

INTRODUCTION OF STATISTICS IN OPTIMIZATION

Fabien Teytaud
Under the supervision of
Marc Schoenauer and Olivier Teytaud



UNIVERSITÉ
PARIS-SUD 11

Inria
INVENTEURS DU MONDE NUMÉRIQUE



OUTLINE

- Evolutionary Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- Multistage Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- Conclusion

OUTLINE

- **Evolutionary Optimization**
 - **Problem**
 - State Of The Art Algorithms
 - Contributions
- Multistage Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- Conclusion

EVOLUTIONARY OPTIMIZATION

- Data :
 - Search space Ω (set of possible solutions)
 - Objective function (quality criterion)
- Goal :
 - Find the best solution (according to the objective function)
- Formally :
 - Consider $\mathcal{F} : \Omega \rightarrow \mathbb{R}$
 - Find $x^* \in \Omega / x^* = \text{ArgMax}(\mathcal{F})$
- Additional properties
 - Black-box optimization
 - Continuous ($\Omega \subset \mathbb{R}^N \rightarrow$ Evolution Strategy)

OUTLINE

- **Evolutionary Optimization**
 - Problem
 - **State Of The Art Algorithms**
 - Contributions
- Multistage Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- Conclusion

EVOLUTION STRATEGIES

[RECHENBERG, 73],[SCHWEFEL, 74]

Initialize y, σ, C

While (we have time)

{

$$x_i = y + \sigma Ni(0, C) \quad \text{for } i = 1, \dots, \lambda$$

Evaluate all offspring x_i (compute $\mathcal{F}(x_i)$ for $i = 1, \dots, \lambda$)

Update y, σ, C

}

Return y

$y \in \mathbb{R}^d$ represents the current search point
 λ is the population size
 $C \in \mathbb{R}^{d \times d}$ is the covariance matrix
 $\sigma \in \mathbb{R}_+$ is the step-size

EVOLUTION STRATEGIES

Initialize y , σ , C

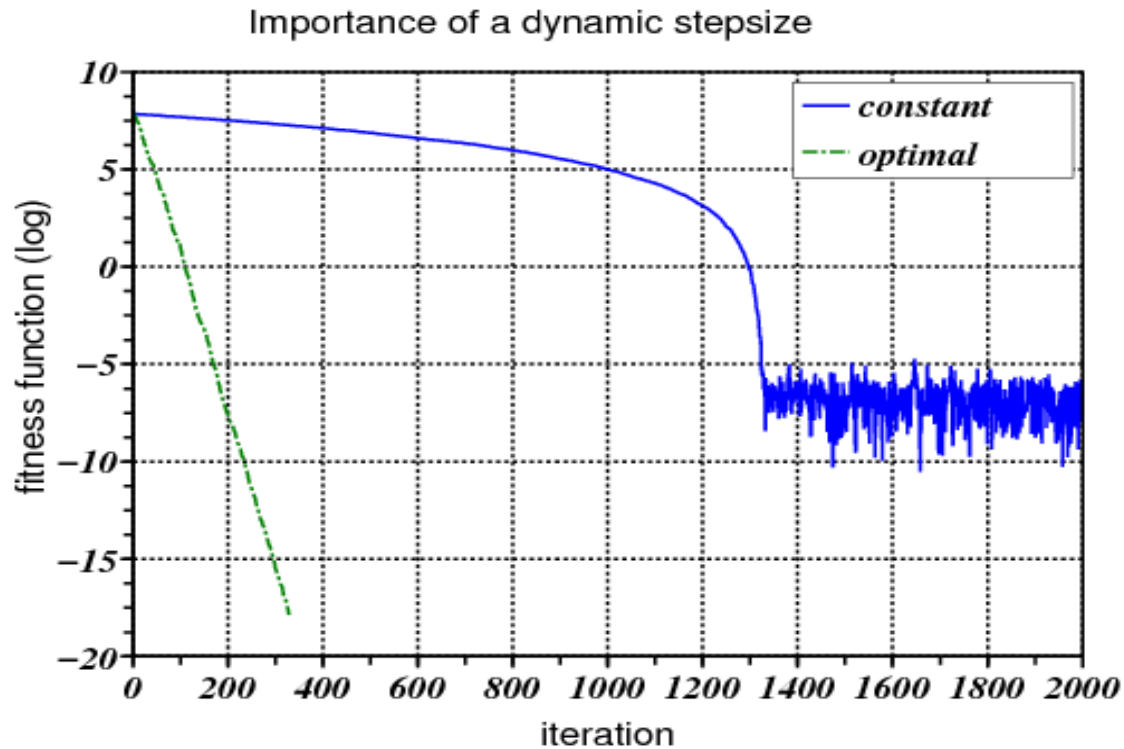
While (we have time)

```
{  
     $x_i = y + \sigma Ni(0, C)$  for  $i = 1, \dots, \lambda$   
    Evaluate all offspring  $x_i$  (compute  $\mathcal{F}(x_i)$  for  $i = 1, \dots, \lambda$ )  
    Update  $y$ ,  $\sigma$ ,  $C$   
}
```

Updating y : $y = \sum_{i=1}^{\mu} w_i x_i \rightarrow (\mu/\mu, \lambda)$ -ES

IMPORTANCE OF UPDATING σ

Scale invariant : $\sigma = \alpha ||x||$



EVOLUTION STRATEGIES

- Covariance matrix : Identity or diagonal
 - 1/5th [Rechenberg, 73]
 - Self-Adaptation (SA) [Rechenberg, 73], [Schwefel, 74]
 - Path Length Control (aka CSA) [Hansen & Ostermeier, 96,01]
- Full covariance matrix – self adaptation [Schwefel, 81]

State of the art algorithms

- Covariance Matrix Adaptation (CMA)
 - [Hansen & Ostermeier, 01]
- Covariance Matrix Self-Adaptation (CMSA)
 - [Beyer & Sendhoff, 08]

ESTIMATION OF MULTIVARIATE NORMAL ALGORITHM (EMNA) [LARRANAGA & LOZANO, 01]

