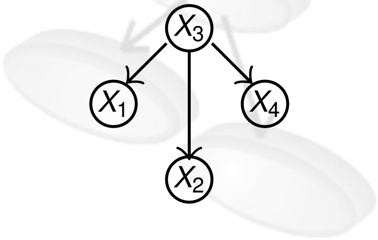
- Order the variables in such a way that parents variables come before children variables



- In our case: X_3, X_1, X_2, X_4
- Variables are sampled in this order

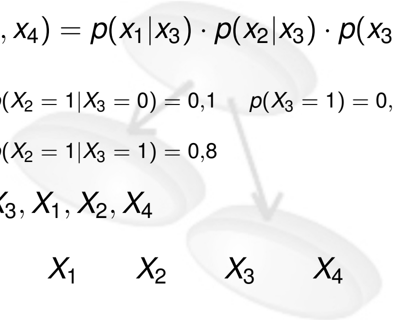
Sampling in TREE

- Order the variables in such a way that parents variables come before children variables



- In our case: X_3, X_1, X_2, X_4
- Variables are sampled in this order

Sampling in TREE



$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

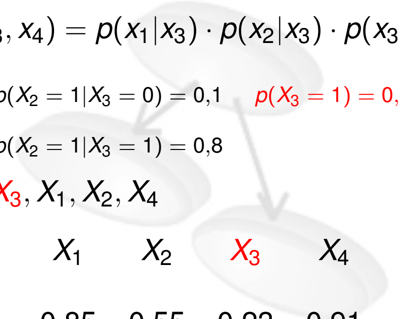
$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Variable order: X_3, X_1, X_2, X_4

$$\begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \\ 0{,}85 & 0{,}55 & 0{,}23 & 0{,}91 \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array}$$

Sampling in TREE



$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

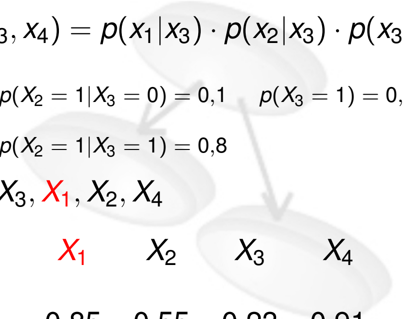
$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Variable order: X_3, X_1, X_2, X_4

X_1	X_2	X_3	X_4
0,85	0,55	0,23	0,91
↓	↓	↓	↓
		1	

Sampling in TREE



$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

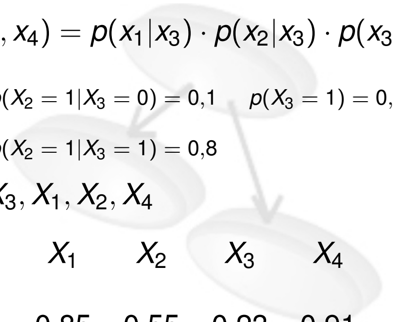
$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Variable order: X_3, X_1, X_2, X_4

X_1	X_2	X_3	X_4
0,85	0,55	0,23	0,91
↓	↓	↓	↓
0		1	

Sampling in TREE



$$p(x_1, x_2, x_3, x_4) = p(x_1|x_3) \cdot p(x_2|x_3) \cdot p(x_3) \cdot p(x_4|x_3)$$

$p(X_1 = 1|X_3 = 0) = 0{,}3$ $p(X_2 = 1|X_3 = 0) = 0{,}1$ $p(X_3 = 1) = 0{,}7$ $p(X_4 = 1|X_3 = 0) = 0{,}5$

$p(X_1 = 1|X_3 = 1) = 0{,}2$ $p(X_2 = 1|X_3 = 1) = 0{,}8$ $p(X_4 = 1|X_3 = 1) = 0{,}6$

Variable order: X_3, X_1, X_2, X_4

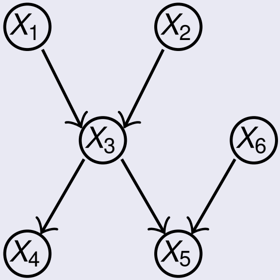
X_1	X_2	X_3	X_4
0,85	0,55	0,23	0,91
↓	↓	↓	↓
0	1	1	0

Unrestrictive models:
Estimation of Bayesian Network Algorithm (EBNA)

Characteristics

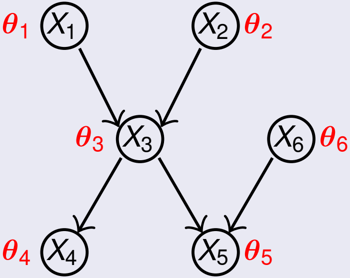
- At each step a Bayesian network is learnt
- Parameters are learnt using maximum likelihood
- Probabilistic Logic Sampling is used to sample new solutions

Bayesian networks



$$p(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3|x_1, x_2) \cdot p(x_4|x_3) \cdot p(x_5|x_3, x_6) \cdot p(x_6)$$

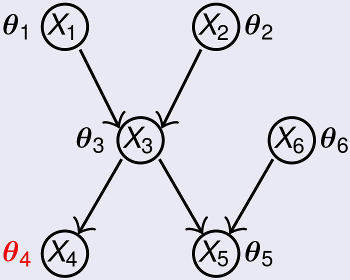
Bayesian networks



$$p(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3|x_1, x_2) \cdot p(x_4|x_3) \cdot p(x_5|x_3, x_6) \cdot p(x_6)$$

Bayesian networks

$$\boldsymbol{\theta}_4 = (\boldsymbol{\theta}_{41}, \boldsymbol{\theta}_{42})$$

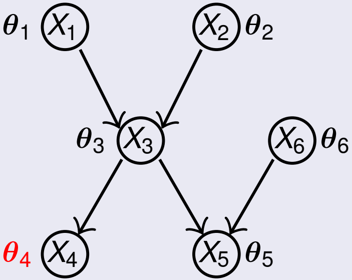


$$p(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3|x_1, x_2) \cdot p(x_4|x_3) \cdot p(x_5|x_3, x_6) \cdot p(x_6)$$

Bayesian networks

$$\boldsymbol{\theta}_4 = (\boldsymbol{\theta}_{41}, \boldsymbol{\theta}_{42})$$

$$\left\{ \begin{array}{l} \theta_{411} = p(X_4 = 0 \mid X_3 = 0) \\ \theta_{412} = p(X_4 = 1 \mid X_3 = 0) \end{array} \right.$$



$$p(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3|x_1, x_2) \cdot p(x_4|x_3) \cdot p(x_5|x_3, x_6) \cdot p(x_6)$$

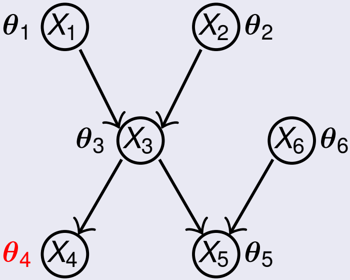
Bayesian networks

$$\boldsymbol{\theta}_4 = (\boldsymbol{\theta}_{41}, \boldsymbol{\theta}_{42})$$

$$\begin{cases} \theta_{411} = p(X_4 = 0 \mid X_3 = 0) \\ \theta_{412} = p(X_4 = 1 \mid X_3 = 0) \end{cases}$$

