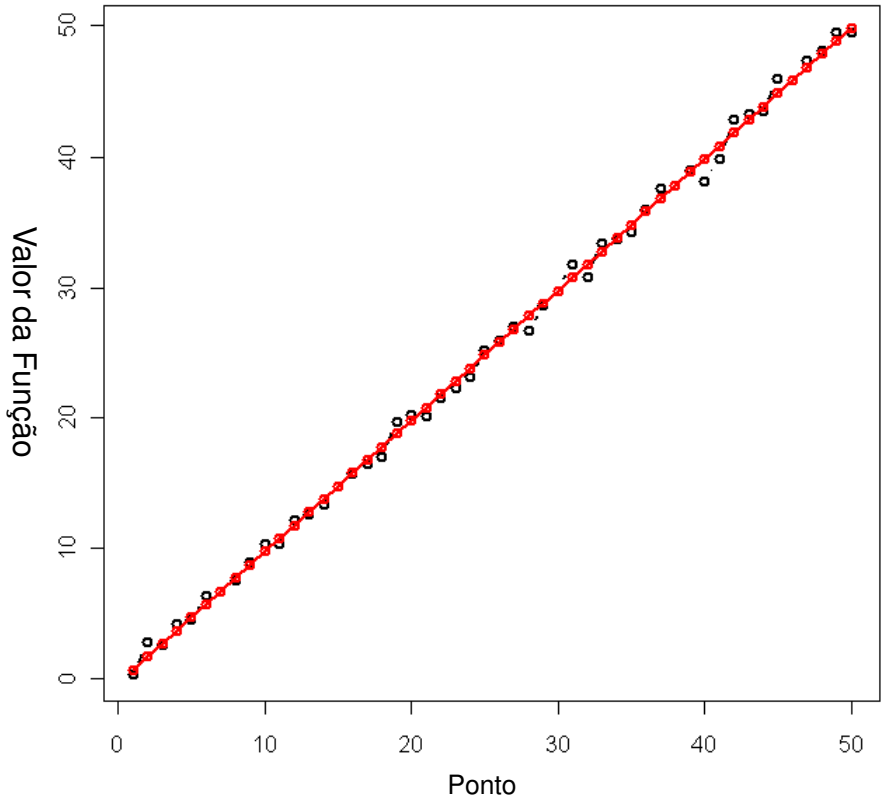
O PROBLEMA

- Exemplo de funcao quase-linear
  - Ajustar modelo linear
- Exemplo de funçao deceptiva de uma armadilha
  - Ajustar modelo linear
- Exemplo de funçao deceptiva de 3 armadilhas
  - Ajustar modelo linear