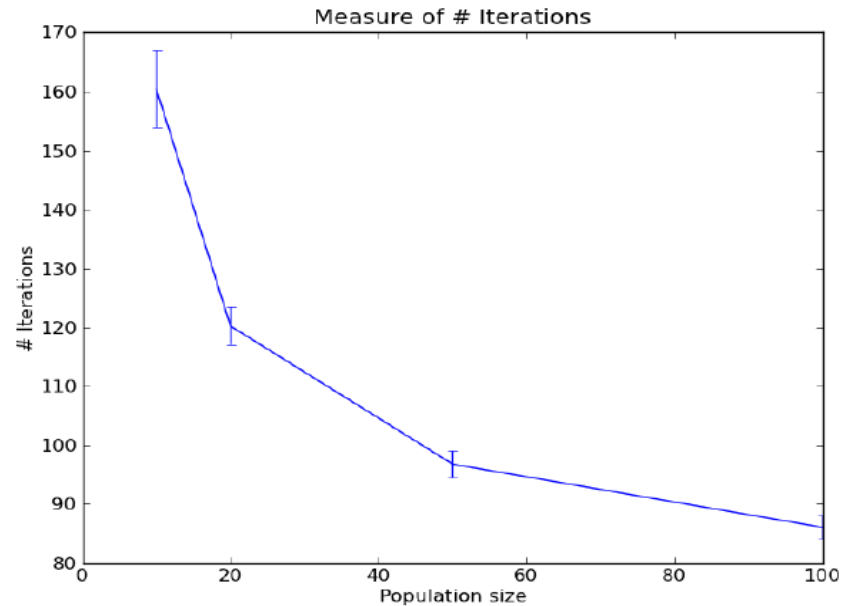
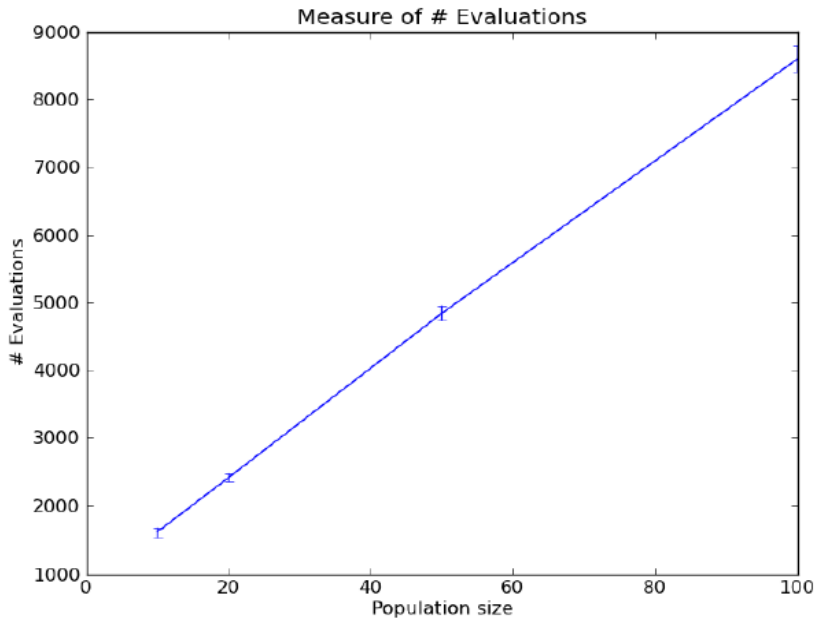
- EDA : evolution of a parameterized probability distribution
 - Sample the domain with the current distribution
 - Evaluate the population
 - Select the best points
 - Update the parameters of the distribution
- EMNA
 - Gaussian distribution (σ, C)
 - Close to ES

OUTLINE

- **Evolutionary Optimization**
 - Problem
 - State Of The Art Algorithms
 - **Contributions**
- Multistage Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- Conclusion

MOTIVATION FOR PARALLELISM

CMA, Sphere function, $f_{\text{target}}=10^{-10}$, $N=10$



→ How to efficiently use all λ available processors ?

FITNESS FUNCTIONS

- Sphere function : $x \rightarrow ||x||$
- Schwefel function : $x \rightarrow \sum_{i=1}^N (\sum_{j=1}^i x_j)^2$
- A multimodal function : $x \rightarrow \sum_i \log(|x_i|) + \cos(\frac{1}{x_i})$
- **Minimization**
 - In all experiments : **The lower the better**

THEORETICAL BOUNDS

Speedup for p processors :

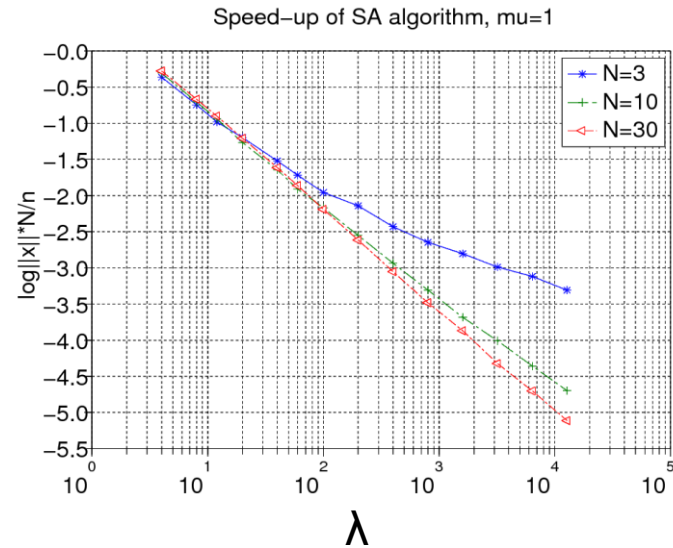
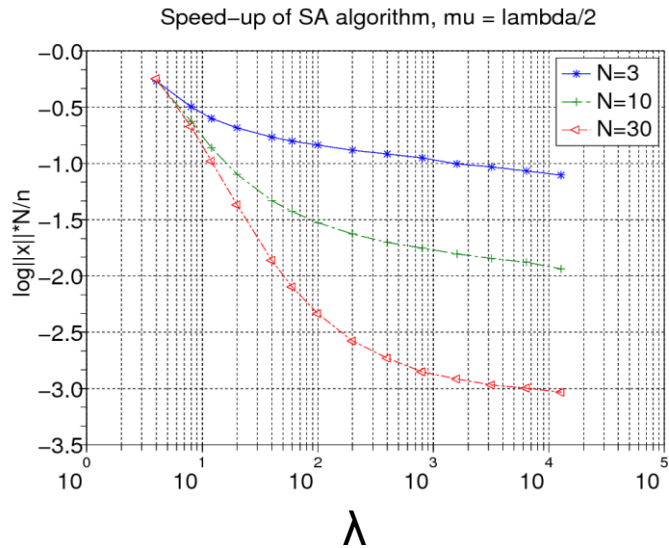
$$S_p = T_1 / T_p$$

- (1, λ)-ES → speedup logarithmic for all λ
- ($\mu/\mu, \lambda$)-ES → speedup linear until $\lambda < N$ and then logarithmic

[Teytaud & Fournier, 10]