General case

$$\theta_{ijk} = p(X_i = x_i^k | \boldsymbol{Pa}_i = \boldsymbol{pa}_i^j)$$



$$p(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3|x_1, x_2) \cdot p(x_4|x_3) \cdot p(x_5|x_3, x_6) \cdot p(x_6)$$

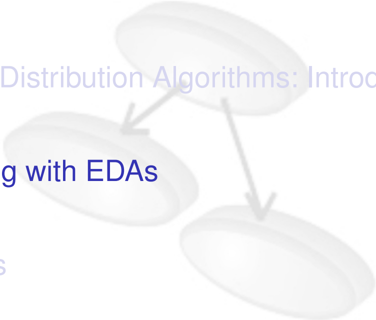
Which are the benefits of using explicit distributions?

- Search for solutions taking into account correlation between variables
- Incorporate information from the fitness function to the probability model:
  - Relations between the variables
  - Structural information
- It is possible to extract information about the objective function from the learnt model:
  - Relations between variables
  - Characteristics of the search space
  - A model of the process that is being optimized

Which are the benefits of using explicit distributions?

- Search for solutions taking into account correlation between variables
- Incorporate information from the fitness function to the probability model:
  - Relations between the variables
  - Structural information
- It is possible to extract information about the objective function from the learnt model:
  - Relations between variables
  - Characteristics of the search space
  - A model of the process that is being optimized

Outline of the presentation



1. Estimation of Distribution Algorithms: Introduction
2. Protein Folding with EDAs
3. Miscellaneous
4. Conclusions

Proteins

Introduction

- Proteins are essential components of living organisms
- Formed by a sequence of (20 different types) amino acids or residues
- Under suitable conditions folded in 3-D to form a functional structure
- Each possible protein fold has associated free energy
- The native structure of a protein is the one for which the free energy achieves the global minimum

Proteins

Introduction

- Proteins are essential components of living organisms
- Formed by a sequence of (20 different types) amino acids or residues
- Under suitable conditions folded in 3-D to form a functional structure
- Each possible protein fold has associated free energy
- The native structure of a protein is the one for which the free energy achieves the global minimum

Proteins

Introduction

- Proteins are essential components of living organisms
- Formed by a sequence of (20 different types) amino acids or residues
- Under suitable conditions folded in 3-D to form a functional structure
- Each possible protein fold has associated free energy
- The native structure of a protein is the one for which the free energy achieves the global minimum

Proteins

Introduction

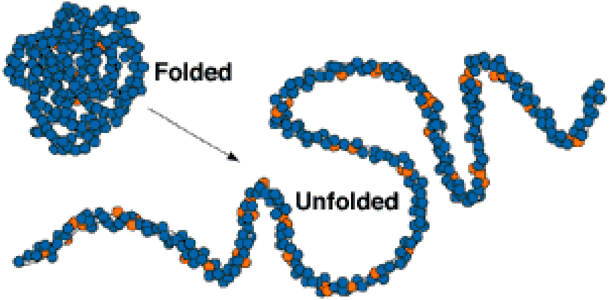
- Proteins are essential components of living organisms
- Formed by a sequence of (20 different types) amino acids or residues
- Under suitable conditions folded in 3-D to form a functional structure
- Each possible protein fold has associated free energy
- The native structure of a protein is the one for which the free energy achieves the global minimum

Proteins

Introduction

- Proteins are essential components of living organisms
- Formed by a sequence of (20 different types) amino acids or residues
- Under suitable conditions folded in 3-D to form a functional structure
- Each possible protein fold has associated free energy
- The native structure of a protein is the one for which the free energy achieves the global minimum

Proteins



Protein Folding

Protein Folding as an Optimization Problem

- Given a sequence of amino acids, which is the native structure of the protein?
- Problems:
  - To define an adequate energy function
  - An exponential number of folding possibilities
- Solutions: use of simplified protein models

Protein Folding

Protein Folding as an Optimization Problem

- Given a sequence of amino acids, which is the native structure of the protein?
- Problems:
    - To define an adequate energy function
    - An exponential number of folding possibilities
- Solutions: use of simplified protein models

UPV - EHU

Protein Folding

Protein Folding as an Optimization Problem

- Given a sequence of amino acids, which is the native structure of the protein?
- Problems:
    - To define an adequate energy function
    - An exponential number of folding possibilities
- Solutions: use of simplified protein models

Protein Folding

Protein Folding as an Optimization Problem

- Given a sequence of amino acids, which is the native structure of the protein?
- Problems:
    - To define an adequate energy function
    - An exponential number of folding possibilities
- Solutions: use of simplified protein models

Protein Folding

Protein Folding as an Optimization Problem

- Given a sequence of amino acids, which is the native structure of the protein?
- Problems:
    - To define an adequate energy function
    - An exponential number of folding possibilities
- Solutions: use of simplified protein models

Protein Folding

Protein Folding as an Optimization Problem

- Given a sequence of amino acids, which is the native structure of the protein?
- Problems:
  - To define an adequate energy function
  - An exponential number of folding possibilities
- Solutions: use of simplified protein models

The Hydrophobic-Polar (HP) Model

Introduction

- The HP model considers two types of residues: hydrophobic (H) residues and hydrophilic or polar (P) residues
- A protein is considered a sequence of these two types of residues, which are located in regular lattice models forming self-avoided paths
- Given two residues we consider two kinds of neighborhoods:
  - Connected neighbors: adjacent in the chain
  - Topological neighbors: adjacent in the lattice but not connected in the chain

The Hydrophobic-Polar (HP) Model

Introduction

- The HP model considers two types of residues: hydrophobic (H) residues and hydrophilic or polar (P) residues
- A protein is considered a sequence of these two types of residues, which are located in regular lattice models forming self-avoided paths
- Given two residues we consider two kinds of neighborhoods:
    - Connected neighbors: adjacent in the chain
    - Topological neighbors: adjacent in the lattice but not connected in the chain

The Hydrophobic-Polar (HP) Model

Introduction

- The HP model considers two types of residues: hydrophobic (H) residues and hydrophilic or polar (P) residues
- A protein is considered a sequence of these two types of residues, which are located in regular lattice models forming self-avoided paths
- Given two residues we consider two kinds of neighborhoods:
  - Connected neighbors: adjacent in the chain
  - Topological neighbors: adjacent in the lattice but not connected in the chain