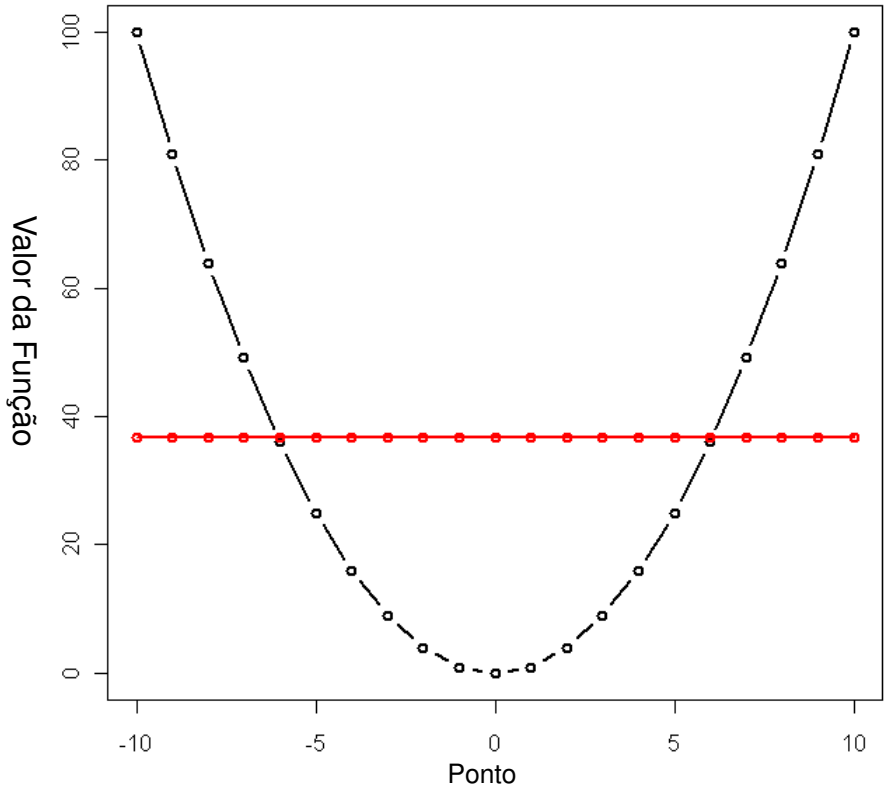
MODELAGEM PARA RESOLUÇÃO DE PROBLEMAS

FUNÇÃO QUASE LINEAR



$$Y = 0.09639 + 0.99222*x$$

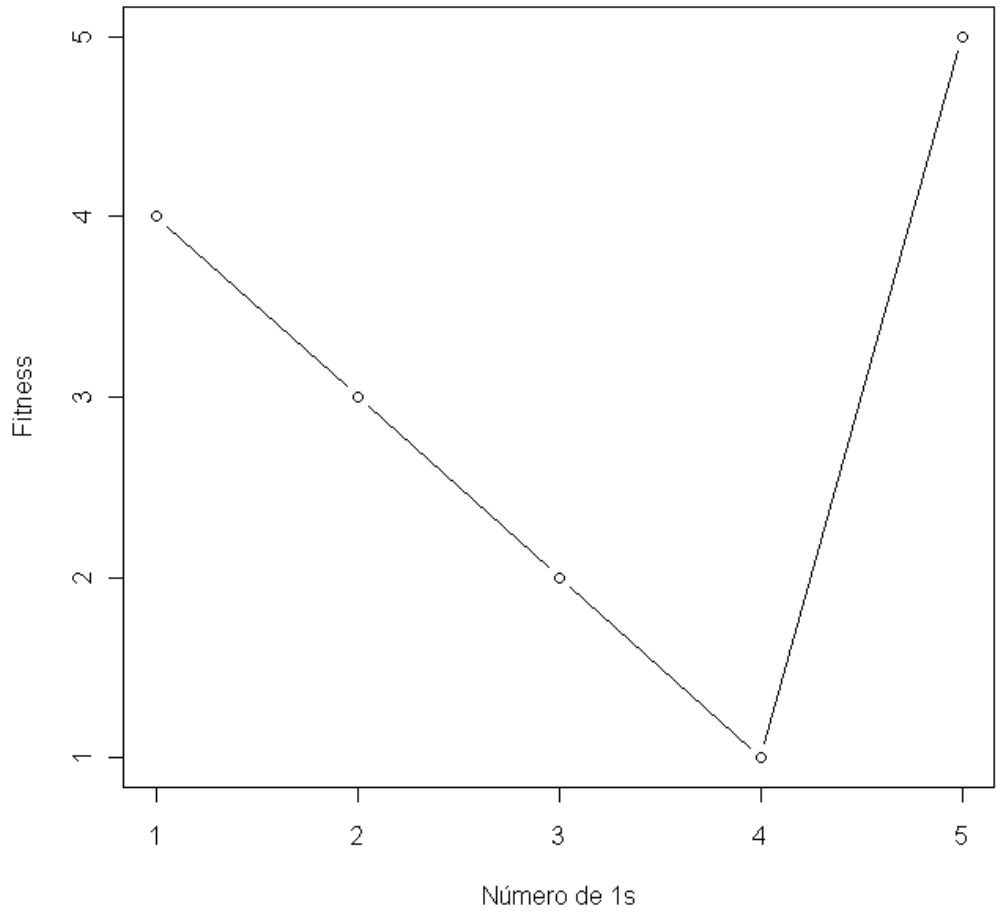
FUNÇÃO PARÁBOLA



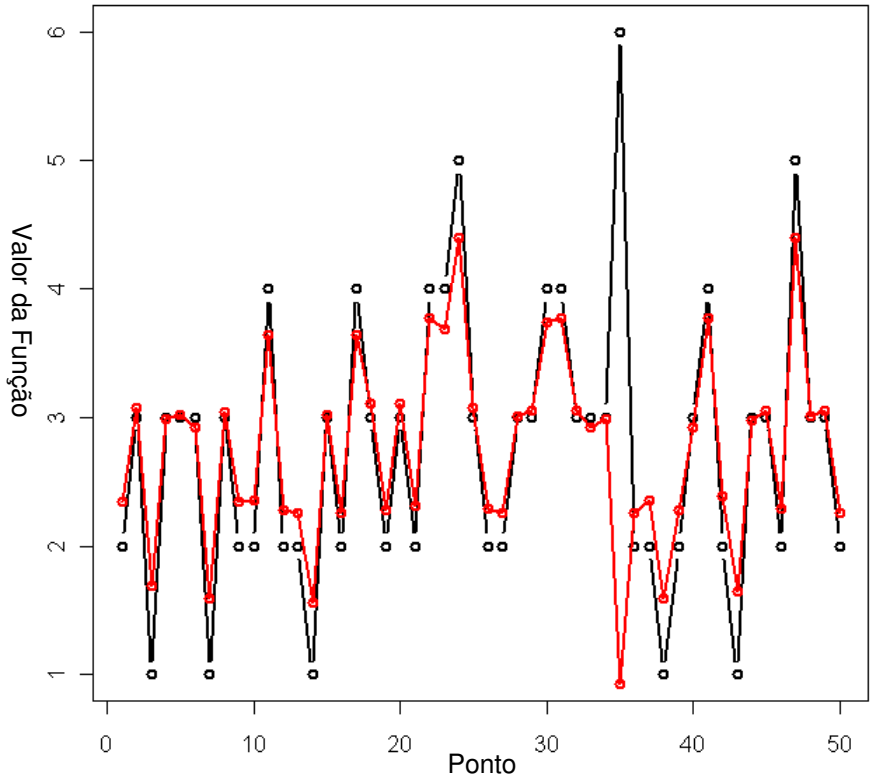
Y = 3.667e+01 - 4.616e-16*x

FUNÇÃO ARMADILHA (TRAP-5)
STRING BINÁRIA (S)

$$f(s) = \begin{cases} sum(s), & if\ sum(s) > 0; \\ l+1, & if\ sum(s) = 0. \end{cases}$$



1 ARMADILHA



Y = 2.0039 + 0.2486*x1 + 0.7163*x2 + 0.2061*x3 + 0.1920*x4 + 0.3965*x5 + 0.000*x6

BLOCOS DE CONSTRUÇÃO "BUILDING BLOCKS"

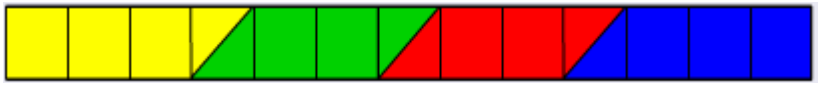
- Blocos construtivos são subconjuntos de bits que supostamente conferem altos valores de aptidão aos cromossomos em que aparecem. Diz-se que indivíduos bem avaliados são formados pela justaposição de blocos supostamente bem avaliados

BLOCOS EM FUNÇÕES ADITIVAS (CONCATENADAS)

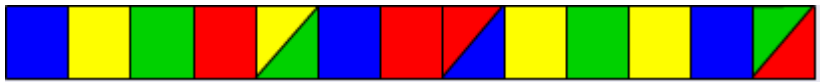
- Cada problema contém *k* bits
- Problemas separáveis contém subproblemas não sobrepostos



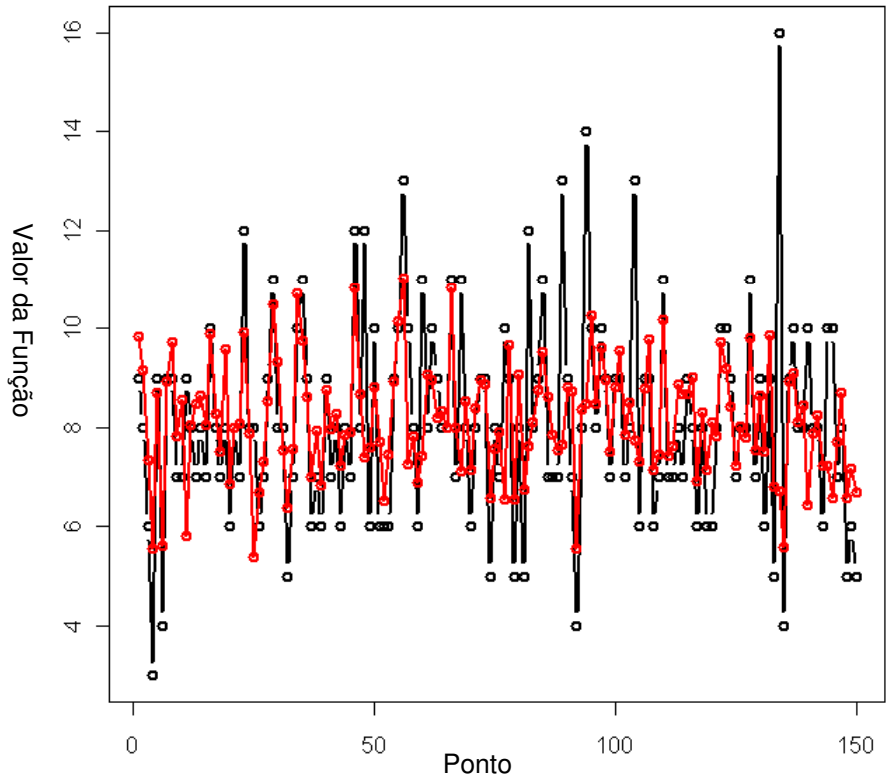
- Pode haver sobreposição em *O* bits entre subproblemas vizinhos com subproblemas definidos em blocos contíguos



- Os bits podem estar embaralhados para dificultar o problema



3 ARMADILHAS



$$Y = 3.574 + 0.752*x1 + 0.63*x2 + 1.06*x3 + 0.654*x4 + 0.733*x5 + 0.8*x6 + 0.408*x7 + 0.875*x8 + 0.62*x9 + 0.974*x10 + 0.374*x11 + 0.039*x12 + 0.669*x13 + 0.31*x14 + 0.186*x15$$

ENCONTRAR OS "BUILDING BLOCKS"

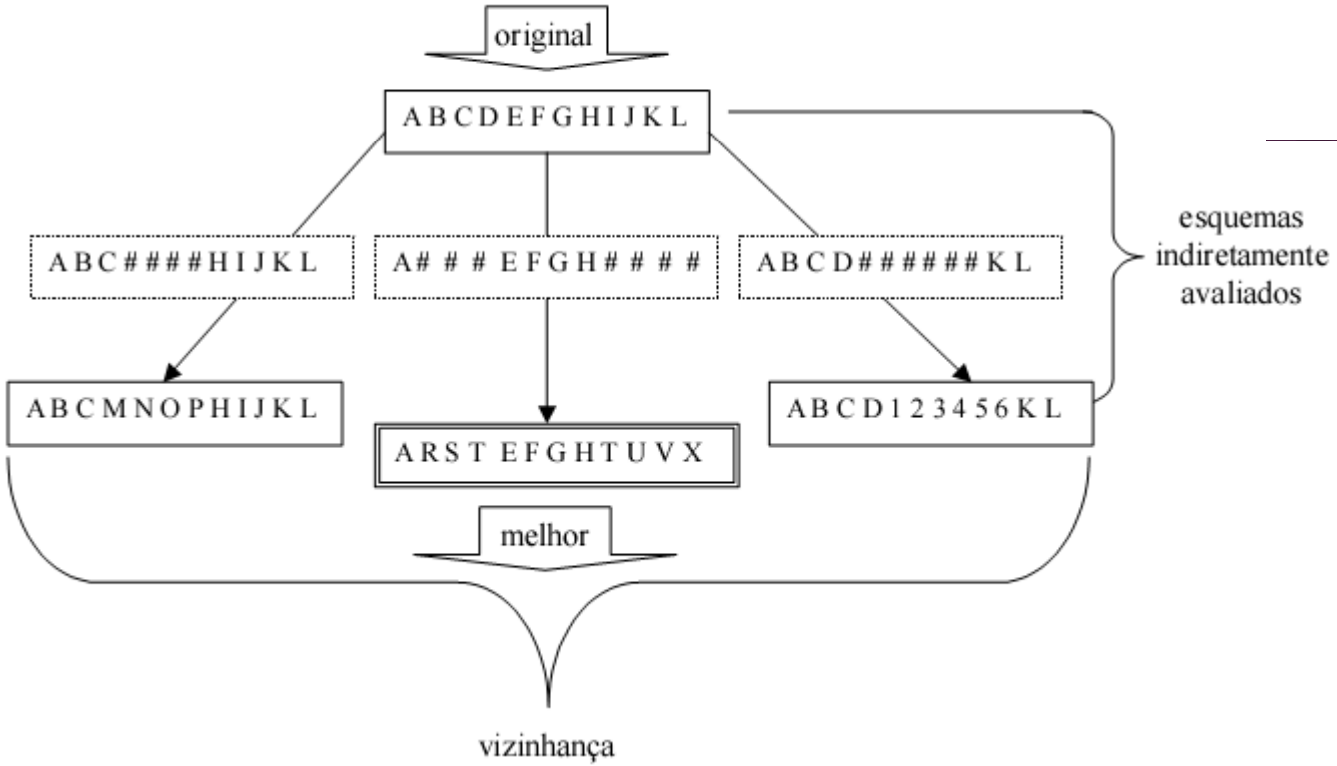
- Operadores clássicos de recombinação quebram os BBs
- Para resolver este tipo de problema, o algoritmo deve ser capaz de encontrar as relações/interações entre as variáveis
  - Teoria dos esquemas
  - Planejamento de Experimentos
  - Redes
  - Clustering