ISSUES WITH LARGE POPULATION SIZES

Current algorithms are far from the theoretical bounds
[Evonum 2009]



$\mu=1$: empirically, a better choice than $\lambda/2$ or $\lambda/4$

CONTRIBUTIONS (1)

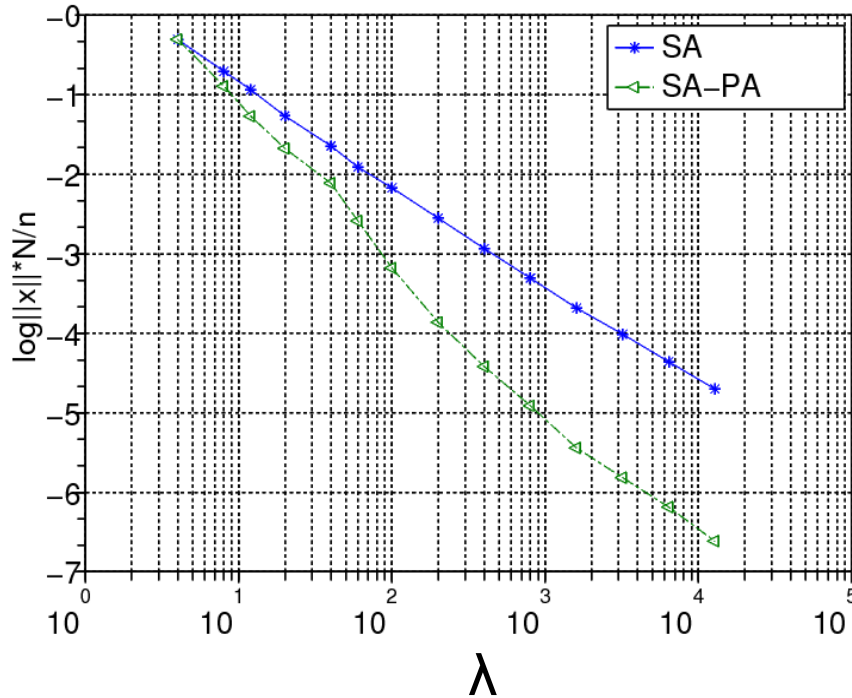
- New selection ratio (aka PA) [Evonum 2010]
 - A bounded selection ratio

$$\mu = \min(N, \lambda/4)$$

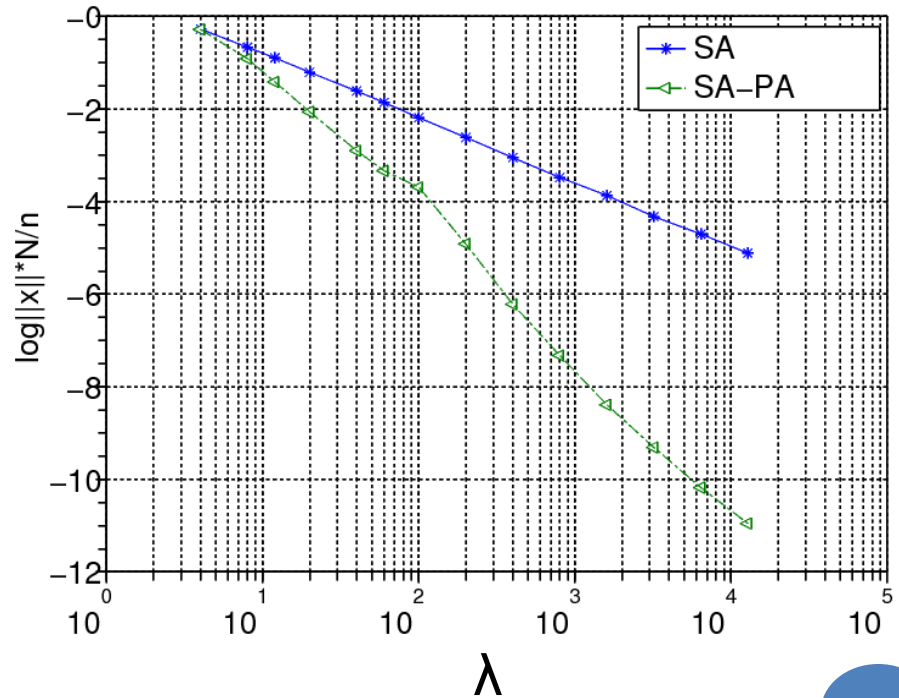
- Experiments
 - SA, Sphere function, against $\mu=1$
 - CMSA, Sphere and Schwefel functions, against $\mu=\lambda/4$

NEW SELECTION RATIO : EXPERIMENTAL RESULTS (1)

