States based Evolutionary Algorithm

Self* 2010 Workshop

Sébastien Verel[1] Philippe Collard[1] Manuel Clergue[1]

http://www.i3s.unice.fr/~verel
[1]I3S Laboratory - University of Nice-Sophia Antipolis / CNRS (France)

September 11, 2010

Position of the work

- One EA key point : exploitation/exploration tradeoff
- One main practicle difficulty : choose operators, value of parameters, and representation.
- Parameters setting (Lobo et al. 2007) :
  - Off-line before the run : *parameter tuning*,
  - On-line during the run : *parameter control*.

Position of the work

- One EA key point : exploitation/exploration tradeoff
- One main practicle difficulty : choose operators, value of parameters, and representation.
- Parameters setting (Lobo et al. 2007) :
  - Off-line before the run : *parameter tuning*,
  - On-line during the run : *parameter control*.

States based Evolutionary Algorithm (SEA)

- Parameter control method : during the run by taking into account the current state of the search.

One previous work : Dynamic Multi Arms Bandit (DMAB)

Adaptive Operator Selection

- Autonomously select the operators to be applied amongst available ones
- Dynamic Multi Arm Bandit
  A. Fialho, L. Da Costa, M. Schoenauer, M. Sebag,
  [GECCO'08] [PPSN'08] [LION'09] [GECCO'09], etc.

One previous work : Dynamic Multi Arms Bandit (DMAB)

Adaptive Operator Selection

- Autonomously select the operators to be applied amongst available ones
- Dynamic Multi Arm Bandit
  A. Fialho, L. Da Costa, M. Schoenauer, M. Sebag,
  [GECCO'08] [PPSN'08] [LION'09] [GECCO'09], etc.

Let's have a look on the slides of LION...

Principle of SEA

- n evolutionary algorithms :

$$\forall i \in [1, n],\ EA_i : 2^{\mathcal{S}} \rightarrow 2^{\mathcal{S}}$$

- Each EA has his own representation, selection operator, variation operators, and parameter settings.
- **State of a solution** s : EA to be applied on it

$$state(s) = i \in [1, n]$$

- Fitness does not change when the state changes :

$$\forall s,\ f_i(s) = f_j(stateMutation_{ij}(s))$$

Principle of SEA

- n evolutionary algorithms :

$$\forall i \in [1, n],\ EA_i : 2^{\mathcal{S}} \rightarrow 2^{\mathcal{S}}$$

- Each EA has his own representation, selection operator, variation operators, and parameter settings.
- **State of a solution** s : EA to be applied on it

$$state(s) = i \in [1, n]$$

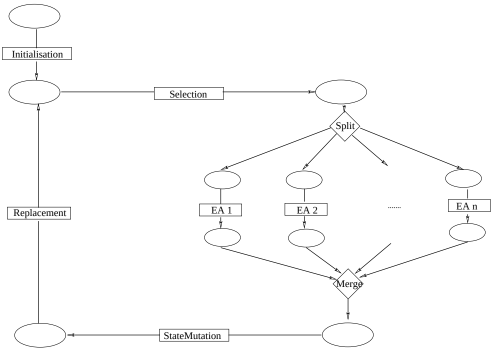
- Fitness does not change when the state changes :

$$\forall s,\ f_i(s) = f_j(stateMutation_{ij}(s))$$

Main principle

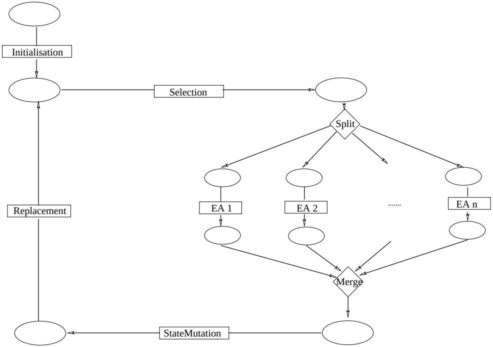
- Control the population size of each EA_i according to the fitness values of the actual solutions

Flow chart



- Initialization of solutions with their state,
- *Selection* : standart ones, not according to the state,
- *State mutation* : change only the state of solutions,
- *Replacement* : standart ones, not according to the state.

Flow chart



- Selection of the state is indirect
- Made only with the fitness of the actual solutions : no measure
- Indirectly, selection puts more solutions in the "right" state.

Problems and algorithms

- Two classical problems : *OneMax* and *long k-path* problems
- Comparison of performances with the exDMAB (of 2009) : $(1+\lambda)-$EA with $\lambda = 50$
- 4 mutations operators : $1/l$ bit-flip, $1-$bit, $3-$bit, and $5-$bit.

Problems and algorithms

- Two classical problems : *OneMax* and *long k-path* problems
- Comparison of performances with the exDMAB (of 2009) : $(1+\lambda)-$EA with $\lambda = 50$
- 4 mutations operators : $1/l$ bit-flip, $1-$bit, $3-$bit, and $5-$bit.

EAs

- 4 EAs with λ_i the population size of EA_i
- State $=$ mutation operator
- Selection of EAs : 1 best solution from the λ_i ones
- From this best solution : λ_i solutions created using the mutation operator
- Generational replacement
- Number of generation g for each EA

Experimental settings

States based EA used

- Total population size $\lambda = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 50$,
- *Selection* : tournament selection of size t,
- *State mutation* : changes a state at random with rate p,
- *Replacement* : generational with elitism,

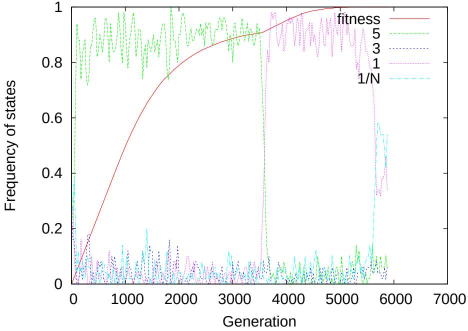
Meta-parameters of the SEA

- $t \in \{2, 3, 5\}$
- $p \in \{0.001, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.4\}$
- $g \in \{1, 5, 10, 25, 50, 100, 200, 400\}$
- Computed off-line by a complete Design Of Experiments campaign

Typical run

OneMax with $l = 10^4$

SEA with $t = 2$, $p = 0.05$ and $g = 25$



- First, 5 bit mutations : the correct one
- Then, 1 bit mutation
- Few 3 bits mutation, drawback...

