

Parameter Control in Evolutionary Algorithms by Domain-specific Scripting Language PPC_{EA}

Shih-Hsi Liu¹
Marjan Mernik²
Barrett R. Bryant¹



¹ Softcom Laboratory, University of Alabama at Birmingham, USA

² Center for Computer Languages, University of Maribor, Slovenia



Outline

- Motivation
- Related Work
- PPC_{EA}
- Experimental Results
- Conclusion





Motivation

- Control parameters (e.g.; probability of crossover, probability of mutation, population size, selection pressure) have significant impact on Evolutionary Algorithm's performance.
- EA user dilemma: what control parameter settings are likely to produce the best results?
- Trial and error approach is time consuming and tedious task.
- Different values might be optimal at different stages of the evolutionary process.



Outline

- Motivation
- Related Work
- PPCEA
- Experimental Results
- Conclusion





Related Work (1/4)

- Empirical studies
 - De Jong (pop_size = 50, $p_m=0.001$, $p_c=0.6$)
 - Grefenstette (pop_size = 30, $p_m=0.01$, $p_c=0.95$)
 - Schaffer ($p_m \in [0.005, 0.01]$, $p_c \in [0.75, 0.95]$)
- Theoretical studies
 - Mühlenbein ($p_m=1/L$)
 - Hesser&Männer ($p_m(t)=\sqrt{\alpha/\beta} \cdot \exp(-\gamma \cdot t/2)/\text{pop_size} \cdot \sqrt{L}$)
 - Harik (pop_size = $-2^k \cdot \sqrt{\pi m \sigma^2} / 2d \cdot \ln \alpha$)
- Specific problems require specific values of control parameters and general near-optimal control parameter setting is not appropriate.



Related Work (2/4)

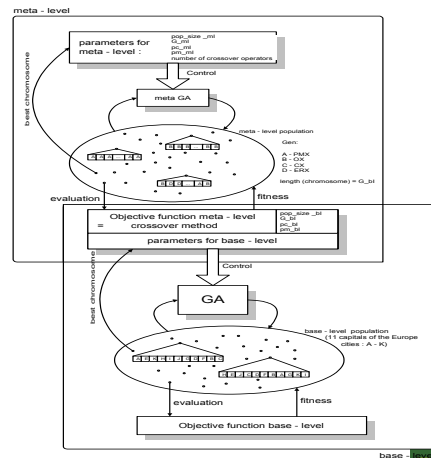
- specific problem-oriented: e.g. GAVaPS
- Parameter Tuning
- Parameter Control (Eiben, Hinterding, Michalewicz)
 - deterministic: deterministic rule
 - adaptive: adapted by feedback of EA
 - self-adaptive: parameters are encoded into chromosomes





Related Work (3/4)

- Meta-evolutionary approach



Related Work (4/4)

- Parameter-less approach (Harik & Lobo)
 - users are relieved from control parameter setting
 - peak performance can not be achieved
 - based on Schema Theorem
 - $s = 4$, $p_c = 0.5$
 - pop_size is eliminated by iteratively running the GA with different population size, doubling the population size each time population converges or by establishing a race among populations of various sizes.





Outline

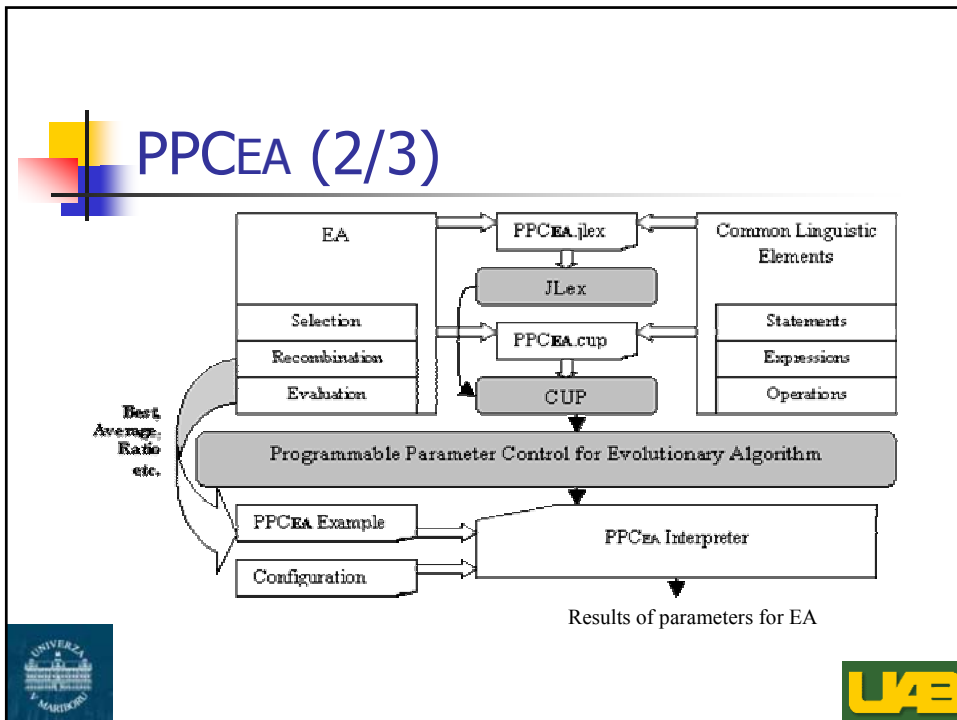
- Motivation
- Related Work
- PPCEA
- Experimental Results
- Conclusion