Speed-up of SA algorithm, N=10

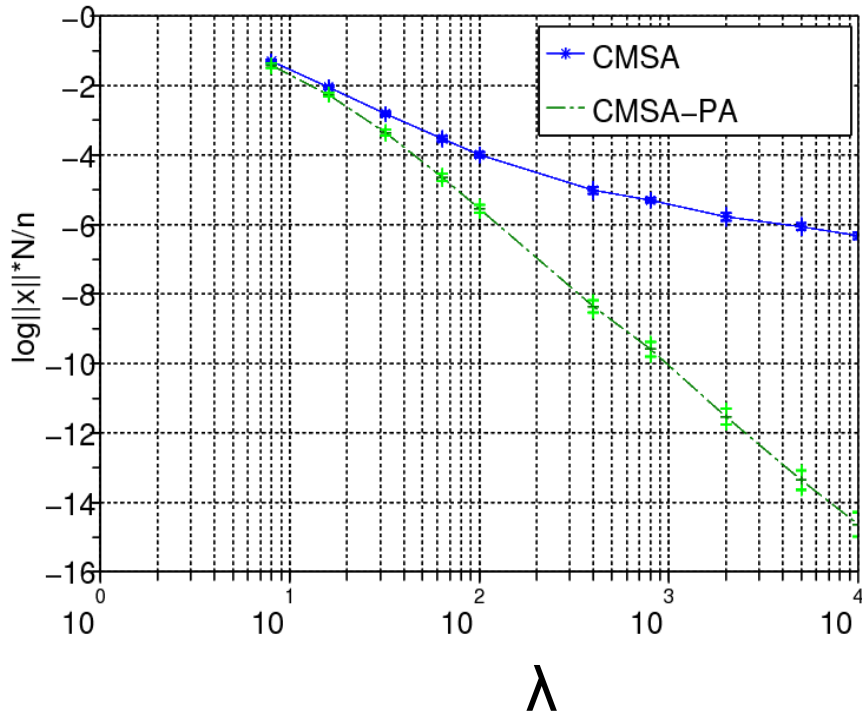


Speed-up of SA algorithm, N=30

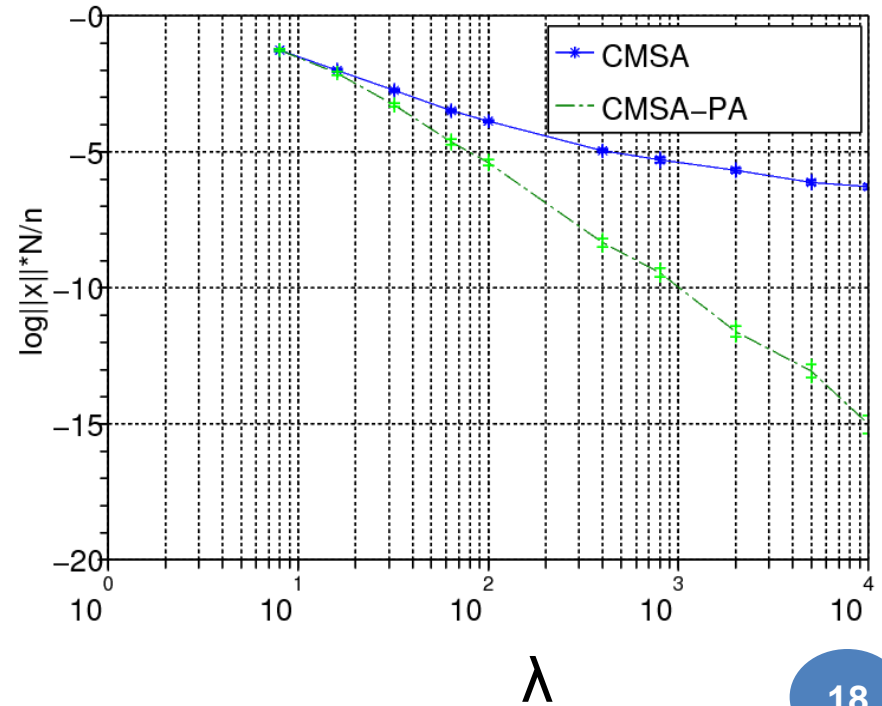


NEW SELECTION RATIO : EXPERIMENTAL RESULTS (2)

Speed-up of CMSA algorithm, Sphere function, N=3



Speed-up of CMSA algorithm, Schwefel function, N=3



CONTRIBUTIONS (2)

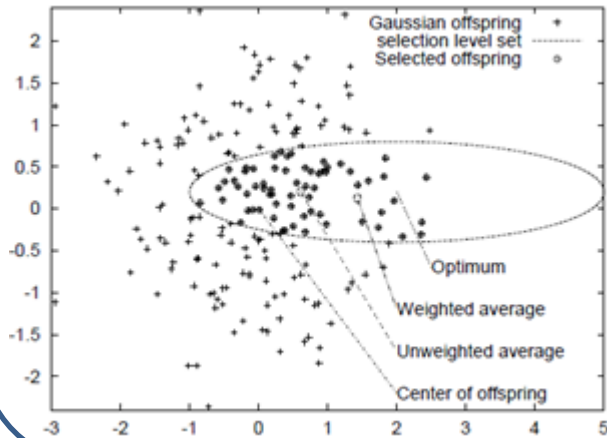
- Log- λ modification [EA 09, PPSN 10]

A faster decrease of the step-size : $\sigma \leftarrow \sigma / \max(1, (\log(\lambda)/2)^{1/N})$

With :

Reweighting [GECCO 09]

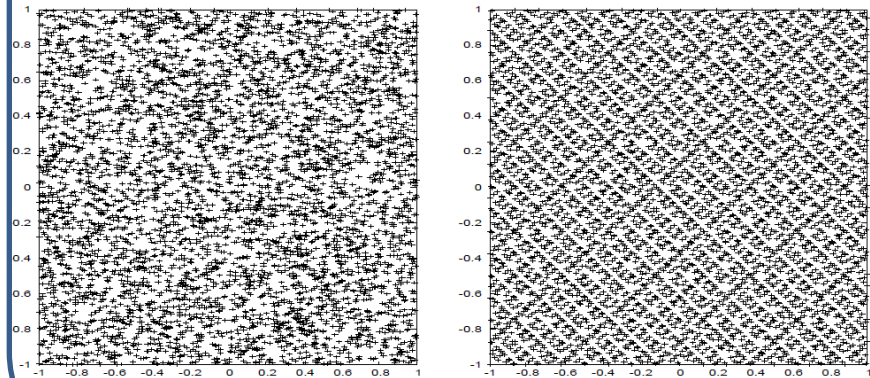
(weight=1/density of distribution)



+

Quasi-Random mutations

[Teytaud, 08]



EXPERIMENTS

- EMNA with
 - Faster decrease of σ
 - Reweighting
 - Quasi-Random mutations
- Functions :
 - Sphere function
 - Multimodal function
- σ initializations :
 - Tuned
 - Not tuned
- All differences statistically significant (with 95% confidence)

EXPERIMENT RESULTS (1)

➤ Sphere function

➤ Initial σ tuned

Dimension, lambda	Baseline	+QR	+ weight	+log(λ)-Modif (IEMNA)
2,20	-0.345	-1.252	-1.743	-2.103
2,60	-1.967	-2.086	-2.022	-2.713
2,200	-2.050	-2.089	-2.112	-3.004
2,600	-2.080	-2.101	-2.061	-3.190
2,2000	-2.103	-2.188	-2.111	-3.434
3,30	-0.697	-2.299	-2.277	-2.398
3,90	-2.330	-2.392	-2.282	-3.047
3,300	-2.340	-2.404	-2.293	-3.302
3,900	-2.369	-2.443	-2.320	-3.519
3,3000	-2.404	-2.476	-2.367	-3.726
4,40	-0.749	-2.541	-2.397	-2.578
4,120	-2.543	-2.627	-2.443	-3.271
4,400	-2.601	-2.658	-2.480	-3.547
4,1200	-2.642	-2.717	-2.519	-3.764
5,50	-1.330	-2.885	-2.677	-2.730
5,150	-2.790	-2.858	-2.622	-3.488
5,500	-2.828	-2.908	-2.673	-3.750
5,1500	-2.886	-2.964	-2.718	-3.975

EXPERIMENT RESULTS (2)