ENCONTRAR OS "BUILDING BLOCKS"

- Teoria dos esquemas
  - Tipo especial de indivíduo constituído de alguns genes rotulados como # (ou *).
  - Genes assim rotulados podem assumir quaisquer outros valores permitidos pelo alfabeto em uso.
  - Genes fixos, que se repetem com alta freqüência na população promissora, podem ser BBs

ENCONTRAR OS "BUILDING BLOCKS"

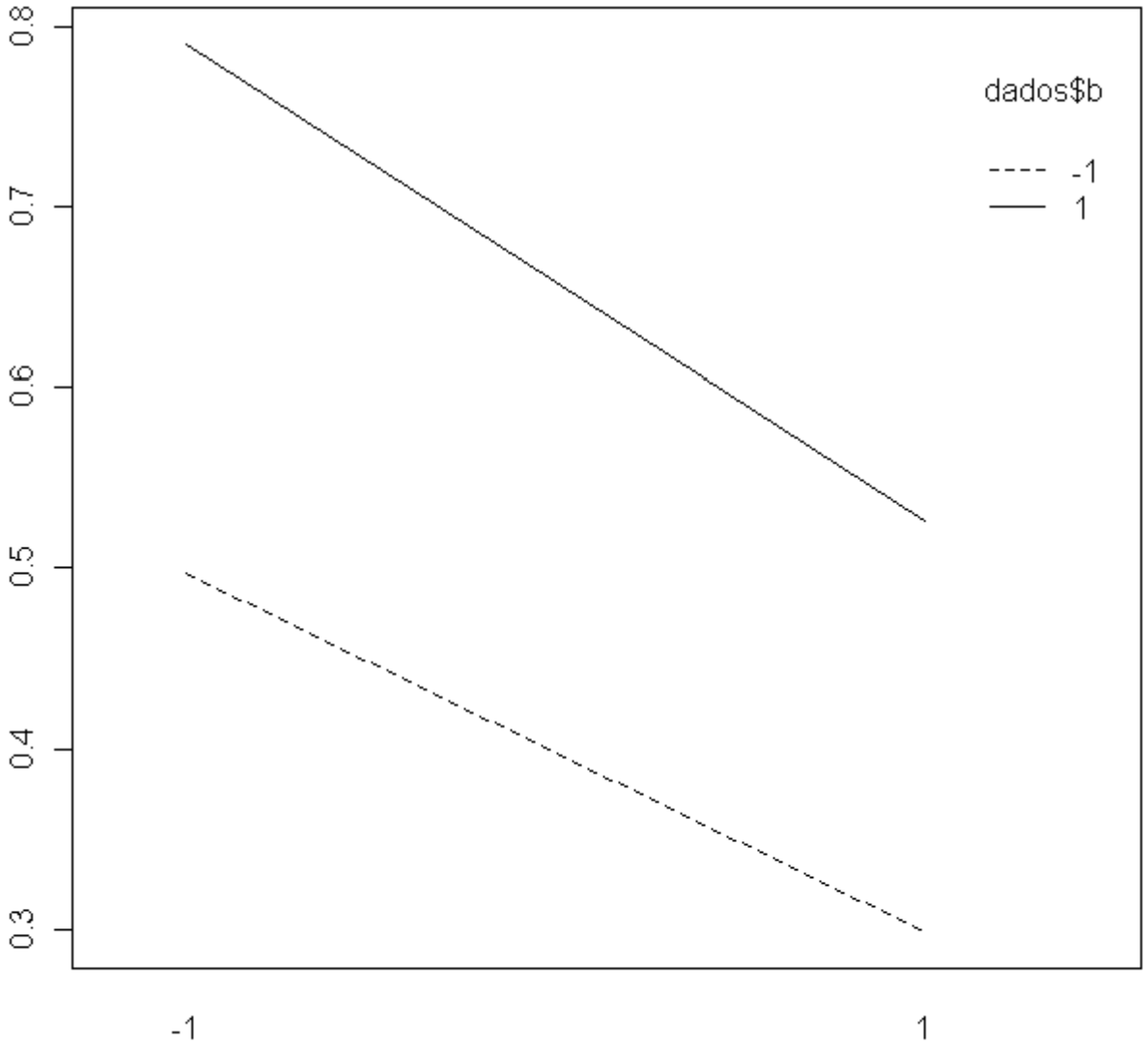
01##110##1#00



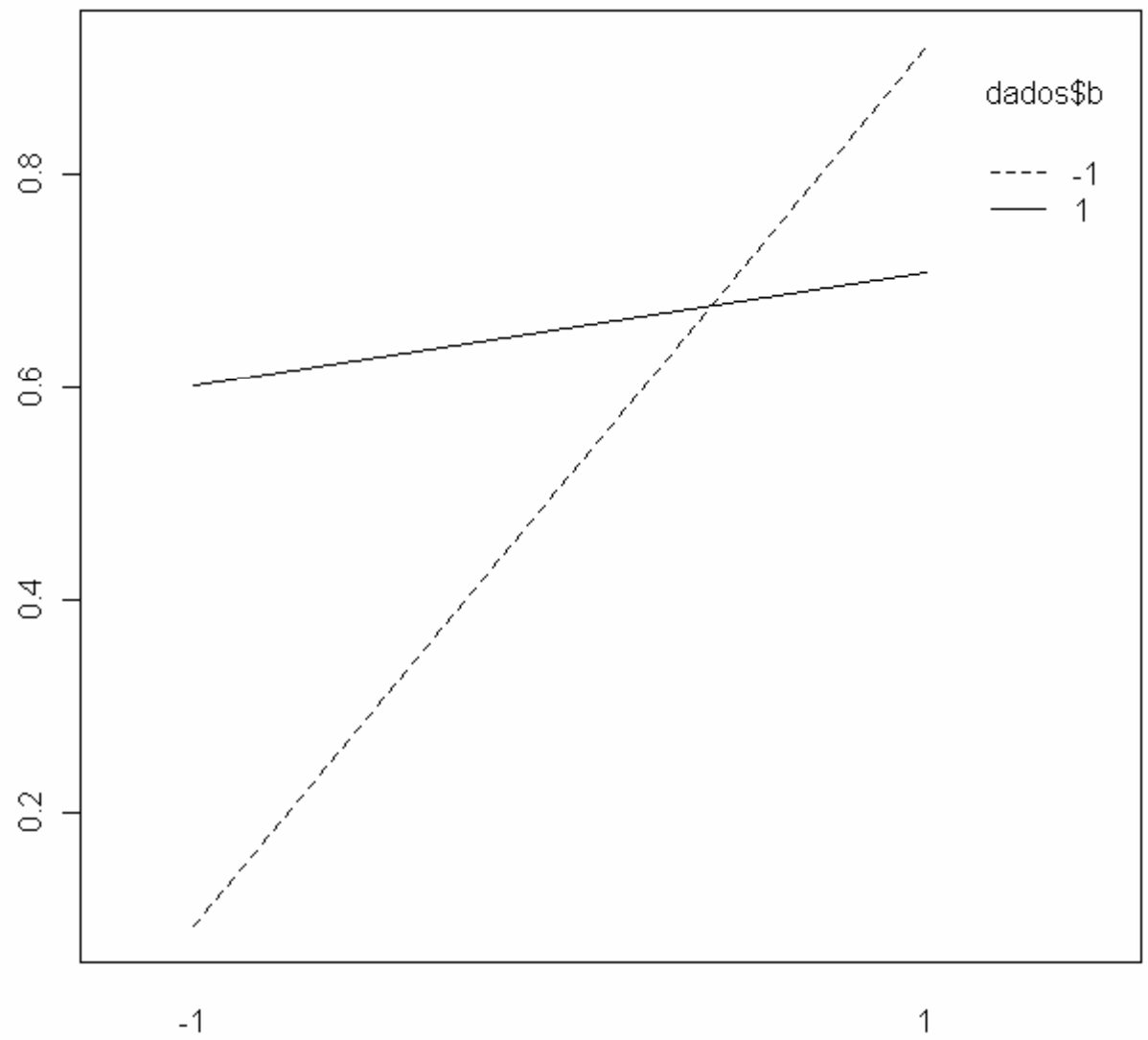
ENCONTRAR OS "BUILDING BLOCKS"