Programmable Parameter Control for Evolutionary Algorithms – PPCEA (1/3)

- Based on the principles of compiler techniques
 - JLex: lexical analyzer generator for Java
 - CUP: parser generator for Java
 - defines syntax and semantics of PPCEA
 - defines how PPCEA interpreter execute its source code
- A domain-specific scripting language
 - common linguistic elements
e.g. if-then-else, while-loop
 - domain-specific statement:
e.g. init, call_EA, writeresult





- ## PPCEA (3/3)
- Characteristics of PPCEA
 - hide implementation details of EA:
 - a) "call_EA" executes selection, recombination and evaluation
 - b) "init" executes initialization of a population
 - runtime parameter configuration checking
 - i.e., PPCEA checks if predefined control parameter settings are reasonable
 - **Finally but most importantly:**
 - PPCEA interpreter executes/interprets EA's parameters at runtime => Parameters can be modified easily and dynamically at runtime
- UNIVERSITAT DE VALÈNCIA
- UVB



Outline

- Motivation
- Related Work
- PPCEA
- Experimental Results
- Conclusion



Experimental Results (1/9)

- Algebraic fitness function

$$f(x,y) = 0.5 + \frac{\sin^2(x^2 + y^2)^{1/2} - 0.5}{(1+0.0001*(x^2 + y^2))^2}$$

where $-100 \leq x, y \leq 100$

- Initial parameters (in most cases)
 - Maxgen = 500 - Epoch = 10
 - Popsiz = 50 - Plover = 0.7
 - Pmutation = 0.1





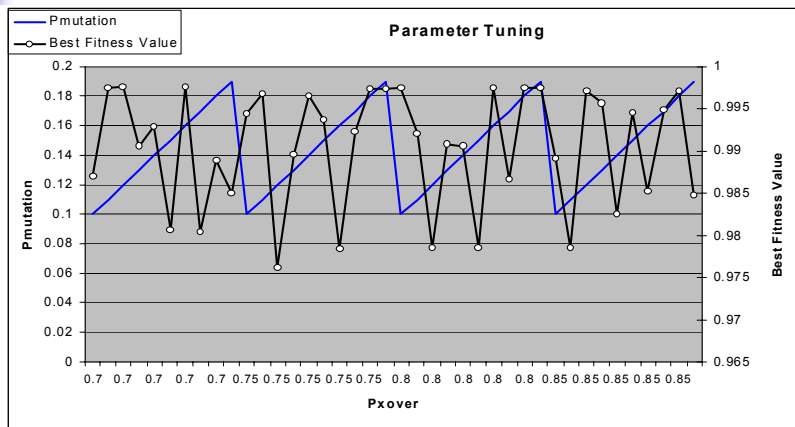
Experimental Results (2/9)

■ Parameter Tuning

```
while (Pxoever <= 0.9) do
  Pmutation := 0.1;
  while (Pmutation <= 0.2) do
    init; // initialize population
    call_EA; // run EA for one epoch
    writeresult;
    Pmutation := Pmutation + 0.01
  end;
  Pxoever := Pxoever + 0.05
end
```



Experimental Results (3/9)





