Schemata Monte Carlo Tree Optimization

P. Isasi
Computer Science Department, Carlos III of Madrid University

M. Drugan[2] , B. Manderick
Articial Intelligence Lab, Vrije Universitieit Brussels

Index

- *Background*
- *Binary schemata*
- *S-MonteCarlo Network*
- *Procedure*
- *Domains*
- *Results*

Background

- *MCTS methods have been taken an increasing attention in the last 15 years*
- *Originally hi gather a spectacular success in the the game of GO*
- *Later applied successfully in designing expert computer players for many others two players games:*
  - *Hex, Kriegspiel, Poker*
- *Outperform alphabeta search even where good heuristic evaluations are difficult to obtain*
- *Recently, some other domains have been addressed*

Binary Combinatorial Optimization

- *BCO consists in the finding of a binary string that represents the optimal combination of yes/no alternatives.*
- *The difficulty resides in the weakness of the binary representation.*
- *The GA tries to overcome it with an implicit parallel evaluation of schemata.*
- *However there is no explicit use of schemata to improve the search of better solutions.*

Binary schemata

- *In this work we try obtain binary schemata from whom good solutions could be generated*
- *Schemata must be generated being fitter and more specific in time*
- *Schemata are used to generate, randomly, solutions to the problem, and those solutions are used to estimate the efficacy of the schemata*

0011111001 0001111010

0111011001 0101011001

0101101010 0**1**10** 0101001000

0111111011 0001001010

0011101001 0001001001

S-MonteCarlo Network

- *Generating good schemata have many intrinsic problems.*
  - *The space of schemata is much bigger that those of the solutions (3^L vs. 2^L)*
  - *Schemata evaluation is a difficult task because its size and structure*
- *We propose the use of MCTS:*
  - *They could deal with incomplete information*
  - *They can improve the efficiency of accuracy estimations for schemata*
  - *They use a tree structure that fits very well the intrinsic nature of the schemata*

S-MonteCarlo Network

- *Nodes in the network are composed by schema*
- *The first node is the more general schema, don't care symbol in every position*
- *Each node is one level more specific that its parent*
- *Last level nodes contains schema of a fixed small specificity, allowing the generation of every posible individual*
- *The other nodes are evaluated by a sampling set of individuals generated randomly*

Procedure

- *In each iteration the most promising node is selected for expansion*
- *The network grows in a unstructured and unbalanced way*
- *All nodes not fully expanded could be considered for selection*
- *The procedure is performed in four phases:*
  - *Selection*
  - *Expansion*
  - *Simulation*
  - *Backpropagation*

Selection

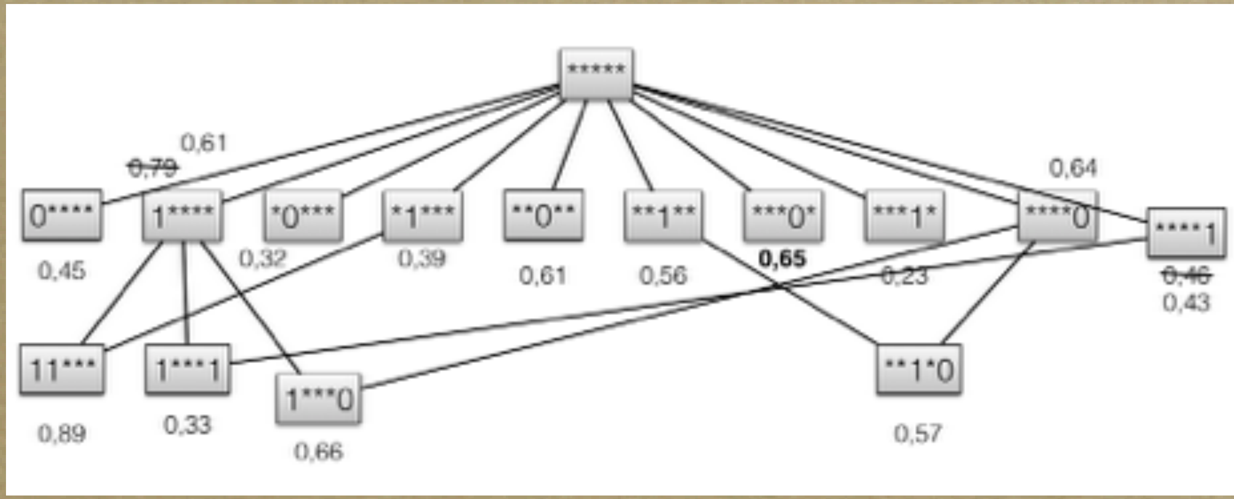
- *The more promising unexpanded node of the network must be selected*
- *A tree policy is designed to decide the meaning of being a promising node*
    - *There are many tree policy, in this work we propose the use of the Upper Confidence Policy (UCP)*