- Planejamento de Experimentos
  - Usar ferramentas estatísticas para detectar interação entre variáveis

SEM INTERAÇÃO

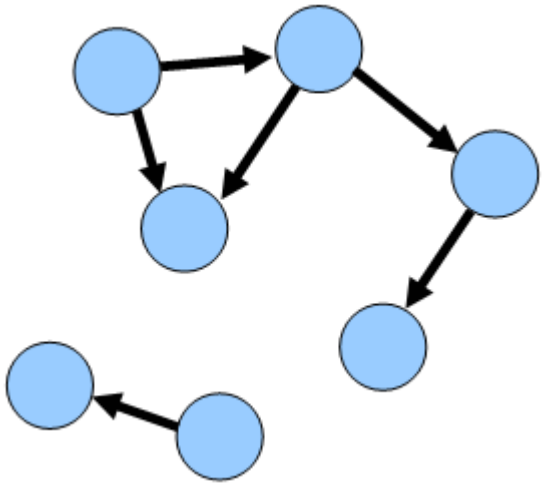


COM INTERAÇÃO



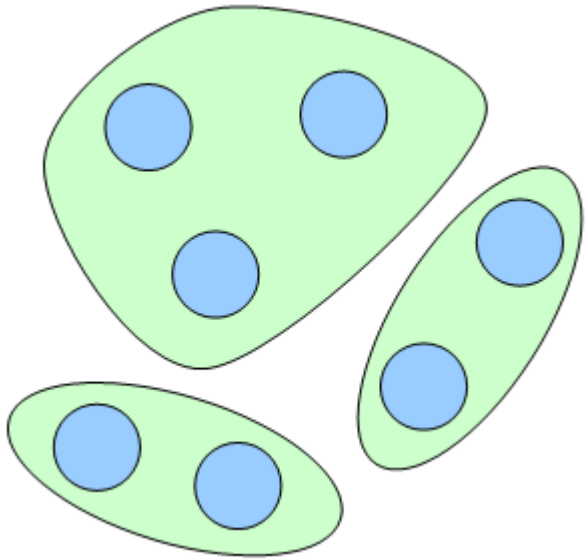
ENCONTRAR OS "BUILDING BLOCKS"

- Redes
  - Montar grafos que "representam" o comportamento (distribuição) da amostra promissora

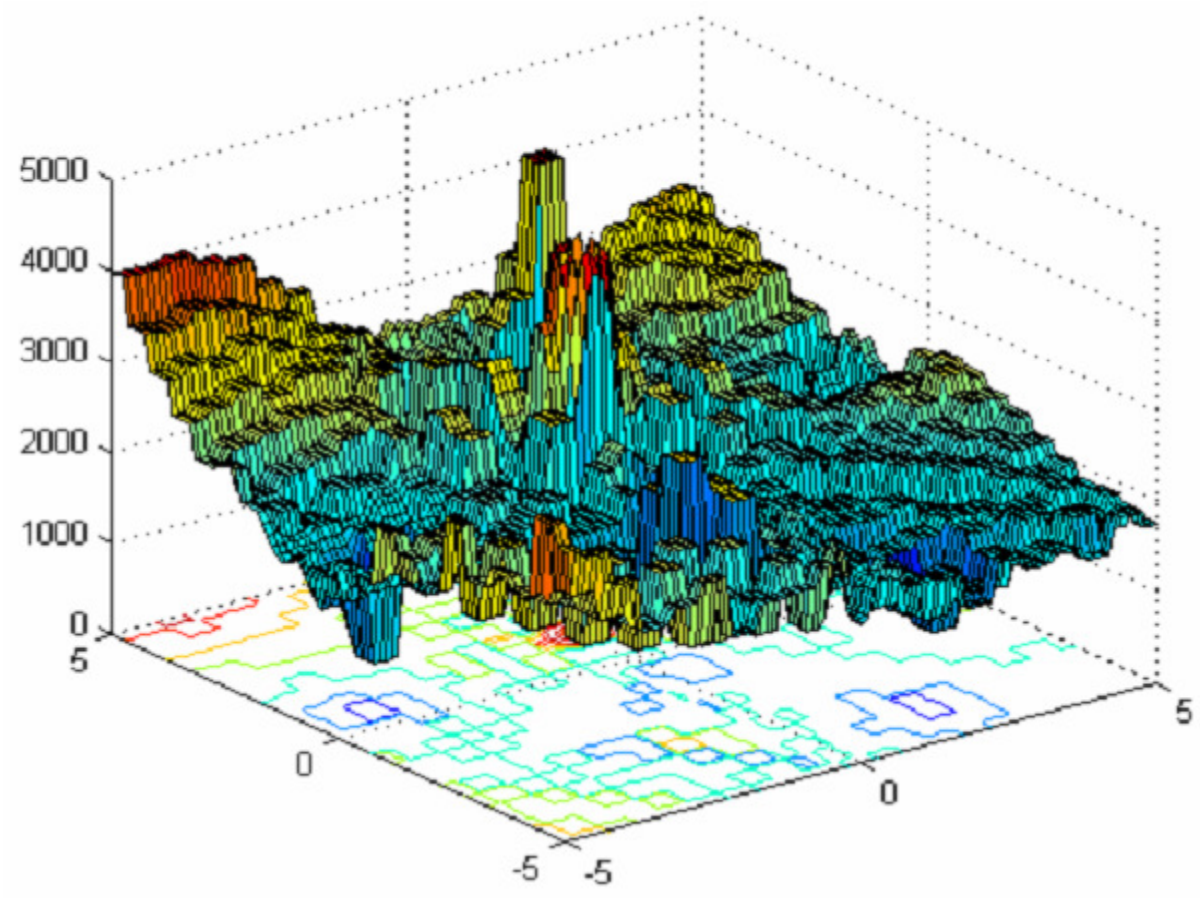


ENCONTRAR OS "BUILDING BLOCKS"

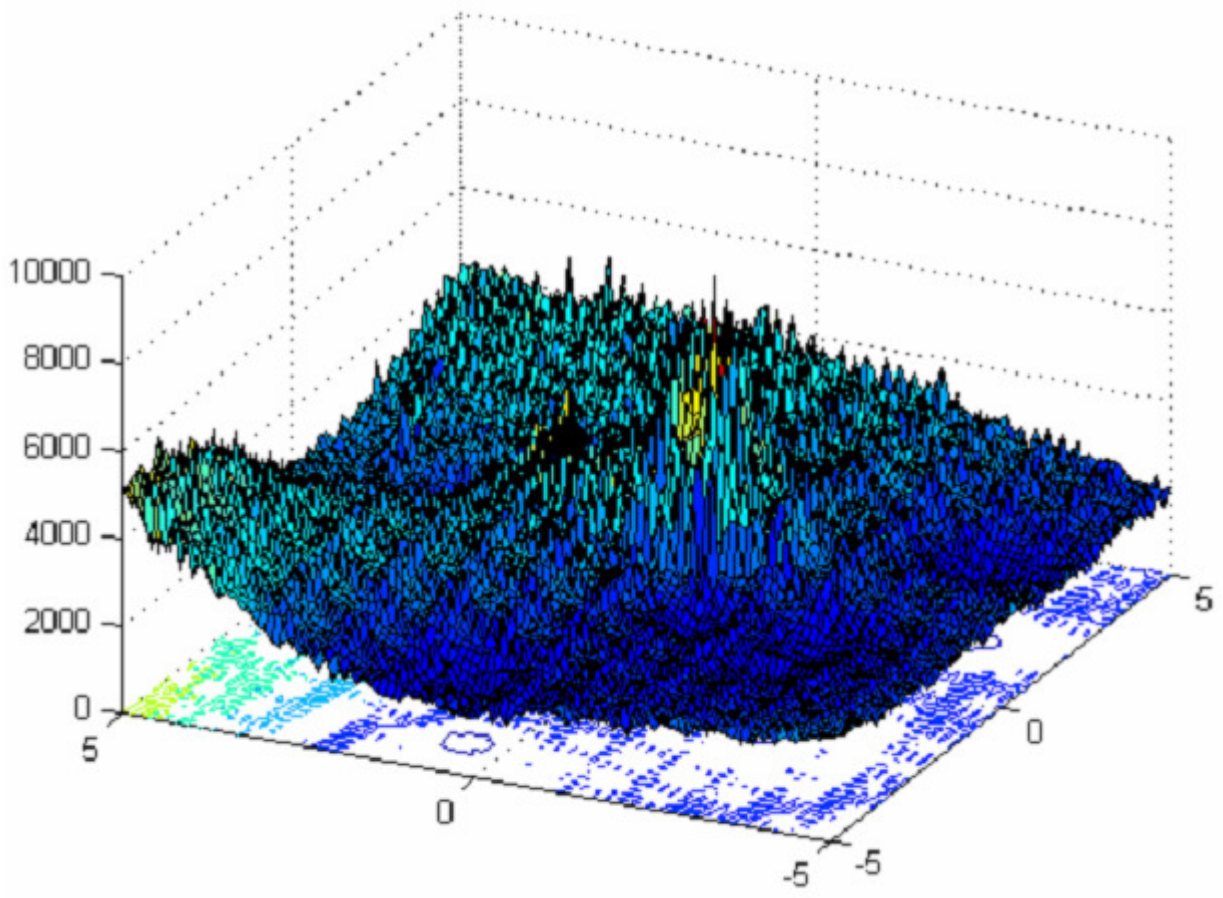
- Clustering
  - Agrupar bits de modo que a compactação da população promissora minimize uma determinada métrica