UPV - EHU

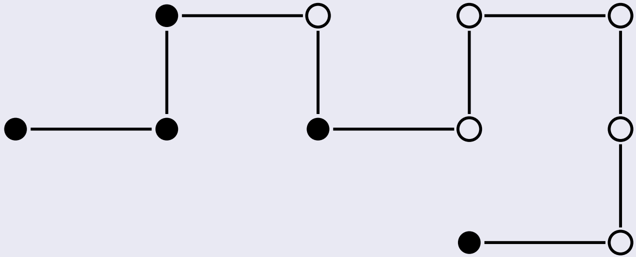
The H-P model

Energy function

- A energy function that measures the interaction between topological neighbor residues is defined as:
  - $\epsilon_{HH} = -1$
  - $\epsilon_{HP} = 0$
  - $\epsilon_{PP} = 0$

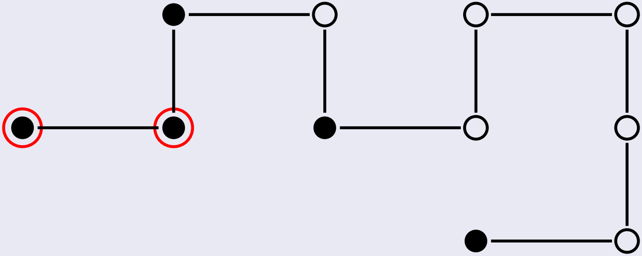
The H-P model

Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



The H-P model

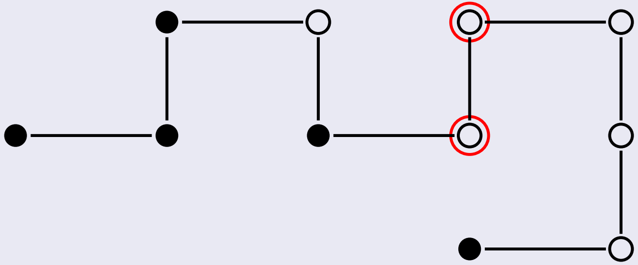
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



connected neighbors

The H-P model

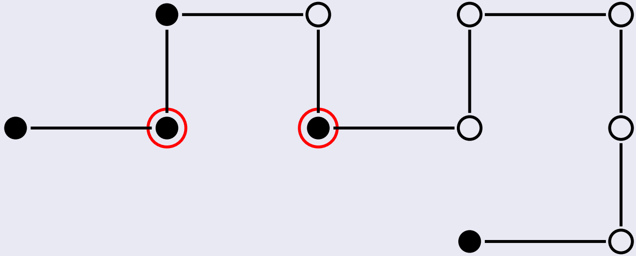
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



connected neighbors

The H-P model

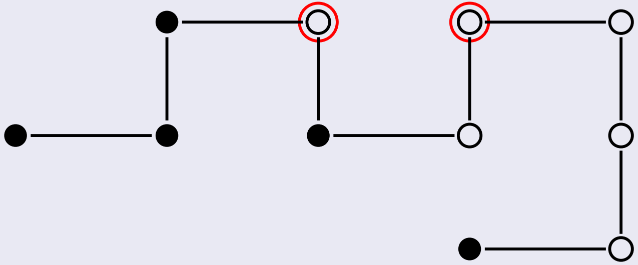
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



topological neighbors

The H-P model

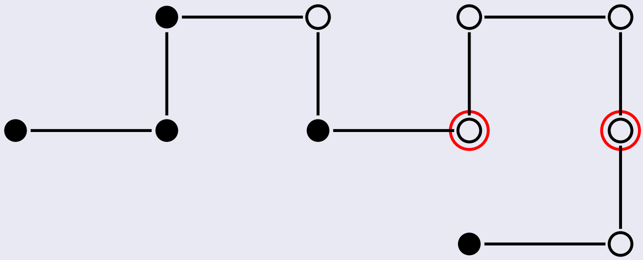
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



topological neighbors

The H-P model

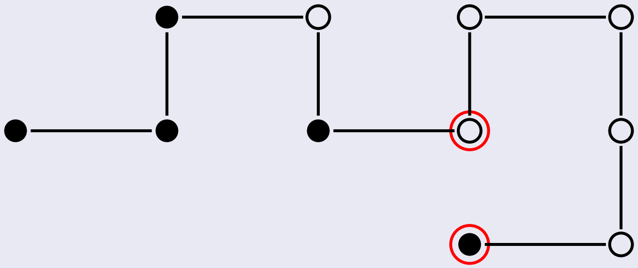
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



topological neighbors

The H-P model

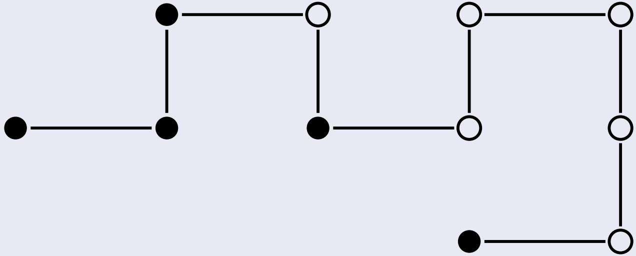
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



topological neighbors

The H-P model

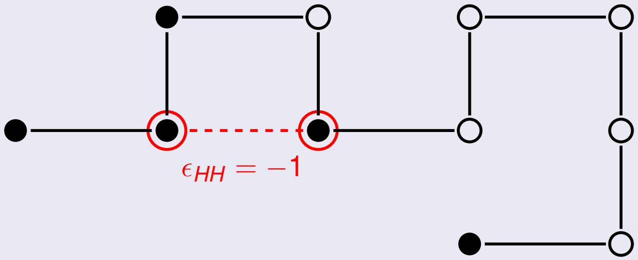
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy=

The H-P model

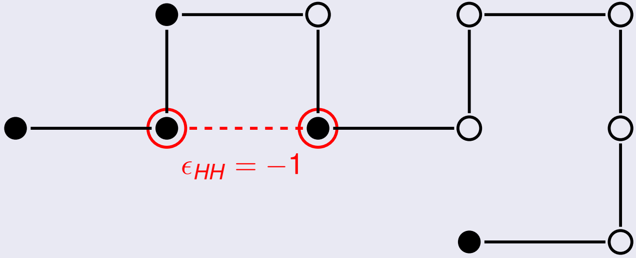
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy=

The H-P model

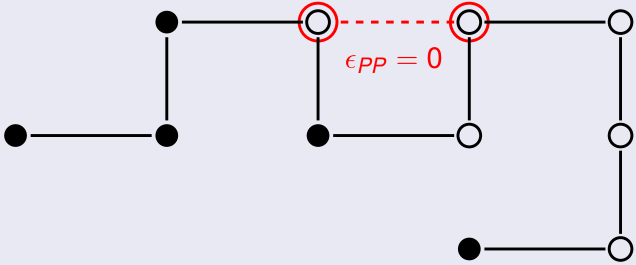
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -1

The H-P model

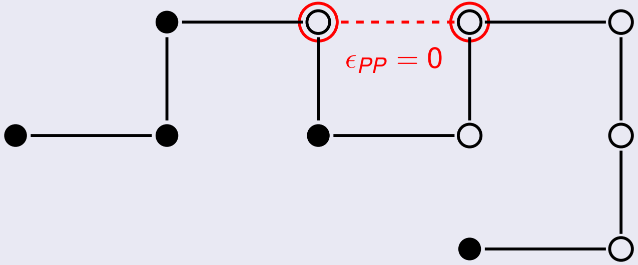
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -1