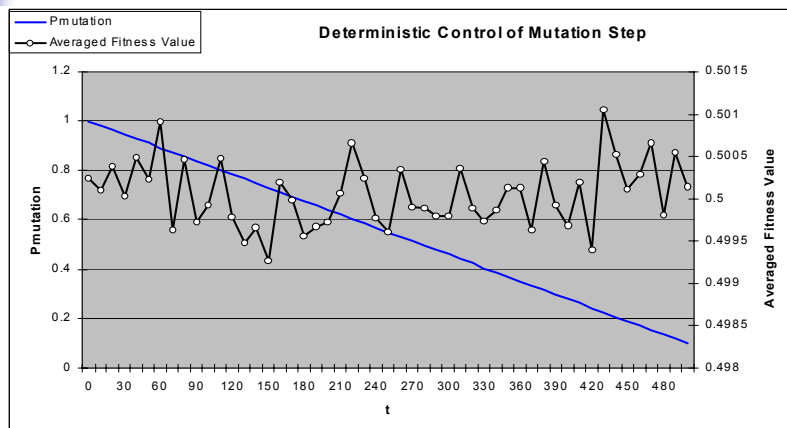
Experimental Results (4/9)

- Deterministic control of mutation step

```
init;  
while (t <= Maxgen) do  
  Pmutation := 1 - (0.9 * t)/Maxgen;  
  call_EA; // run EA for one epoch  
  writeresult;  
  t := t + Epoch  
end
```



Experimental Results (5/9)





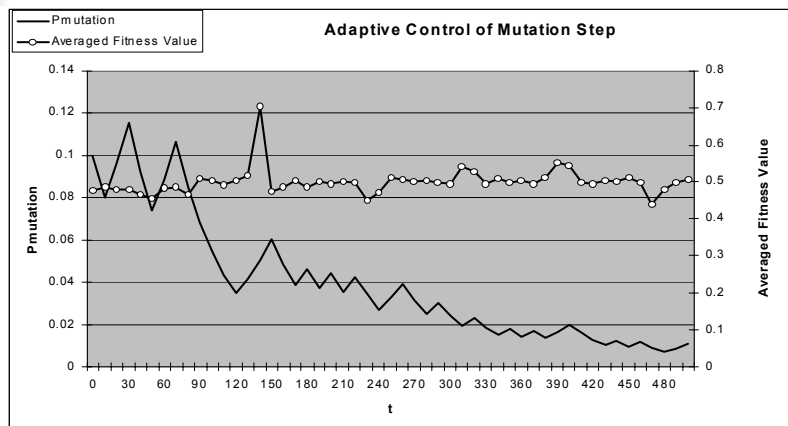
Experimental Results (6/9)

■ Adaptive control of mutation step

```
init;  
while (t <= Maxgen) do  
  call_EA;  
  writeresult;  
  if (Ratio > 0.2) then  
    Pmutation := Pmutation * 1.2  
  else  
    Pmutation := Pmutation * 0.8  
  fi;  
  t := t + Epoch  
end
```



Experimental Results (7/9)





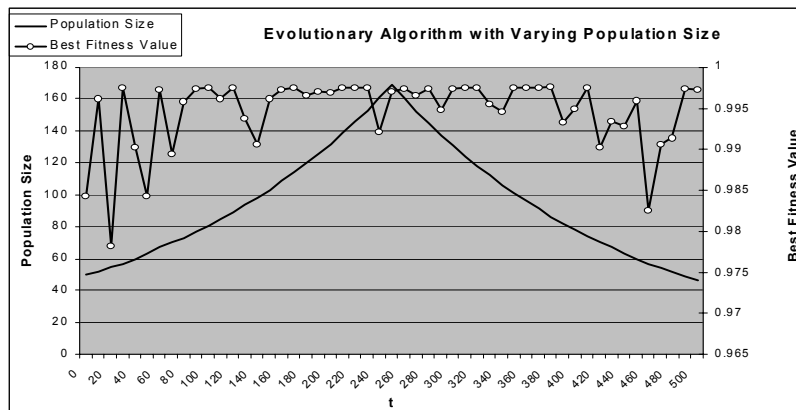
Experimental Results (8/9)

■ Varying population size

```
init;  
while (t <= Maxgen) do  
  call_EA;  
  writeresult;  
  if (Best < Average*Maxgen/(t+1)) then  
    Popsiz := Popsiz * 1.05  
  else  
    Popsiz := Popsiz * 0.95  
  fi;  
  t := t + Epoch  
end
```



Experimental Results (9/9)





Outline

- Motivation
- Related Work
- PPCEA
- Experimental Results
- Conclusion



Conclusion

- The main advantage of our approach is that the problem of control parameters is lifted to the higher abstraction layer while keeping EA simple as before.
- All control parameters are treated uniformly, hence achieving a good flexibility of the proposed approach.





Thank you
Questions?