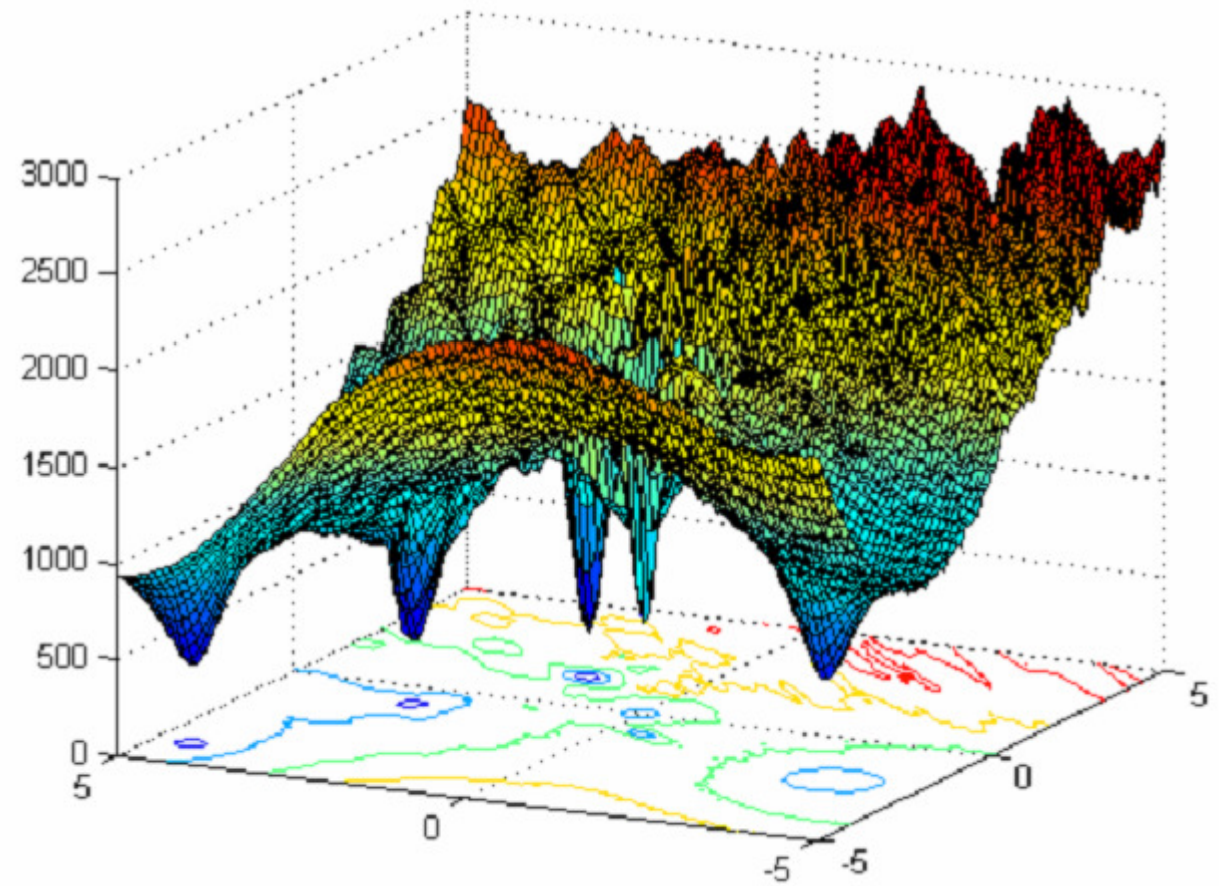
SUPERFÍCIES BIZARRAS



SUPERFÍCIES BIZARRAS



SUPERFÍCIES BIZARRAS



CAMINHO USUAL: EVOLUIR OPERADORES

- Alterar
  - a representação das soluções
  - operadores de recombinação
- Para que BBs fiquem menos propícios de serem quebrados
- Desvantagens
  - Várias operações de mapeamento e ordenação
  - Computacionalmente caro
  - Redução de quebras não é expressiva
  - Pode causar convergência prematura

BOA

- Bayesian optimization algorithm – [Pelikan etal., 1999]
- Usado em problemas decomponíveis
  - Existe uma estrutura de interação entre as variáveis
  - O algoritmo é capaz de identificar os subproblemas envolvidos
- Resolve (tenta)
  - Problema de quebra dos blocos de construção (crossover)
- Soluções
  - Mudar a representação das soluções
  - Evoluir operadores de recombinação
  - Extrair informação do conjunto de soluções promissoras para gerar novas soluções
    - Modelagem probabilística

MODELAGEM PROBABILÍSTICA

- Algoritmos de Estimação de Distribuição (Estimation of Distribution Algorithms)
  - Sem crossover
  - Novas soluções são geradas pelo uso de informações extraídas do conjunto de soluções promissoras

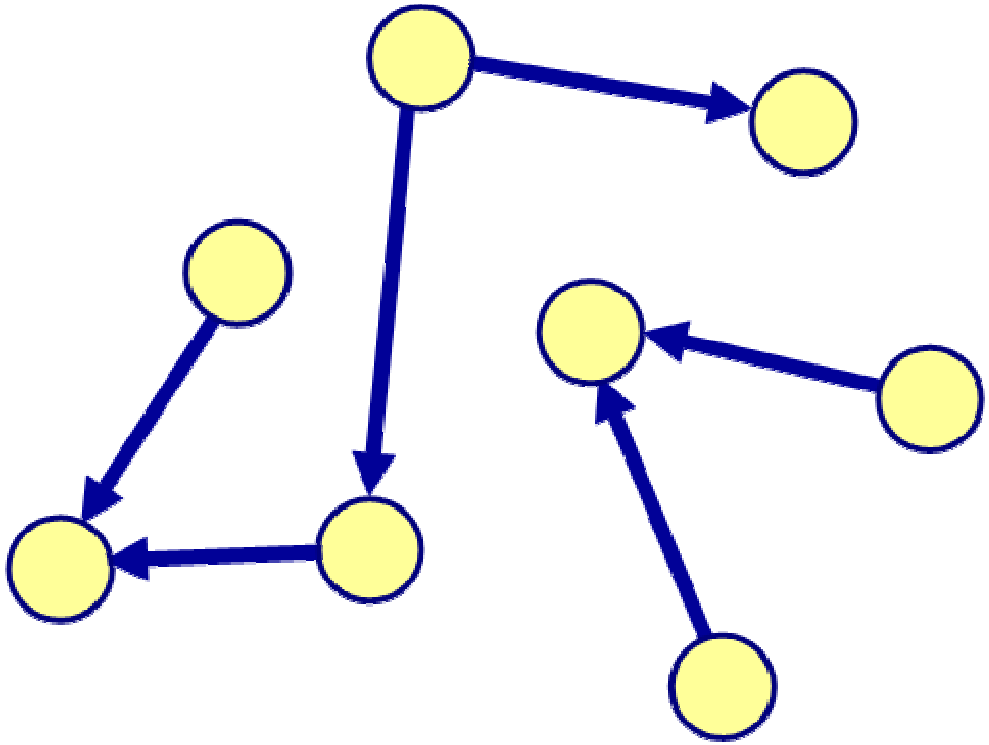
- Como extrair essas informações?