The H-P model

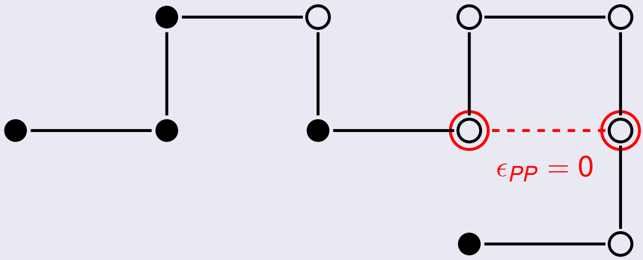
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -1+ 0

The H-P model

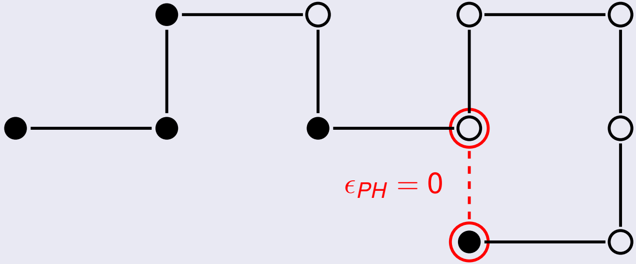
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -1+ 0 + 0

The H-P model

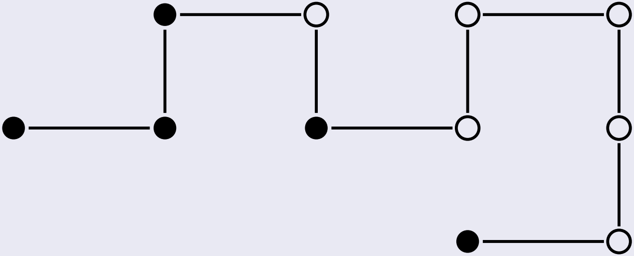
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -1+ 0 + 0 + 0

The H-P model

Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -1+ 0 + 0 + 0 = -1

The functional model

Characteristics

- The functional model is a “shifted” H-P model
- This model has native states, some of which are not maximally compact
- They have cavities or potential binding sites
- The main difference with the H-P model is the definition of the energy

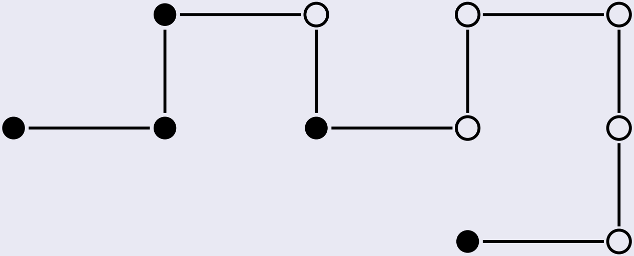
The functional model

Energy function

- A energy function that measures the interaction between topological neighbor residues is defined as:
  - $\epsilon_{HH} = -2$
  - $\epsilon_{HP} = 1$
  - $\epsilon_{PP} = 1$

The functional model

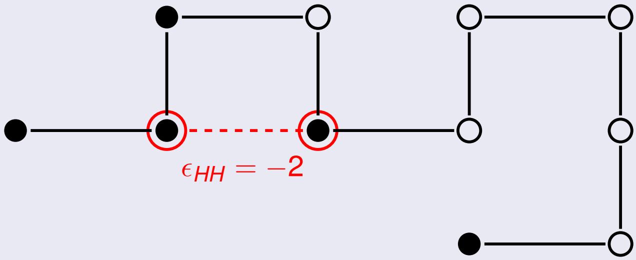
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy=

The functional model

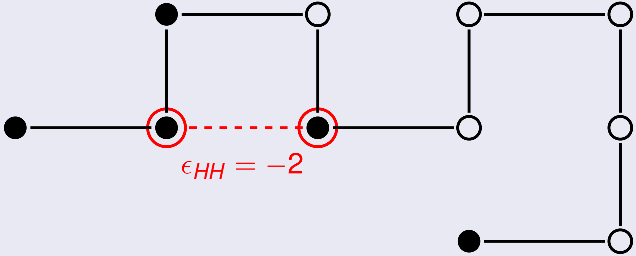
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy=

The functional model

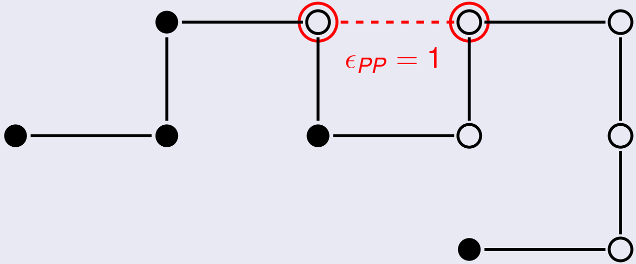
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -2

The functional model

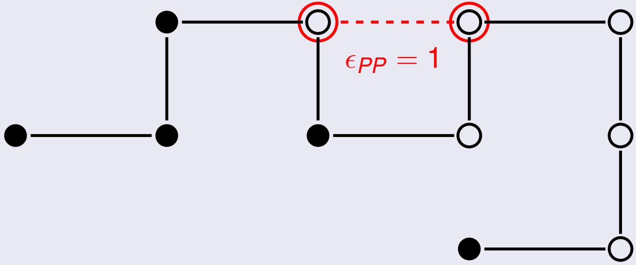
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -2

The functional model

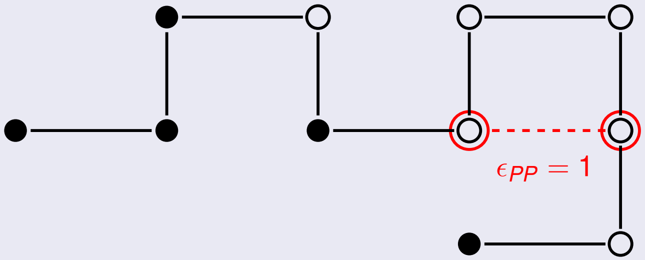
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -2+ 1

The functional model

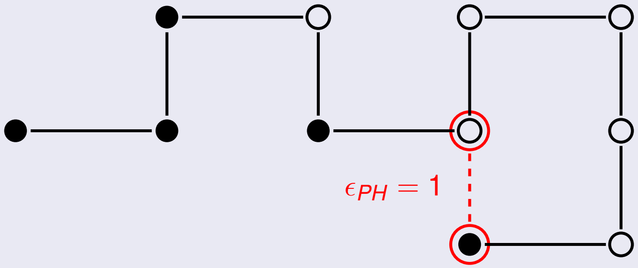
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -2+ 1 + 1