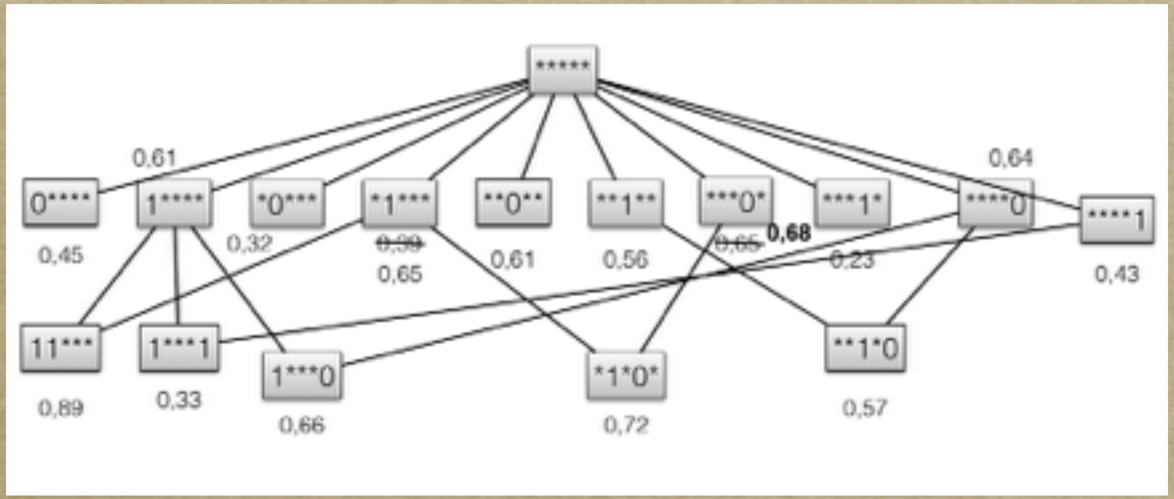
$$M(S_k) = f(S_k) + C\sqrt{\frac{\log N_k}{n_k}}$$

- *M(Sk) -> Criteria for a node Sk to be selected*
- *f(Sk) -> Evaluation of node Sk*
- *c is the exploration parameter*
- *Nk -> Size of the sampling set for node Sk*
- *nk -> Number of examples used historically to evaluate node Sk*

Expansion

- *The selected node is expanded, generating one descendant, following random expansion policy*
- *Once a descendant is generated, it is liked with all their parents, those that are less general and matching the new one*





Simulation

- *The new generated node is from a sampling set of individuals represented by the schema of that node*
- *The solutions in the sampling set are generated randomly, fitting the rules of the schema, and are not stored*
- *The value assigned to the schema will be the average of the evaluation values of its sampling set*