BOA

- EDA que adota um modelo do tipo rede Bayesiana para representar a população das melhores soluções a cada geração
  - Grafo orientado acíclico
  - Os nós são as variáveis (posições na *string*)
  - Dependências condicionais (arestas)
  - Independências condicionais (implícito)

EXEMPLO: REDE BAYESIANA

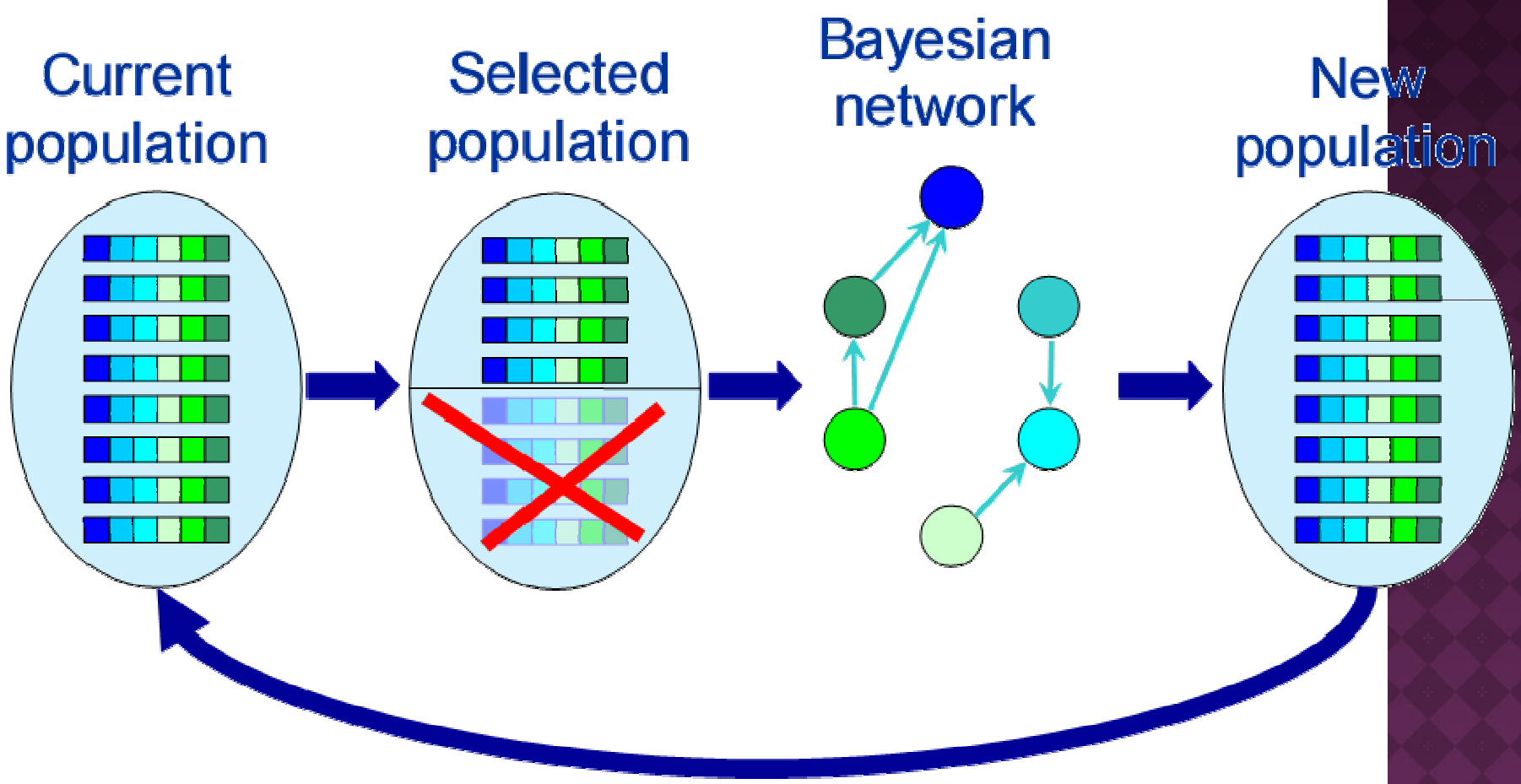
- Dependências condicionais
- Independências condicionais



BOA

- Não exige uma distribuição válida como entrada
- É capaz de aprender a distribuição durante a execução
  - sem a utilização de qualquer informação específica do problema.
- Informações pré-conhecidas podem ser incorporadas.

BOA


Current population
Selected population
Bayesian network
New population

QUESTÕES

- AG não resolve problemas do tipo TRAP
  - Resolve, se forem pequenos e compactos (bits dos subproblemas não espalhados)
- O BOA é um algoritmo completo
  - Não depende do AG
- A RB do BOA original é construída por um algoritmo guloso

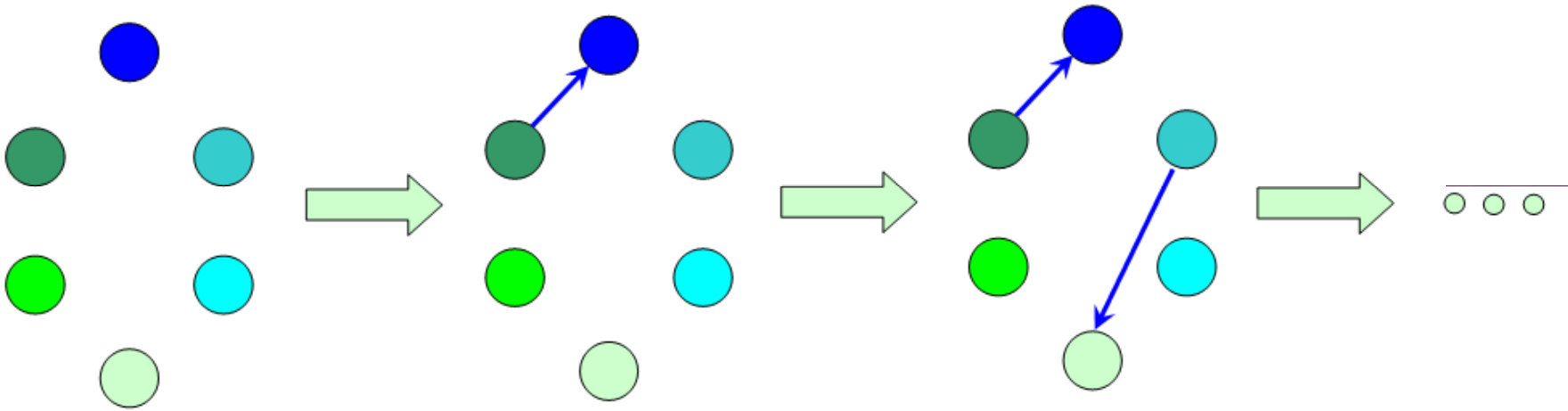
APRENDIZADO DA RB

- Encontrar um grafo que otimize uma métrica
- Algoritmo usado pelo BOA
  - *Greedy* (Guloso)
    - Garante um ótimo local
    - É rápido
- Métodos de otimização global
  - não produzem resultados muito melhores para o aprendizado da RD
  - são significativamente mais lentos
  - Exemplos
    - AG [Faulkner, 2007], PSO [Du, Tao 2005], ...
- **Balanço: qualidade do modelo X custo**

APRENDIZADO DE RB

- Métricas de avaliação do modelo
  - Usado pelo BOA
    - Bayesian-Dirichlet com equivalência de verossimilhança
    - K2 (BDe simplificado)
  - Bayesian information criterion – BIC [Schwarz, G. (1978)]
    - *-2*log-likelihood + nVar*log(nObs)*
  - *Likelihood* (Verossimilhança) do valor parâmetro b - L(b)
    - P(B|A) = (P(A|B)P(B)) / P(A)
    - L(b|A) = c P(A | B=x), c > 0
    - L(b1|A) / l(b2|A) é informativo
    - B em nosso caso são valores de variáveis de uma amostra – população promissora

EXEMPLO DE APRENDIZADO (GREEDY)



•DIFERENÇAS PARA O DOE

SEM INTERAÇÃO (VARIÁVEIS INDEPENDENTES)