The functional model

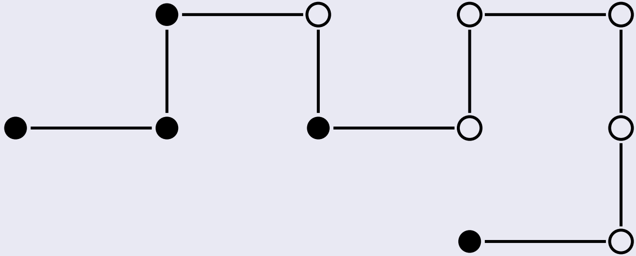
Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -2+ 1 + 1 + 1

The functional model

Example of folding of *HHHPHPPPPPH* protein in a 2-D lattice



Energy= -2+ 1 + 1 + 1 = 1

EDAs applied to the H-P problem . Problem formalization

Solution representation

- Given a chain of length n a solution is an array of length n
- Absolute codification
  - Each position X_i in the array will represent the global move of residue i
- Relative codification:
  - Each position X_i in the array will represent the relative move of residue i in relation to the previous two residues
  - In this way for a 2-D lattice $X_i \in \{0, 1, 2\}$ and for a 3-D lattice $X_i \in \{0, \ldots, 4\}$

EDAs applied to the H-P problem . Problem formalization

Solution representation

- Given a chain of length n a solution is an array of length n
- Absolute codification
  - Each position X_i in the array will represent the global move of residue i
- Relative codification:
  - Each position X_i in the array will represent the relative move of residue i in relation to the previous two residues
  - In this way for a 2-D lattice $X_i \in \{0, 1, 2\}$ and for a 3-D lattice $X_i \in \{0, \ldots, 4\}$

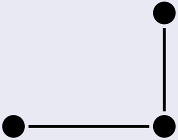
Problem formalization

Solution representation

$$(0, 0, -, -, -, -, -, -, -, -, -)$$

Problem formalization

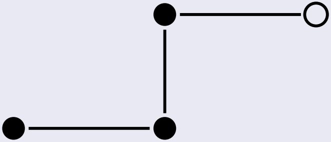
Solution representation



$$(0, 0, 1, -, -, -, -, -, -, -, -)$$

Problem formalization

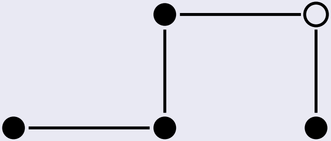
Solution representation



$$(0, 0, 1, -1, -, -, -, -, -, -, -)$$

Problem formalization

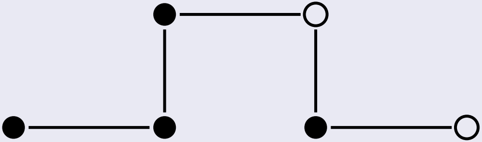
Solution representation



$$(0, 0, 1, -1, -1, -, -, -, -, -, -)$$

Problem formalization

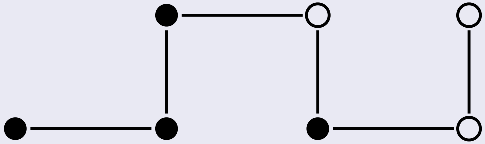
Solution representation



$$(0, 0, 1, -1, -1, 1, -, -, -, -, -)$$

Problem formalization

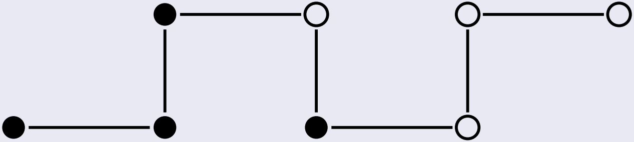
Solution representation



$$(0, 0, 1, -1, -1, 1, 1, -, -, -, -)$$

Problem formalization

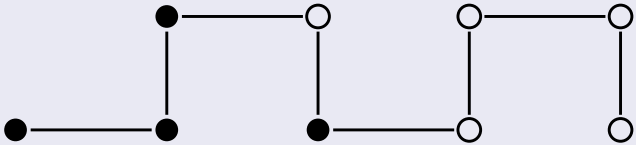
Solution representation



$$(0, 0, 1, -1, -1, 1, 1, -1, -, -, -)$$

Problem formalization

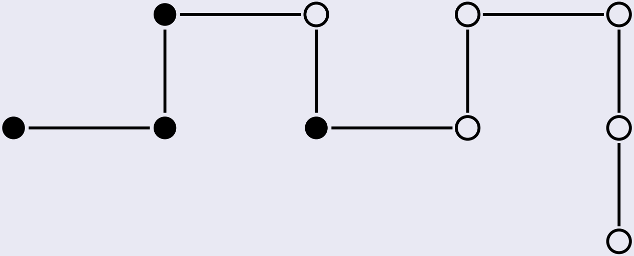
Solution representation



$$(0, 0, 1, -1, -1, 1, 1, -1, -1, -, -)$$

Problem formalization

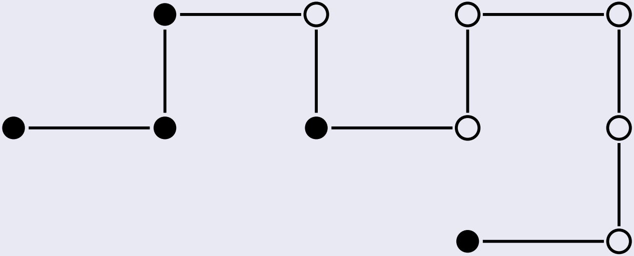
Solution representation



$$(0, 0, 1, -1, -1, 1, 1, -1, -1, 0, -)$$

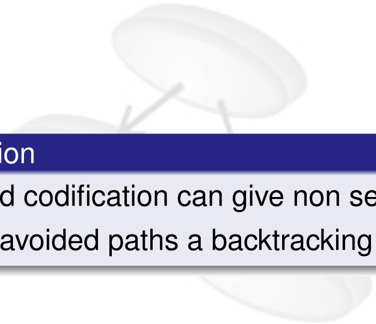
Problem formalization

Solution representation



$$(0, 0, 1, -1, -1, 1, 1, -1, -1, 0, -1)$$

Problem formalization



Relative codification

- The proposed codification can give non self-avoided paths
- To have self-avoided paths a backtracking method is used

Estimation of Distribution Algorithms

Pseudocode for an EDA

```
Obtain an initial population of individuals $D_0$
Repeat until a stopping criterion is met
    Select from $D_i$ a subset of individuals $D_i^S$
    Learn a probability distribution $p_i(\boldsymbol{x})$ from $D_i^S$
    Sample $p_i(\boldsymbol{x})$ to obtain $D_{i+1/2}$
    Create the new population $D_{i+1}$ from $D_i$ and $D_{i+1/2}$
```

UPV - EHU

Which probabilistic model to use?

Dependencies and regularities in the search space

- Our first idea is to use an independence model $p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$
- However, are there dependencies and regularities between the variables of the problem?
- How to look for dependencies and regularities between the variables of the problem?