➤ Multimodal function

➤ Initial σ tuned

Dimension, lambda	Baseline	+QR	+ weight	+log(λ)-Modif (IEMNA)
2,20	-0.709	-1.301	-1.105	-0.529
2,60	-1.139	-1.181	-0.655	-1.157
2,200	-1.104	-1.074	-0.402	-0.822
2,600	-1.100	-1.133	-0.119	-0.534
2,2000	-1.124	-1.146	-0.124	-0.210
2,6000	-1.144	-1.162	-0.173	-0.181
3,30	-0.971	-1.332	-0.799	-0.537
3,90	-1.231	-1.229	-0.481	-0.619
3,300	-1.210	-1.243	-0.178	-0.233
3,900	-1.240	-1.252	-0.181	-0.179
3,3000	-1.269	-1.307	0.081	-0.031
4,40	-1.204	-1.388	-0.713	-0.858
4,120	-1.357	-1.353	-0.344	-0.480
4,400	-1.352	-1.368	-0.205	-0.183
4,1200	-1.389	-1.427	0.224	0.093
5,50	-1.359	-1.520	-0.702	-0.445
5,150	-1.477	-1.503	-0.351	-0.391
5,500	-1.495	-1.518	-0.145	-0.161
5,1500	-1.539	-1.579	0.726	0.715

EXPERIMENT RESULTS (3)

➤ Sphere function

➤ Initial σ not tuned

Dimension, lambda	Baseline	+QR	+ weight	+log(λ)-Modif (IEMNA)
2,20	-0.000	-0.001	-0.001	-0.001
2,60	-0.001	-0.001	-0.014	-0.005
2,200	-0.001	-0.001	-1.501	-2.106
2,600	-0.001	-0.001	-1.748	-2.640
2,2000	-0.001	-0.001	-1.853	-2.952
3,30	-0.001	-0.001	-0.003	-0.002
3,90	-0.002	-0.003	-0.165	-0.096
3,300	-0.003	-0.003	-1.821	-2.437
3,900	-0.003	-0.003	-1.995	-2.945
4,40	-0.002	-0.003	-0.005	-0.004
4,120	-0.004	-0.004	-0.395	-0.508
4,400	-0.004	-0.005	-1.970	-2.693
4,1200	-0.005	-0.005	-2.131	-3.086
5,50	-0.003	-0.004	-0.007	-0.008
5,150	-0.005	-0.006	-0.609	-0.779
5,500	-0.006	-0.007	-2.087	-2.786
5,1500	-0.007	-0.007	-2.288	-3.250

DISCUSSION

- New selection ratio (μ/λ)
 - Positive improvement
 - No new parameter
 - Better as λ increases
- Faster decrease of σ
 - Improved the speed-up of EMNA
 - Can be dangerous if no proper reweighting/initialization
 - (Quasi-random always good)

OUTLINE

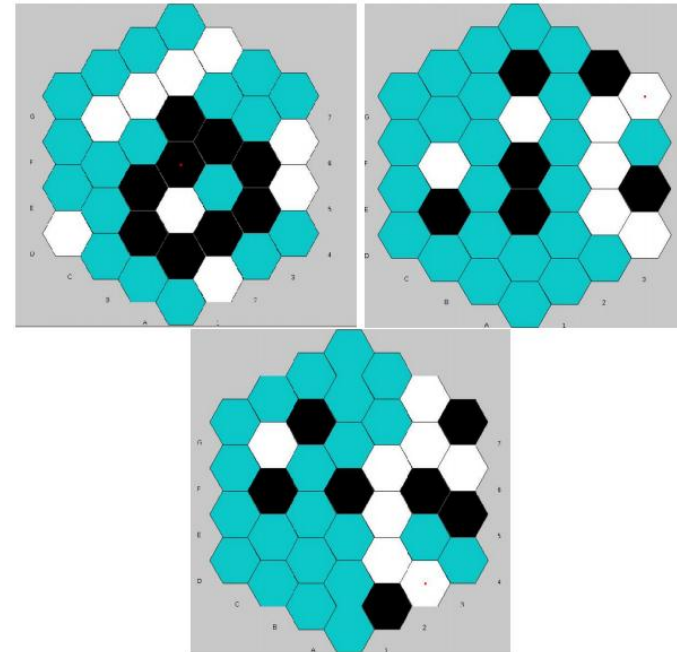
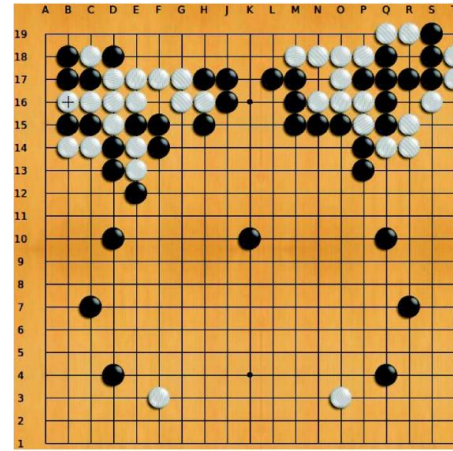
- Evolutionary Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- **Multistage Optimization**
 - **Problem**
 - State Of The Art Algorithms
 - Contributions
- Conclusion

PROBLEM

- Making decisions in an environment which is
 - Discrete
 - Fully observable
 - With finite horizon
 - Reward at the end
 - With a large number of states
- Goal : Find the best decision for each state

GAMES

- Why games ?
 - Well-designed
 - Simple and practical
- Which games ?
 - Go
 - Havannah
- Goals
 - Improving the algorithm
 - Keeping the generality of the algorithm



OUTLINE

- Evolutionary Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- **Multistage Optimization**
 - Problem
 - **State Of The Art Algorithms**
 - Contributions
- Conclusion

STATE OF THE ART ALGORITHMS

- Two-player games
 - Min-Max
 - Alpha-Beta
 - **Monte-Carlo Tree Search**
- One-player games
 - Dynamic programming
 - Nested Monte-Carlo

Monte-Carlo Tree Search (MCTS)

- Recent developments

 - [Coulom 06], [Chaslot et al., 06], [Kocsis and Szepesvari, 06]

- Numerous applications

 - Active learning [Rolet et al.,09]

 - Non-linear optimization [Auger & Teytaud, 10]

 - Feature selection [Gaudel & Sebag, 09]

 - Planning [Xie et al.,11]

 - Games

 - Go,

 - Havannah (First use [ACG 2010])

 - ...