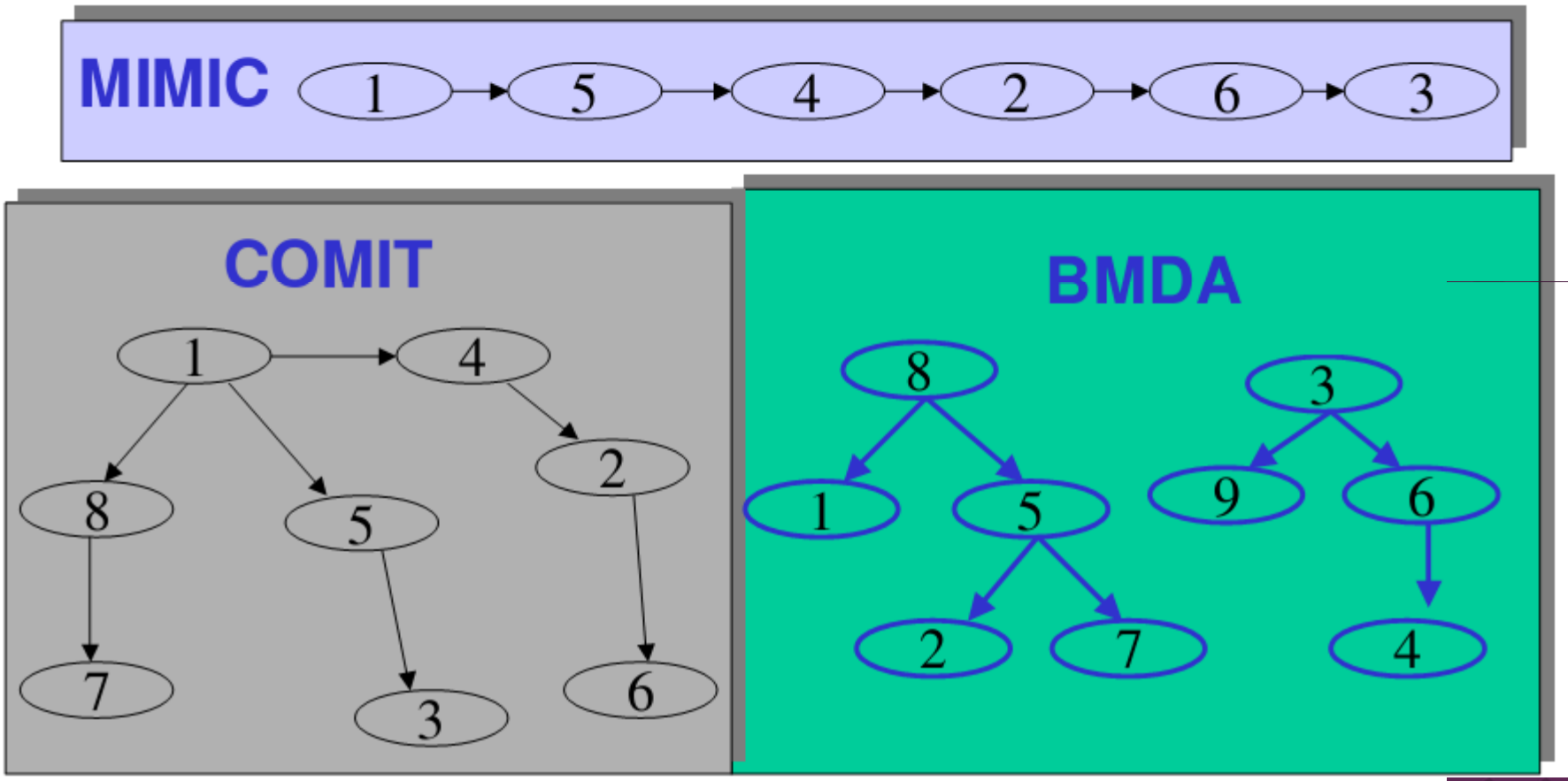
- Population Based Incremental Learning (PBIL) (1994)
- Compact Genetic Algorithm (cGA) (1998)
- Univariate Marginal Distribution Algorithm (UMDA) (1997)

- PBIL e CGA usam um vetor de probabilidades (p) enquanto o UMDA usa população
- PBIL e CGA usam regras diferentes para atualizar *p*

INTERAÇÃO ENTRE PARES

- Combining Optimizers with Mutual Information Trees (COMIT) (1997)
- Mutual-Information-Maximization Input Clustering (MIMIC) (1997)
- Bivariate Marginal Distribution Algorithm (BMDA) (1999)

INTERAÇÃO ENTRE PARES



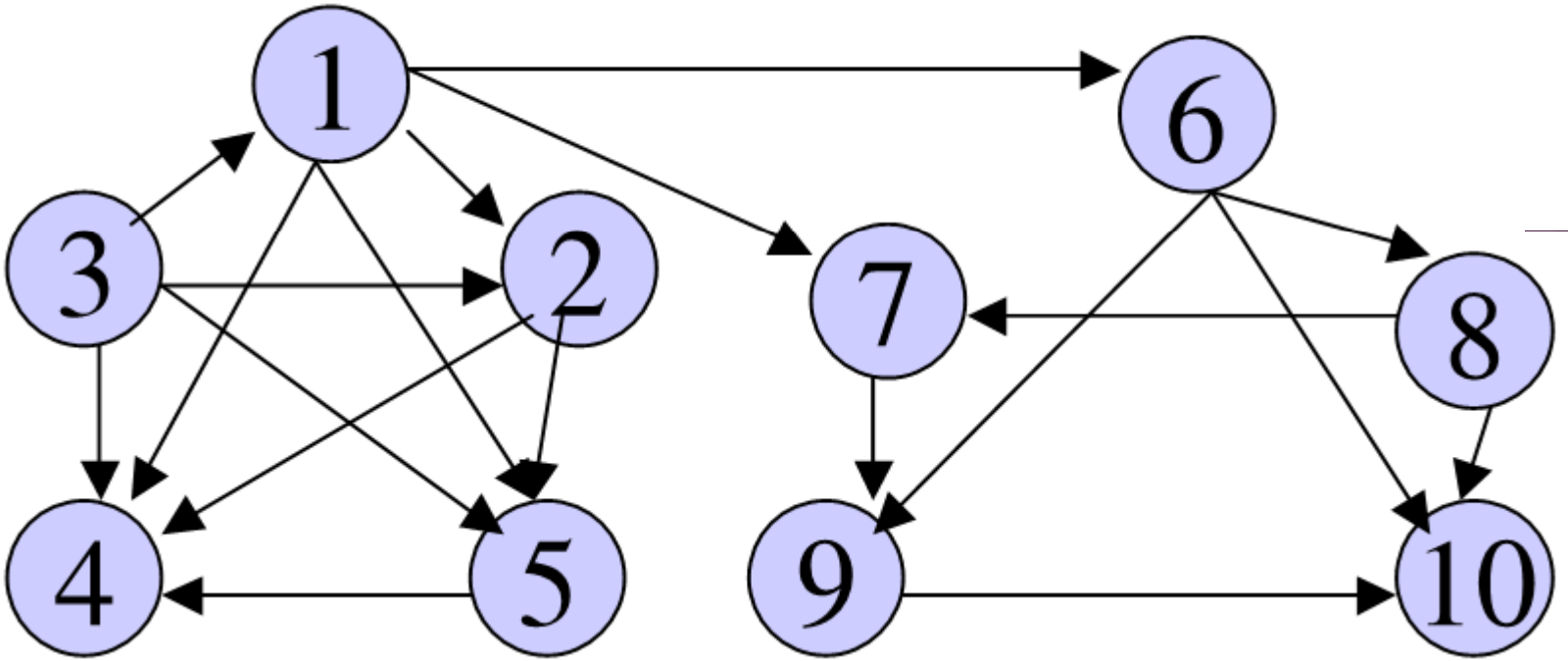
INTERAÇÕES MULTIVARIADAS

- Factorized Distribution Algorithm (FDA) (1998)
- Extended Compact Genetic Algorithm (ECGA) (1999)
- *Bayesian Optimization Algorithm (BOA) (1999)*

INTERAÇÕES MULTIVARIADAS

- BOA, EBNA e LFDA
- Usam redes bayesianas, mas diferentes métricas de avaliação
  - BOA: BDe
  - EBNA: BIC + K2 + penalização
  - LFDA: BIC com restrição de grau

INTERAÇÕES MULTIVARIADAS



PRÓS E CONTRAS

- Sem interações
  - Eficiente em problemas lineares.
  - Ignora BBs
- Pares
  - Eficiente com BBs de ordem 2.
  - Apenas em BBs de ordem 2

PRÓS E CONTRAS

- FDA
  - Eficiente em problemas decomponíveis
  - Informação a priori é essencial
- ECGA
  - Eficiente em problemas separáveis
  - Não funciona em BBs com sobreposição