Solution: Boltzmann Distribution

Boltzmann distribution

Definition

- Given a function $f(x)$ we define the Boltzmann distribution as:

$$p(x) = \frac{e^{\beta f(x)}}{\sum_y e^{\beta f(y)}}$$

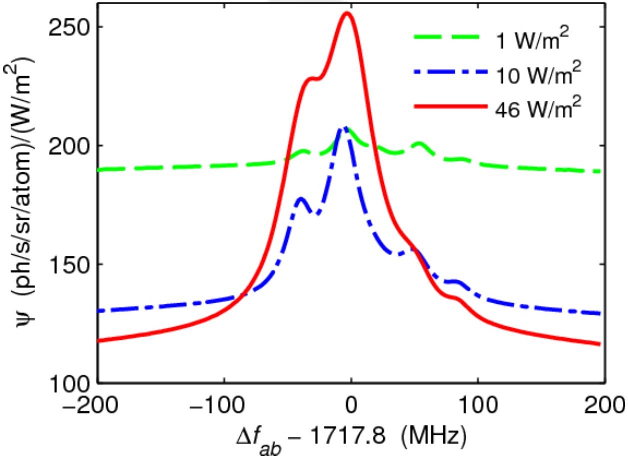
  where $\beta > 0$

Properties of the Boltzmann distribution

Properties

- Establish a relation between the function to optimize and the probability distribution
- Assign probabilities such that if $f(x) > f(y)$ then $p(x) > p(y)$
- When $\beta \to \infty$, only optimal points have positive probabilities
- If we could efficiently sample the Boltzmann distribution the optimization problem would be solved

Boltzmann distribution



Relation between ADFs and the Boltzmann distribution

Additive decomposable functions

$$f(\mathbf{x}) = f_1(\mathbf{x}_{s_1}) + f_2(\mathbf{x}_{s_2}) + \ldots + f_l(\mathbf{x}_{s_l})$$

Theorem

Given an ADF function $f(\mathbf{x}) = \sum_{i=1}^{l} f_i(\mathbf{x}_{s_i})$, if X_j y X_k are not in the argument of any $f_i(\mathbf{x}_{s_i})$ of $f(\mathbf{x})$ then X_j y X_k are conditionally independent given the rest of the variables for the Boltzmann distribution $p(\mathbf{x})$ associated to $f(\mathbf{x})$.

Which probabilistic model to use?

Dependencies and regularities in the search space

- Our first idea is to use an independence model $p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i)$
- However, are there dependencies and regularities between the variables of the problem?
- How to look for dependencies and regularities between the variables of the problem?

Solution: Boltzmann Distribution

Regularities in the search space

Experimental set-up

- Protein used: HHHPHPPPPPH
- We have calculated the whole search space

$H(\mathbf{x})$	-4	-3	-2	-1	0	1	2	3	4	5	invalid	total
HP_f	0	0	16	1428	9581	0	0	0	0	0	8658	19683

- Build the Boltzman distribution associated with the objective function of the protein:

$$p(\mathbf{x}) = \frac{e^{\frac{-H(\mathbf{x})}{t}}}{\sum_{\mathbf{x}'} e^{\frac{-H(\mathbf{x}')}{t}}}$$

- Calculated the marginal probability of (X_3, X_4, X_5) and the marginal of each variable separately

Marginal probabilities

	$p_{HP}(x_3, x_4, x_5)$		
Seq	0 – –	1 – –	2 – –
–00	**0,000**	0,057	0,069
–01	0,035	0,036	0,038
–02	0,035	0,036	0,035
–10	0,029	0,034	0,035
–11	0,038	0,040	0,038
–12	0,035	0,034	0,029
–20	0,035	0,036	0,035
–21	0,038	0,036	0,035
–22	0,069	0,057	**0,000**

	$p_{HP}(x_3) \cdot p_{HP}(x_4) \cdot p_{HP}(x_5)$		
Seq	0 – –	1 – –	2 – –
–00	0,036	0,042	0,036
–01	0,036	0,042	0,036
–02	0,036	0,042	0,036
–10	**0,033**	0,038	**0,033**
–11	0,033	0,039	0,033
–12	**0,033**	0,038	**0,033**
–20	0,036	0,042	0,036
–21	0,036	0,042	0,036
–22	0,036	0,042	0,036

Marginal probabilities

	$p_{HP}(x_3, x_4, x_5)$		
Seq	0 − −	1 − −	2 − −
−00	**0,000**	0,057	0,069
−01	0,035	0,036	0,038
−02	0,035	0,036	0,035
−10	0,029	0,034	0,035
−11	0,038	0,040	0,038
−12	0,035	0,034	0,029
−20	0,035	0,036	0,035
−21	0,038	0,036	0,035
−22	0,069	0,057	**0,000**

	$p_{HP}(x_3) \cdot p_{HP}(x_4) \cdot p_{HP}(x_5)$		
Seq	0 − −	1 − −	2 − −
−00	0,036	0,042	0,036
−01	0,036	0,042	0,036
−02	0,036	0,042	0,036
−10	**0,033**	0,038	**0,033**
−11	0,033	0,039	0,033
−12	**0,033**	0,038	**0,033**
−20	0,036	0,042	0,036
−21	0,036	0,042	0,036
−22	0,036	0,042	0,036

IMPORTANT REGULARITIES IN THE FUNCTION

EDAs for the H-P model

Probabilistic models used

- Markov model (only parameter learning required):

$$p_{MK}(\mathbf{x}) = p(x_1, \ldots, x_{k+1}) \prod_{i=k+2}^{n} p(x_i \mid x_{i-1}, \ldots, x_{i-k})$$

- Tree model (structure and parameter learning):

$$p_{Tree}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i \mid pa(x_i))$$

EDAs for the H-P model

Probabilistic models used

- Mixture of trees (structure and parameter learning):

$$p_{MT}(\mathbf{x}) = \sum_{j=1}^{m} \lambda_j p^j_{Tree}(\mathbf{x})$$

con $\lambda_j > 0$, $j = 1, \ldots, m$, $\sum_{j=1}^{m} \lambda_j = 1$

UPV - EHU

EDAs for the H-P model

Experimental results in 2-d

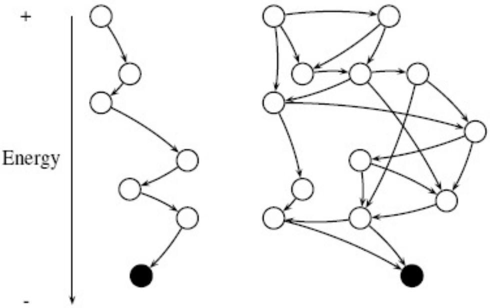
	BestEDA	*GA*	*MMA*	*ACO*	*NewACO*	*PERM*
inst.	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$	$H(\mathbf{x})$
$s1(20)$	-9	-9	-9	-9	-9	-9
$s2(24)$	-9	-9	-9	-9	-9	-9
$s3(25)$	-8	-8	-8	-8	-8	-8
$s4(36)$	-14	-14	-14	-14	-14	-14
$s5(48)$	-23	-22	-22	-23	-23	-23
$s6(50)$	-21	-21		-21	-21	-21
$s7(60)$	-35	-34		-34	-36	-36
$s8(64)$	-42	-37		-32	-42	-38
$s9(85)$	-52				-51	-53
$s10(100)$	-47				-47	-50
$s11(100)$	-48				-47	-48