MONTE-CARLO TREE SEARCH (MCTS)

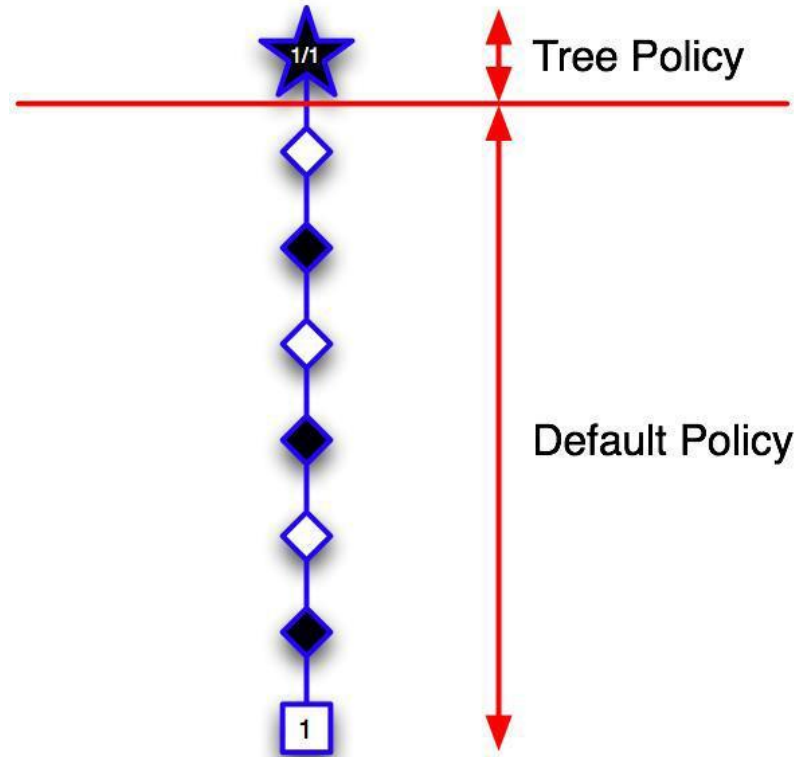
➤ Principle

- Construction of an imbalance subtree of possible futures
- Evaluation through Monte-Carlo simulations
- Use of a bandit formula to bias the subtree

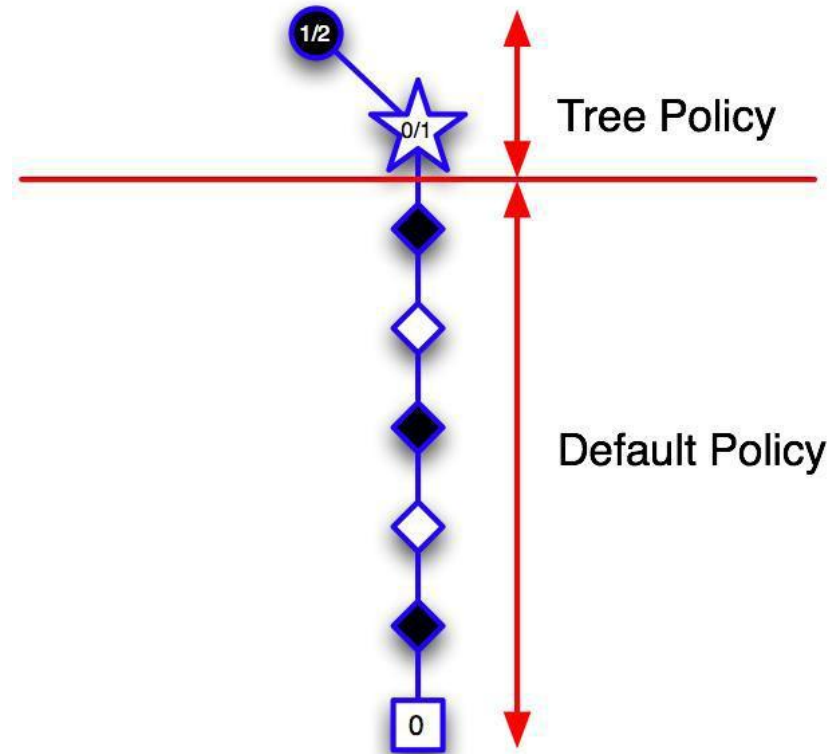
➤ 3 main steps

- Descent in the subtree
- Evaluation of the leaves
- Growth and update of the subtree

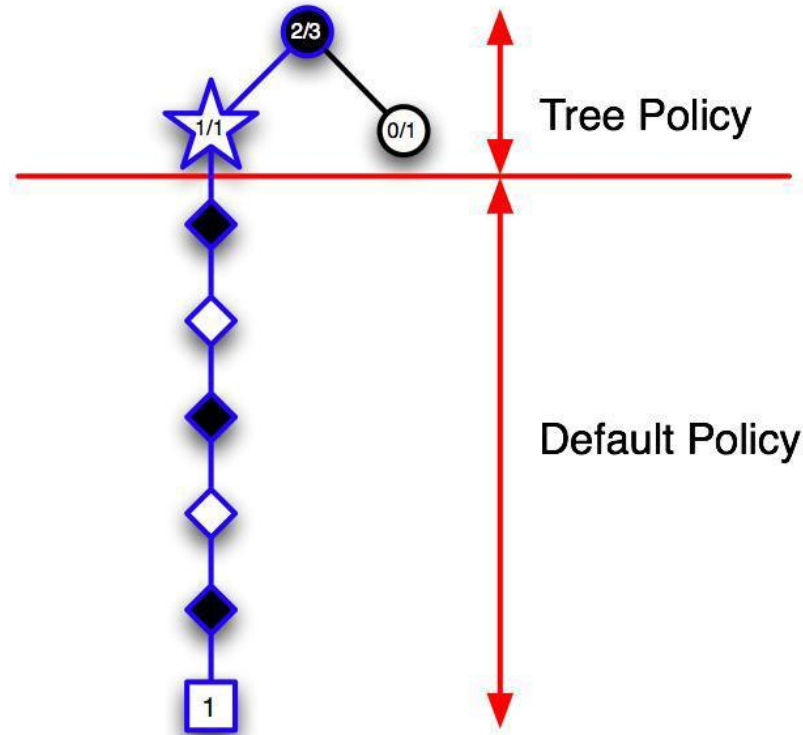
Monte-Carlo Tree Search (MCTS)