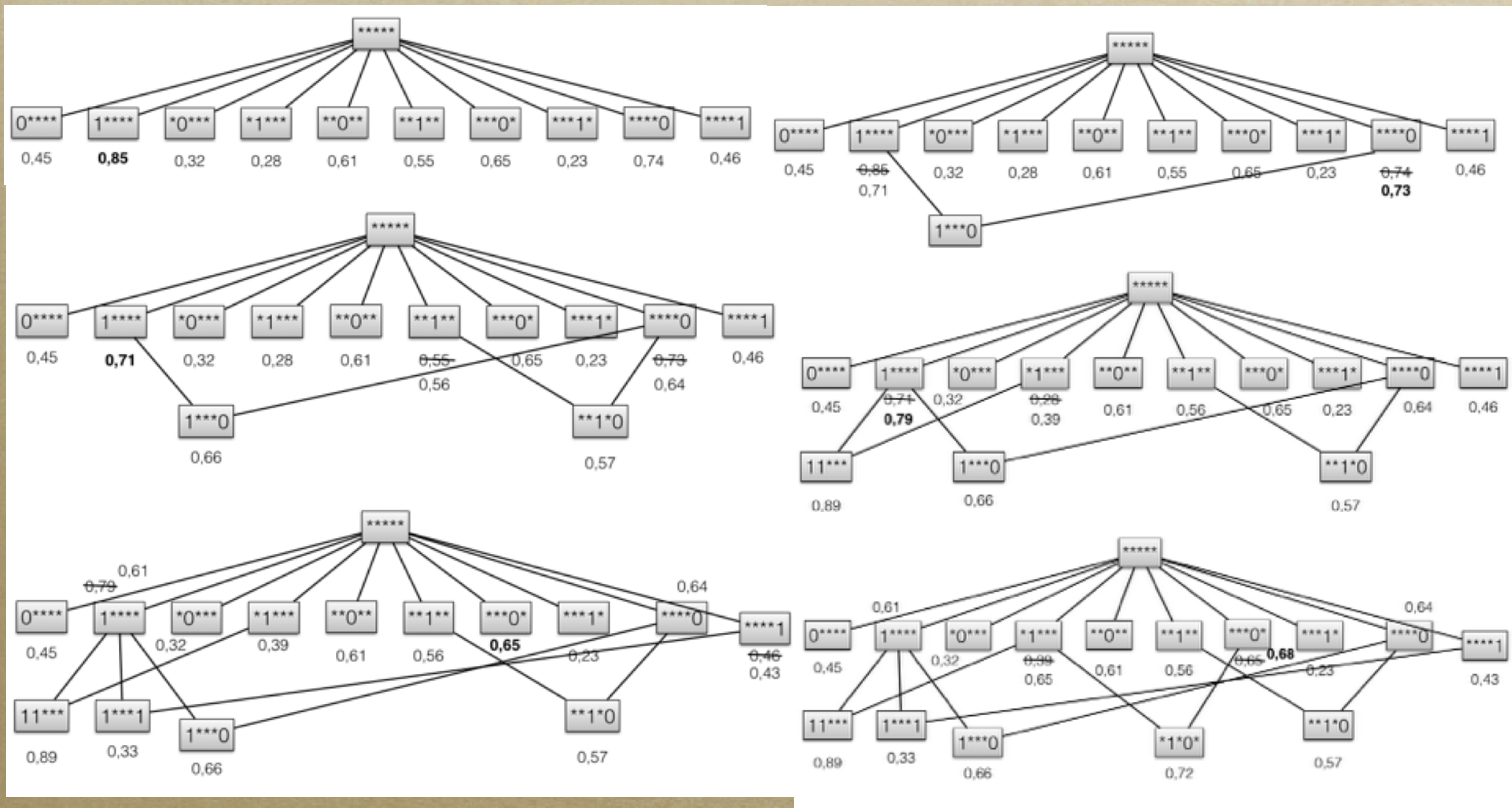
BackPropagation

- *All the evaluation values of all the nodes that are ancestors to the new created node are updated*