EDAs for the H-P model

Experimental results in 3-d

	hybrid GA		MK-EDA$_2$		TreeEDA		MT-EDA$_4$	
	$H(\mathbf{x})$	*mean*	$H(\mathbf{x})$	*mean*	$H(\mathbf{x})$	*mean*	$H(\mathbf{x})$	*mean*
$s1$	−11	−10,5	−11	−10,8	−11	−10,6	−11	−10,8
$s2$	−13	−11,2	−13	−12,0	−13	−11,3	−13	−11,8
$s3$	−9	−8,5	−9	−8,9	−9	−8,9	−9	−9,0
$s4$	−18	−15,7	−18	−16,4	−18	−16,2	−18	−16,5
$s5$	−28	−24,6	−29	−27,2	−29	−26,8	−29	−27,0
$s6$	−26	−23,0	−29	−25,7	−31	−25,9	−28	−25,7
$s7$	−49	−41,1	−49	−46,3	−49	−43,7	−48	−42,0
$s8$	−46	−40,4	−52	−46,7	−49	−43,7	−50	−45,6

EDAs as a model for protein folding



- Classical and new approaches to protein folding
- We postulate that EDAs can be considered as a model of protein folding in the new view

EDAs as a model of protein folding

General considerations

- EDAs model: At each step t the probability of being in a particular folded structure is given by the probabilistic model
- Similarities between both models:
  - Both models, define a sampling of the space of configurations
  - The probability of sampling a configuration depends on the quality of its energy function
  - Both entities tend to preserve local favorable conformational features through successive generations (time steps)

EDAs as a model of protein folding

General considerations

- EDAs model: At each step *t* the probability of being in a particular folded structure is given by the probabilistic model
- Similarities between both models:
  - Both models, define a sampling of the space of configurations
  - The probability of sampling a configuration depends on the quality of its energy function
  - Both entities tend to preserve local favorable conformational features through successive generations (time steps)

UPV - EHU

EDAs as a model of protein folding

General considerations

- EDAs model: At each step *t* the probability of being in a particular folded structure is given by the probabilistic model
- Similarities between both models:
  - Both models, define a sampling of the space of configurations
  - The probability of sampling a configuration depends on the quality of its energy function
  - Both entities tend to preserve local favorable conformational features through successive generations (time steps)

UPV - EHU

EDAs as a model of protein folding

General considerations

- EDAs model: At each step *t* the probability of being in a particular folded structure is given by the probabilistic model
- Similarities between both models:
  - Both models, define a sampling of the space of configurations
  - The probability of sampling a configuration depends on the quality of its energy function
  - Both entities tend to preserve local favorable conformational features through successive generations (time steps)

EDAs as a model of protein folding

General considerations

- EDAs model: At each step *t* the probability of being in a particular folded structure is given by the probabilistic model
- Similarities between both models:
    - Both models, define a sampling of the space of configurations
    - The probability of sampling a configuration depends on the quality of its energy function
    - Both entities tend to preserve local favorable conformational features through successive generations (time steps)

EDAs as a model of protein folding

General considerations

- EDAs model: At each step *t* the probability of being in a particular folded structure is given by the probabilistic model
- Similarities between both models:
    - Both models, define a sampling of the space of configurations
    - The probability of sampling a configuration depends on the quality of its energy function
    - Both entities tend to preserve local favorable conformational features through successive generations (time steps)

EDAs as a model of protein folding

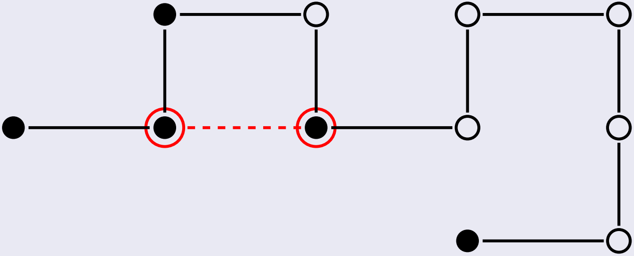
New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
    - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
  - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

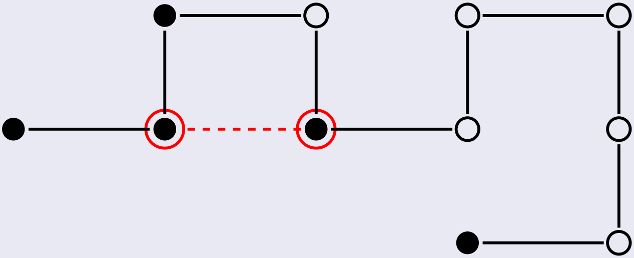


Contact Order=

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
  - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

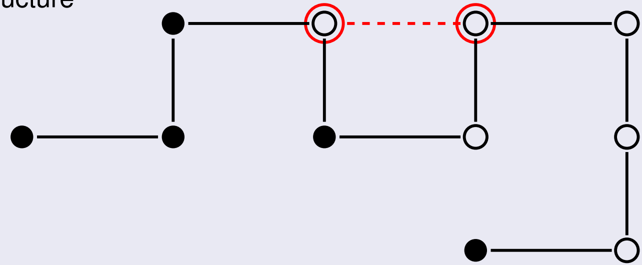


Contact Order= 3

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
    - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

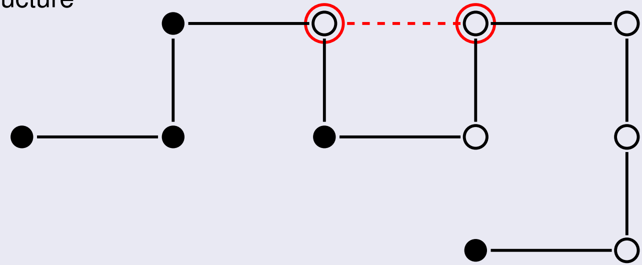


Contact Order= 3

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
  - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

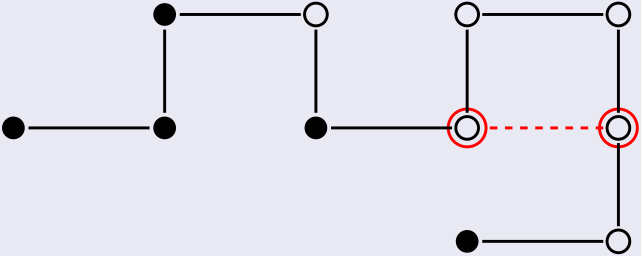


Contact Order= 3+ 3

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
    - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