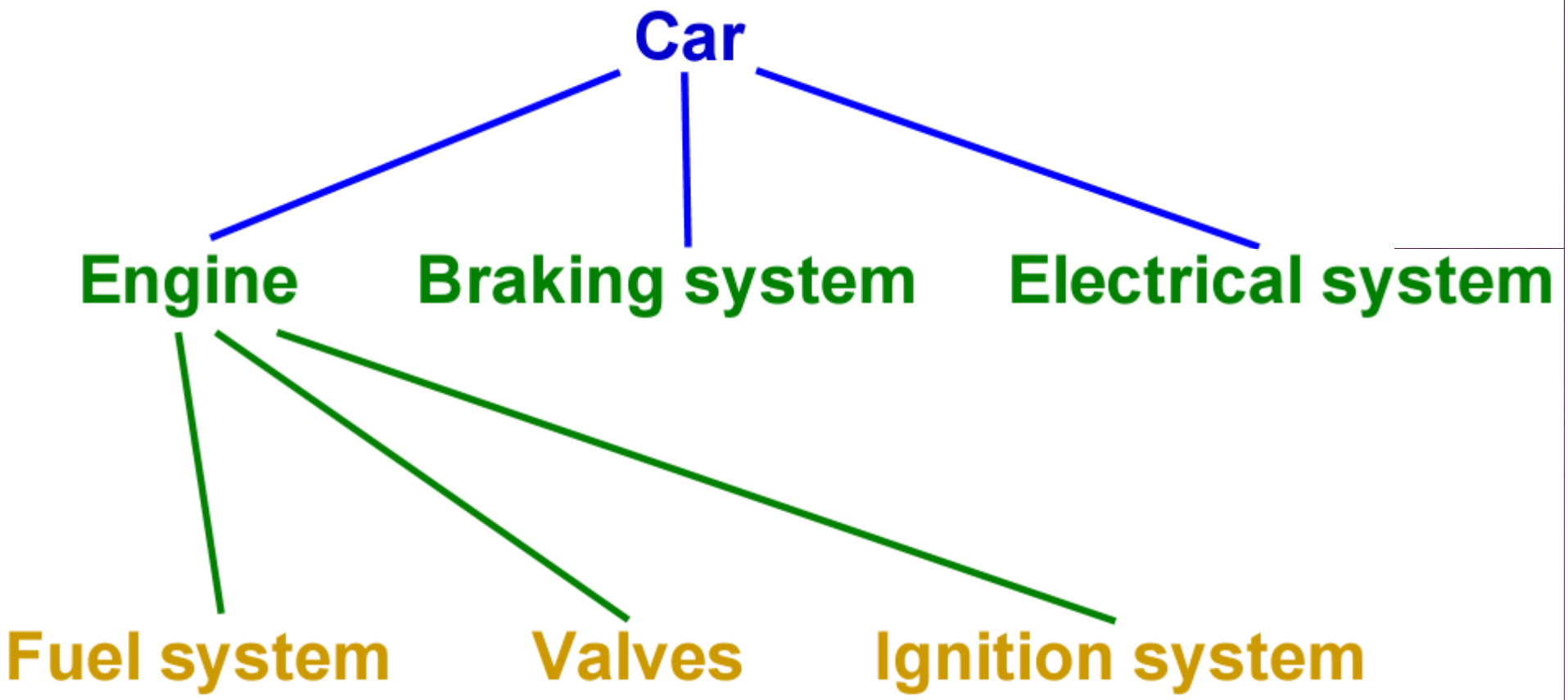
PRÓS E CONTRAS

- BOA
  - Genérico
  - Não tem contras ?
    - Custo do Modelo é maior
    - Definir Métrica adequada é um problema
    - Variável com alto grau de interação
      - Aumenta significativamente custo de cálculos
      - Diminui a representatividade do modelo

DE UM ÚNICO NÍVEL PARA UMA HIERARQUIA

- Poderoso na decomposição de problemas de único nível
- Mas e se não for suficiente?
- Aprenda com os seres humanos e a natureza
  - Decompor problema através de vários níveis
  - Usar soluções de baixo nível como módulos básicos
  - Resolver problema hierarquicamente

DECOMPOSIÇÃO HIERÁRQUICA



DECOMPOSIÇÃO HIERÁRQUICA

- Das **letras** de todos os textos em português
  - Aprender todas **palavras** (BBs)
  - Das palavras
    - As combinações possíveis, **sintaxe**
    - Da sintaxe
      - a **gramática**
      - Da gramática
        - » A **semântica**

PONTOS CHAVE

- A boa decomposição
  - Deve decompor problema em cada nível corretamente
- Chunking
  - Deve representar e manipular soluções de grande ordem
- Preservação de soluções alternativas
  - Deve preservar soluções parciais alternativas (pedaços)

HIERARCHICAL BOA (HBOA)

- Pelikan& Goldberg (2000, 2001)
- A boa decomposição
  - Usa redes Bayesianas como o BOA
- Chunking
  - Usar estruturas locais nas redes Bayesianas
- Preservação de soluções alternativas (niching)
  - Aumentar a diversidade entre as selecionadas
  - Usa substituição por torneio restrito (RTR)

ESTRUTURAS LOCAIS NAS REDES BAYESIANAS

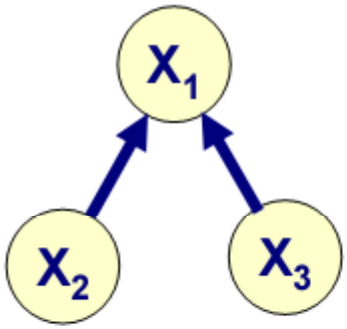
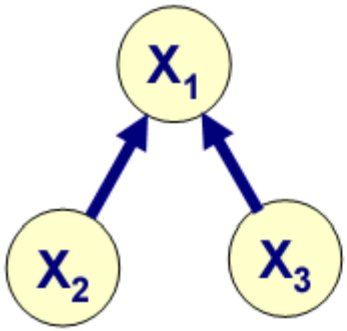


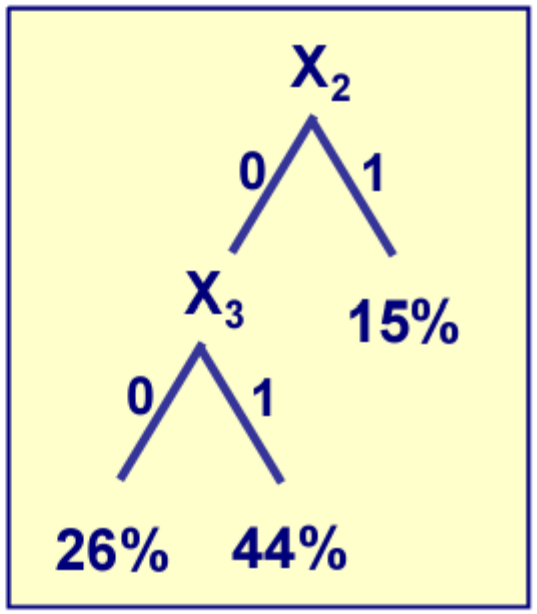
Tabela de Probabilidades Condicionais

X_2X_3	$P(X_1=0\|X_2X_3)$
00	26 %
01	44 %
10	15 %
11	15 %

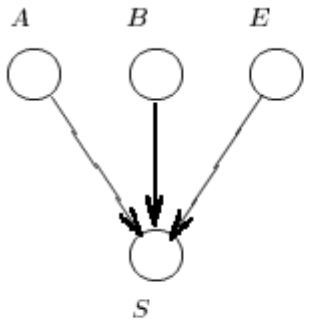
ESTRUTURAS LOCAIS NAS REDES BAYESIANAS



Árvore de Decisão



ESTRUTURAS LOCAIS NAS REDES BAYESIANAS

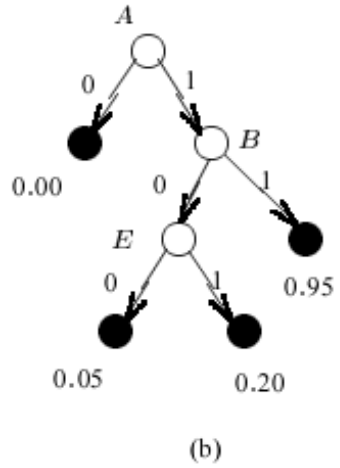


A	B	E	$\Pr(S \mid A, B, E)$
1	1	1	0.95
1	1	0	0.95
1	0	1	0.20
1	0	0	0.05
0	1	1	0.00
0	1	0	0.00
0	0	1	0.00
0	0	0	0.00

Figure 1: A simple network structure and the associated CPT for node S.

A	B	E	$\Pr(S \mid A, B, E)$
1	1	1	0.95
1	1	0	0.95
1	0	1	0.20
1	0	0	0.05
	*		0.00

(a)



(b)

Figure 2: Example of the two representations of the local CPT structure. Part (a) shows a *default table*, and Part (b) shows a *decision tree*.

RESTRICTED TOURNAMENT REPLACEMENT (RTR)

- Usado no HBOA para niching
- Inserção de uma nova solução x na população:
  - Escolha um subconjunto aleatório da população original
  - Encontre a solução y mais semelhante à x no subconjunto (distância de Hamming)
  - Substituir y por x se x é melhor do que y.

ARMADILHAS HIERÁRQUICAS

- Combinar armadilhas em mais níveis
- Cada nível contribui para o fitness
- Grupos de bits mapeiam para o próximo nível