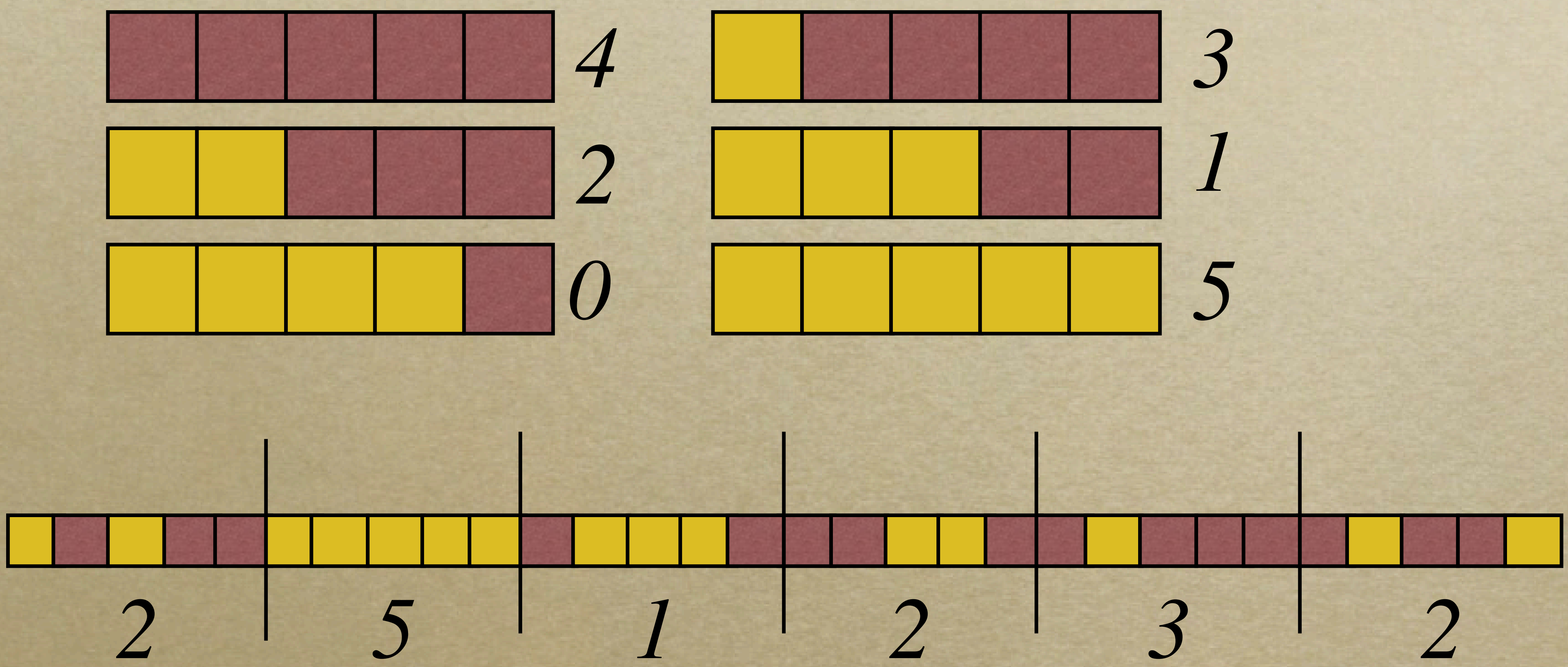
$$f(S_k) = \frac{(t \cdot f(S_k)) + v(S_k)}{t+1}$$

$$v(S_k) = \frac{\sum_{I_i \in S_k} f(I_I)}{n}$$

Example of iterations



Deceptive trap function



$f=15/30=0{,}5$

Knapsack problem

$$\sum_{i=1}^{n} w_i x_i$$

$$\sum_{i=1}^{n} w_i x_i \leq W$$

$$x \in \{0,1\}$$

Results

Deceptive function			
Size	Best solution	Accuracy (mean, deviation)	Evaluations (mean, deviation)
20	0,96	0,94±0,02	154831±152774
25	0,96	0,92±0,01	282726±185339
30	0,91	0,89 ±0,01	404122±269166
35	0,91	0,88±0,01	526249±237347
40	0,9	0,87±0,01	623444±141584
45	0,89	0,86±0,01	754003±146172
50	0,88	0,85±0,008	890116±104346
Knapsack problem			
24	1,0	1,0±0,0	368562±93717