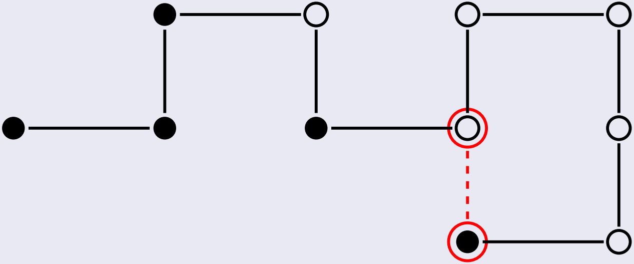


Contact Order= 3+ 3 + 3

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
  - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure

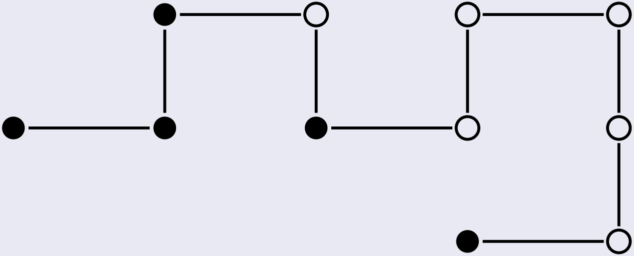


Contact Order= 3+ 3 + 3 + 5

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
  - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure



Contact Order= $\frac{3+3+3+5}{4}$ = 3.5

EDAs as a model of protein folding

New view of protein folding

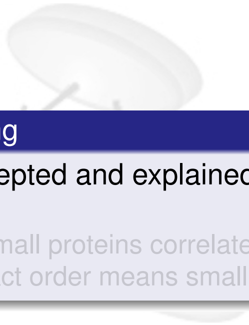
- Several parameters are used to describe a protein and the folding degree:
    - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure
    - Folding rate: Amount of time the protein takes to fold

EDAs as a model of protein folding

New view of protein folding

- Several parameters are used to describe a protein and the folding degree:
    - Contact order: Average sequence separation between residues that make contact in the native three-dimensional structure
    - Folding rate: Amount of time the protein takes to fold
    - Fraction of native contacts (in the current folding)

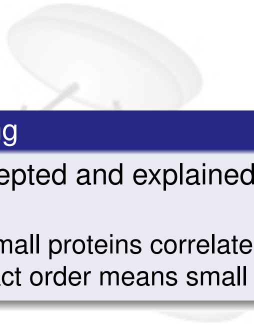
EDAs as a model of protein folding



New view of protein folding

- Facts commonly accepted and explained in the new view of protein folding:
    - Folding rates of small proteins correlate with their contact order: small contact order means small folding rate

EDAs as a model of protein folding



New view of protein folding

- Facts commonly accepted and explained in the new view of protein folding:
    - Folding rates of small proteins correlate with their contact order: small contact order means small folding rate

EDAs as a model of protein folding

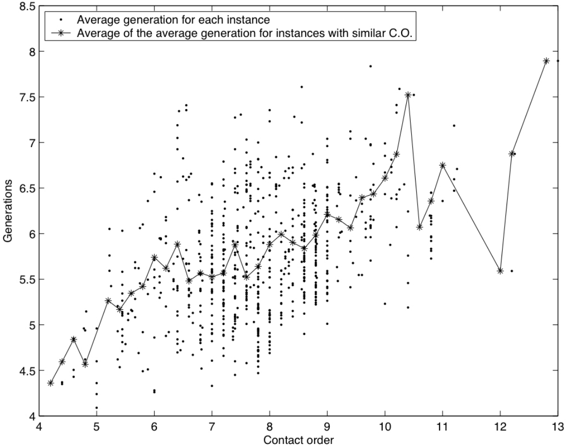
Experiments

- The EDA used is MK-EDA$_2$ with probabilistic model:

$$p_{MK}(\mathbf{x}) = p(x_1, x_2, x_3) \prod_{i=4}^{n} p(x_i \mid x_{i-1}, x_{i-2})$$

- Experiments are carried out with 15545 instances of the functional model
- 100 experiments are performed with each sequence

EDAs as a model of protein folding



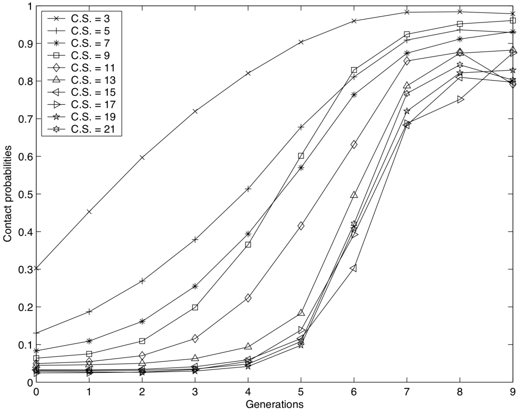
UPV - EHU

EDAs as a model of protein folding

New View of Protein Folding

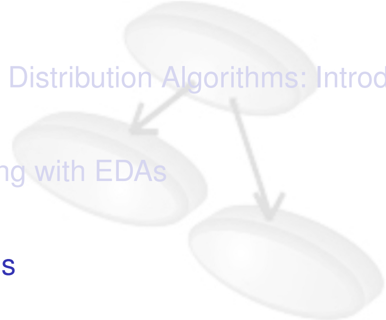
- Facts commonly accepted and explained in the new view of protein folding:
  - Folding rates of small proteins correlate with their contact order: small contact order means small folding rate
  - Local interactions are more likely to form early in folding than non-local interactions

EDAs as a model of protein folding



UPV - EHU

Outline of the presentation



1. Estimation of Distribution Algorithms: Introduction
2. Protein Folding with EDAs
3. Miscellaneous
4. Conclusions

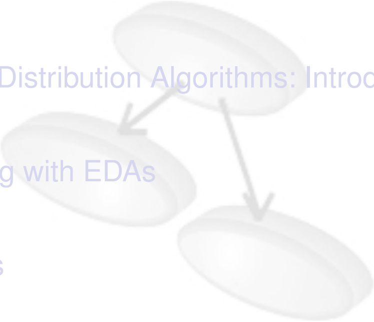
Parallel EDAs

- The structure of EDAs are very suitable for parallelization
- Two main approaches:
  - Parallelization maintaining the whole behavior of the algorithm
  - Island models: migration of models, different probabilistic models at each island

More stuff

- EDAs in multimodal problems: clustering methods, probabilistic model for clustering
- EDAs in multiobjective problems
- Memetic EDAs
- Incorporation of a priori information of the function in EDAs without fixed probabilistic model

Outline of the presentation



1. Estimation of Distribution Algorithms: Introduction
2. Protein Folding with EDAs
3. Miscellaneous
4. Conclusions

Conclusions

- EDAs are an attractive field of research in Evolutionary Computation
- EDAs are encouraging cross-fertilization with other fields
- They are founded in mathematical basis
- Many, many things to be done

From a practical point of view

- They offer competitive alternatives for many problems
- The possibility of incorporating knowledge of the function is the key point in applications
- In the case of ADF: use algorithms with fixed probabilistic model
- In the case of no knowledge about the function: use EDAs without fixed structure