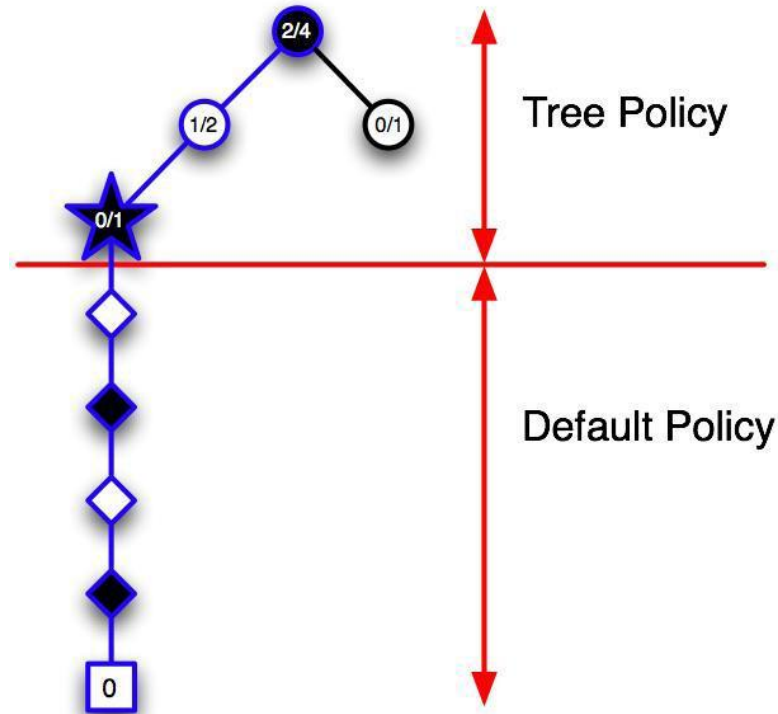
MONTE-CARLO TREE SEARCH (MCTS)



MONTE-CARLO TREE SEARCH (MCTS)



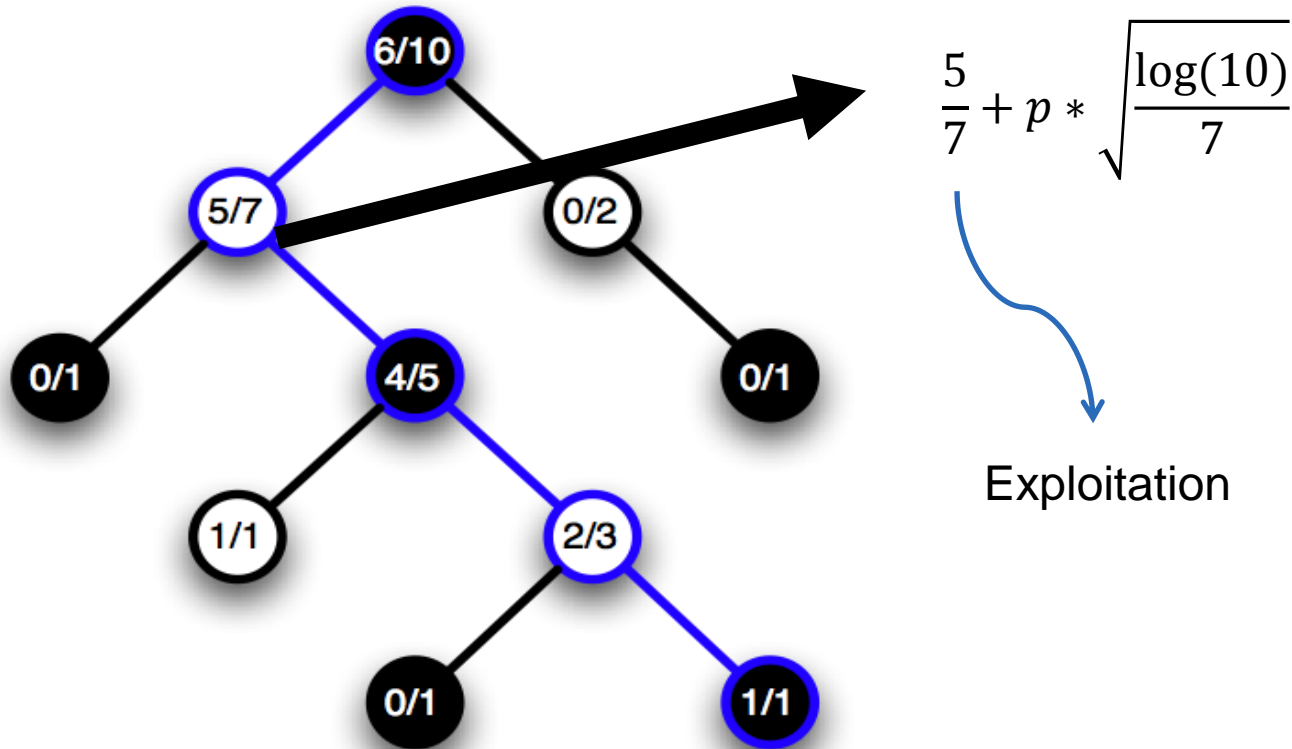
MONTE-CARLO TREE SEARCH (MCTS)



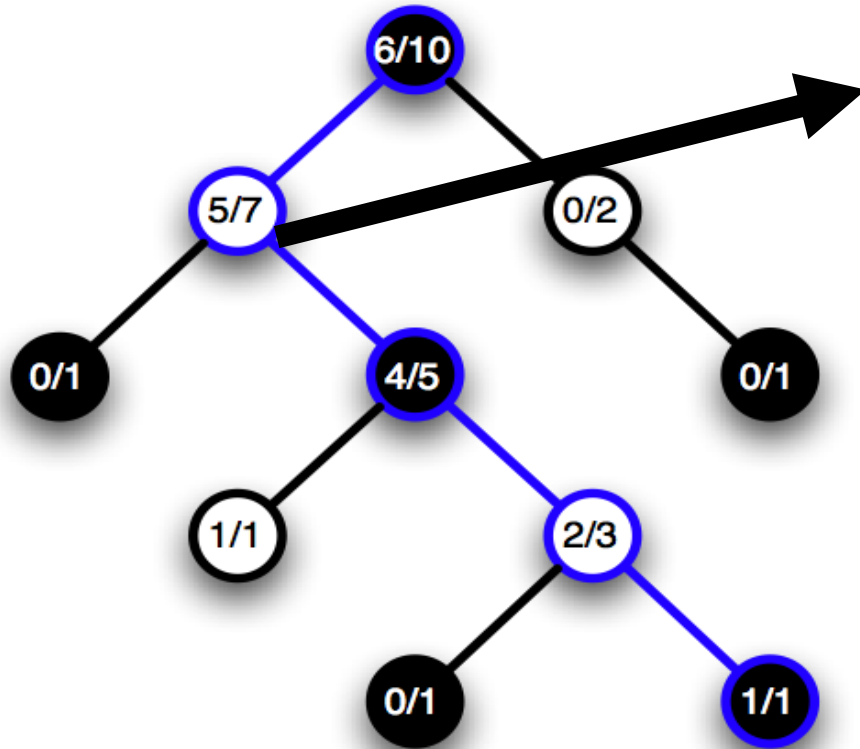
MONTE-CARLO TREE SEARCH (MCTS)

- Tree policy :
 - UCB1 formula [Auer et al., 02]
 - play arm i that maximizes $\hat{X}_i + p \sqrt{\frac{\log T}{T_i}}$
 - \hat{X}_i : Empirical average reward for move i
 - T : Total number of trials
 - T_i : Number of trials for move i
- Default policy :
 - Monte-Carlo simulation
 - → Random choice until the end

MONTE-CARLO TREE SEARCH (MCTS)



Monte-Carlo Tree Search (MCTS)



$$\frac{5}{7} + p * \sqrt{\frac{\log(10)}{7}}$$

Exploration

MONTE-CARLO TREE SEARCH (MCTS)

➤ Pros

- Efficient
- Evaluation function not needed
- Generic
- Anytime

➤ Cons

- Can we do better than pure Monte-Carlo ?