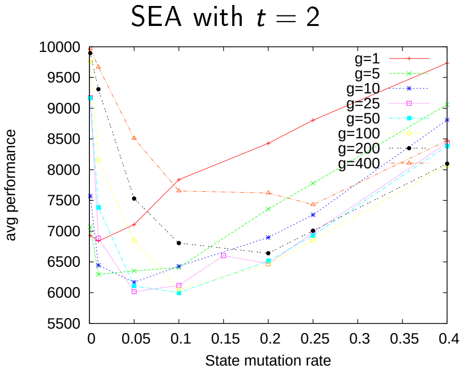
Performances

Algo.	Configuration	Gens to Opt. (avg_{sd})
SEA	$t = 2$ $p = 0.05$ $g = 25$	6076_{804}
avg-DMAB	$C = 10$ $\gamma = 25$ $W = 50$	7727_{642}
Ex-DMAB	$C = 1$ $\gamma = 250$ $W = 50$	5467_{513}
Oracle strategy		5069_{292}

- Better than avg-DMAB (2008)
- But not than ex-DMAB (2009)
- Far from the oracle stategy

Robustness of SEA against meta-parameters

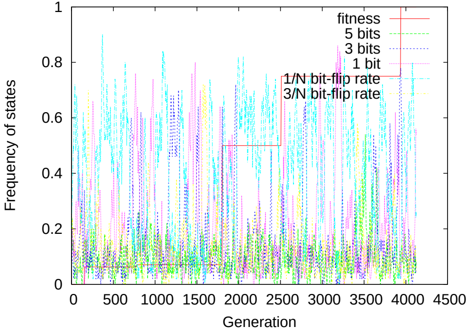
Are the choice of meta-parameters robust ?



- Performance is relatively independent of the tournament size
- Bad performances for $g \in \{1, 200, 400\}$
- State mutation rate robust for $p \in [0.05, 0.1]$
- Window of robust meta-parameters quite large

Typical run

Long 3-path problem $l = 55$
$t = 2$, $p = 0.1$ and $g = 5$



- Different from oneMax problem
- No large stage where one state dominates
- $1/l$ bit-flip mutation is the larger one during 60% of generations
- All mutations are used

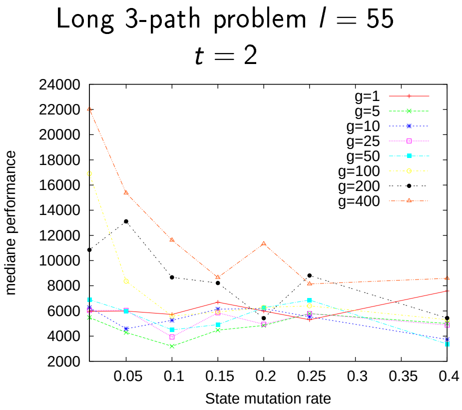
State of $3/l$ bit-flip operator is added

Performances

l	Algo.	Gens to Opt. (min – $median$)
43	SEA	20 – 3151
	Ex-DMAB	11 – 2216
	Oracle strategy	2 – 1202
49	SEA	5 – 3755
	Ex-DMAB	17 – 3244
	Oracle strategy	19 – 2668
55	SEA	51 – 4126
	Ex-DMAB	161 – 6190
	Oracle strategy	45 – 3224
61	SEA	59 – 7950
	Ex-DMAB	80 – 10253
	Oracle strategy	8 – 5408

Robustness of SEA against meta-parameters

Are the choice of meta-parameters robust ?



- Similar performances for $t \in \{3, 5\}$
- Good performances for $g \leq 100$, $g = 1$ can be used
- State mutation rate more robust for long path problem
- Meta-parameters more robust

Discussion

- No measure of performances on the operator needed (evolvability or diversity) :
  Directly the distribution of fitness of solutions
- Manages two kinds of dynamics :
  evolve the solutions / evolve the parameters
- Crucial point : well adjust the two dynamics

Discussion

- No measure of performances on the operator needed (evolvability or diversity) :
  Directly the distribution of fitness of solutions
- Manages two kinds of dynamics :
  evolve the solutions / evolve the parameters
- Crucial point : well adjust the two dynamics
- Difference with AOS :
  - In AOS : one operator used at the same time,
    controls the number of generations for each operator
  - In SEA : several EAs at the same time,
    controls the sub-population size of each EA
- SEA is more parallel / AOS more sequential

Conclusions and Futur Works

- State based EA : general framework
- Not competive on oneMax, competive on long path problem

Conclusions and Futur Works

- State based EA : general framework
- Not competive on oneMax, competive on long path problem

Future works

- Performances : tradeoff between the exploitation and the exploration components
  Theoretical works must be done to analyse (and coordinate) this tradeoff (in progress)
- New experiments on others problems :
  study the performance of the SEA on **multimodal problems**
- **Compare** the SEA with the extreme Compass DMAB [Maturana et al. 09]
- Design a **distributed version** of SEA

Thanks to Alvaro Fialho for his help on DMAB