Improving the default policy

OUTLINE

- Evolutionary Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- **Multistage Optimization**
 - Problem
 - State Of The Art Algorithms
 - **Contributions**
- Conclusion

CONTRIBUTIONS

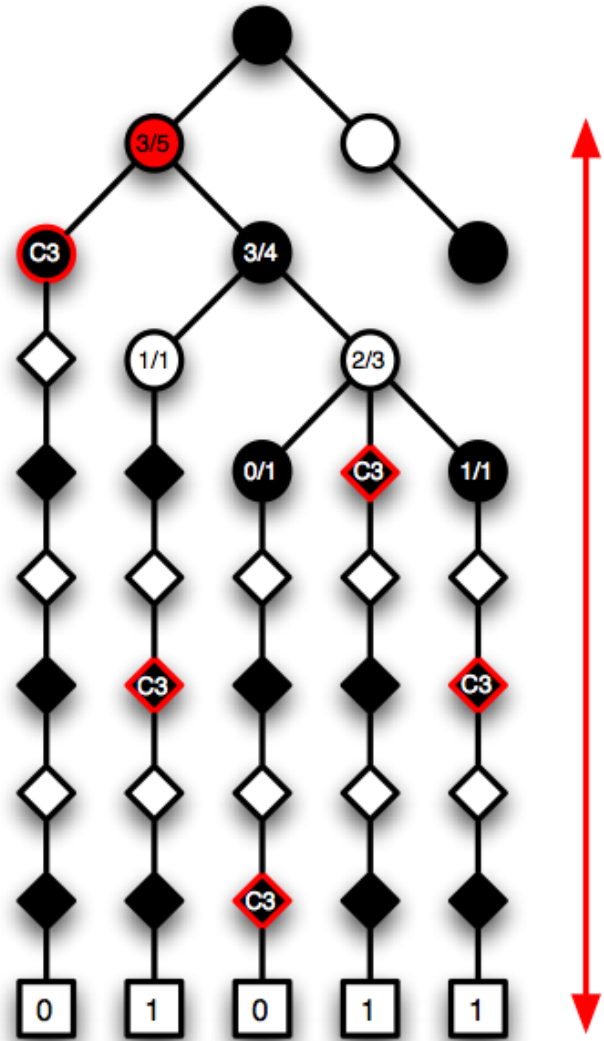
- 3 generic rules
 - poolRave
 - [ACG 2011]
 - Contextual Monte-Carlo
 - [EvoGames 2010]
 - Decisive moves and anti-decisive moves
 - [CIG 2010]

RAPID ACTION VALUE ESTIMATE (RAVE)

[GELLY & SYLVER,07]

- Keep for each node n and each move i :
 - the number of wins and losses where i has been played after n
- Compute a score $V_{\text{RAVE}}(i)$:
 - empirical score when i has been played after n
- RAVE scores are biased but have a small variance

RAVE



POOLRAVE

- Based on Rapid Action Value Estimation (RAVE)
- Use RAVE values for biasing the next Monte-Carlo simulations
 - Compute a pool of good moves according to RAVE
 - When a decision has to be made in the default policy,
 - Play a move in the pool with probability p
 - Play a random move with probability $1-p$

CONTEXTUAL MONTE-CARLO (CMC)

- Keep for each node n and for each moves A and B :
 - The number of wins and losses where A and B have been played after n by the current player
- Compute a score $V_{\text{CMC}}(A,B)$:
 - empirical score when A and B have been played after n by the current player
- If B is the last move played by the current player
- Find the move A which maximizes $V_{\text{CMC}}(A,B)$
 - Play A with probability p
 - Play a random move with probability $1-p$

DECISIVE MOVES AND ANTI-DECISIVE MOVES

- Decisive moves (exist in many games):
 - If you have a winning move, play it.
- Anti-decisive moves (exist if you can cancel a winning move of the opponent only by playing it):
 - If your opponent has a winning move and you have none, play its winning move (in order to avoid a loss).

RESULTS ON THE GAME OF HAVANNAH

Improvement	# of simulations	Best score against the baseline
poolRave	1000	$54.32 \pm 0.46\%$
	10000	$54.45 \pm 0.75\%$
	20000	$54.42 \pm 0.89\%$
CMC	1000	57%
DM	1000	$95.9 \pm 1.5\%$
DM+ADM	1000	$85 \pm 1.4\%$

DISCUSSION

- All rules are generic.
- A small but significant improvement
- All rules have been tested for the game of Havannah
- The poolRave rule works also in the game of Go (MoGo):
 - Without expert knowledge : 62.7%
 - With expert knowledge : 51.7%
- Unfortunately when the number of simulations increases, the effectiveness of DM and ADM is smaller
 - 1 second per move : 80%
 - 30 seconds per move : 50%

OUTLINE

- Evolutionary Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- Multistage Optimization
 - Problem
 - State Of The Art Algorithms
 - Contributions
- **Conclusion**

CONCLUSION AND PERSPECTIVES

➤ **Evolutionary Optimization**

- Clear improvements for efficient use of large number of processors
- Test on more algorithms and on more functions
- Hybridization with speculative parallelization [Gardner et al, 10]

➤ **Multistage optimization**

- Generic improvements of the default policy
- Comparison with (recent) existing methods :
 - Last Reply [Drake, 09]
 - N-grams [Stankiewicz et al, 11]
- Propagate decisive moves

➤ **Not presented works**

- Nested algorithm for solving TSPTW [EvoTranslog 2011]
- A cognitive science perspective (random positions in Go, blind Go) [CIG 2011]
- Using QR restarts and decreasing σ for multimodal optimization [EA 2011]

Thank you