CULTURAL ALGORITHMS: A TUTORIAL

1 author:

Robert G. Reynolds
Wayne State University

CULTURAL ALGORITHMS: A TUTORIAL

DR. ROBERT G. REYNOLDS
WAYNE STATE UNIVERSITY
DETROIT, MICHIGAN

OUTLINE

- I. Ideational Theories of Cultural Evolution
- II. Cultural Algorithms: A Computational Framework
- III. General Features
- IV. Suitable Problems
- V. Designing Cultural Algorithms
- Embedding a weak method into the Cultural Algorithm Framework: A Genetic Algorithm Example
- IV. Example Applications
- V. Future Directions

Ideational Approaches to Cultural Evolution

- Edward B. Tylor was the first to introduce the term “Culture” in his two volume book on *Primitive Culture* in 1881.
- He described culture as “that complex whole which includes knowledge, belief, art, morals, customs, and any other capabilities and habits acquired by man as a member of society”.
- Early approaches to studying culture focused on classification of cultures worldwide into groups based upon “adhesions” between cultural elements.
- George Murdoch (1957) produced a “catalog” of 565 cultures based upon 30 sample characteristics.
- Research in Cybernetics and Systems Theory in 1960’s spawned new views of culture as a system that interacted with its environment. It provided regulatory mechanisms that provide positive and negative feedback that can respectively amplify and counteract behavioral deviations of individuals within a cultural group. Flannery 1968.

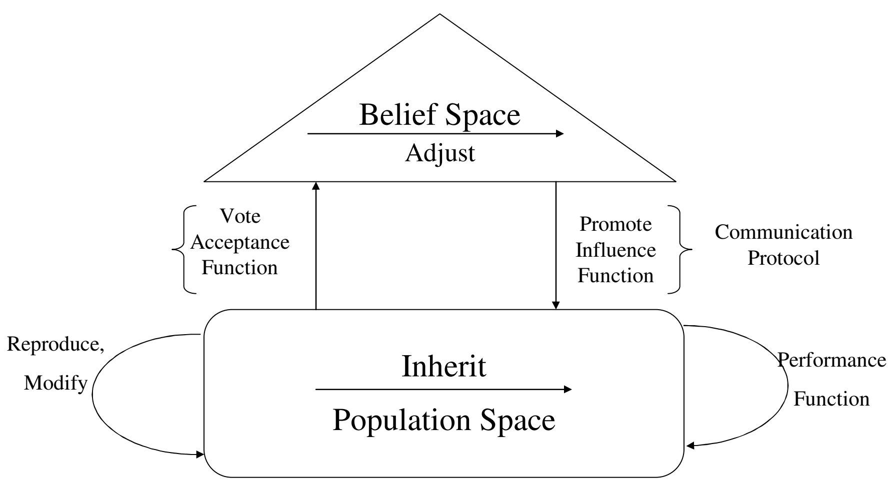
Ideational Approaches Continued

- In the 1960’s Cultural Ecology emerged as a discipline concerned with the nature of the interactions between the cultural system and its environment.
- In the 1970’s saw a new emphasis on how culture shaped the flow of information in a system, a generalization of the cultural ecology perspective.
- Geertz (1973)“Culture is the fabric of meaning in terms of which human beings interpret their experience and guide their actions.
- Durham(1990)“Culture is shared ideational phenomena (values, ideas, beliefs, and the like)”. Less purposeful.

CULTURAL ALGORITHMS ARE COMPUTATIONAL MODLES OF CULTURAL EVOLUTION

BASIC PSEUDOCODE FOR CURTURAL ALGORITHMS IS A AS FOLLOWS:

```
Begin
  t = 0;
  Initialize Population POP(t);
  Initialize Belief Space BLF(t);
   repeat
       Evaluate Population POP(t);
       Adjust(BLF(t), Accept(POP(t)));
       Adjust(BLF(t));
       Variation(POP(t) from POP(t-1));
   until termination condition achieved
End
```



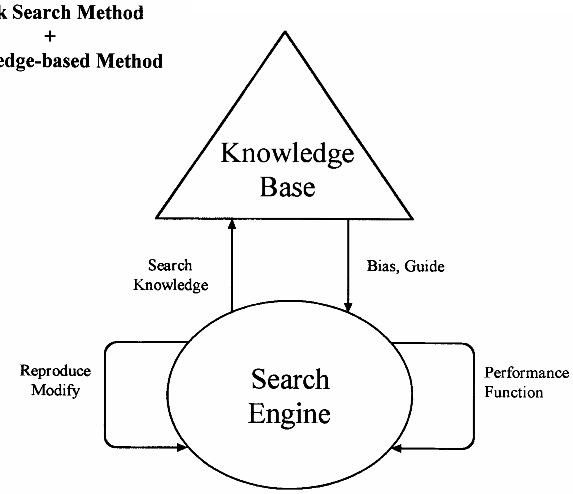
The cultural algorithm components consists of a belief space and a population space. The components interacts through a communication protocol

General Features

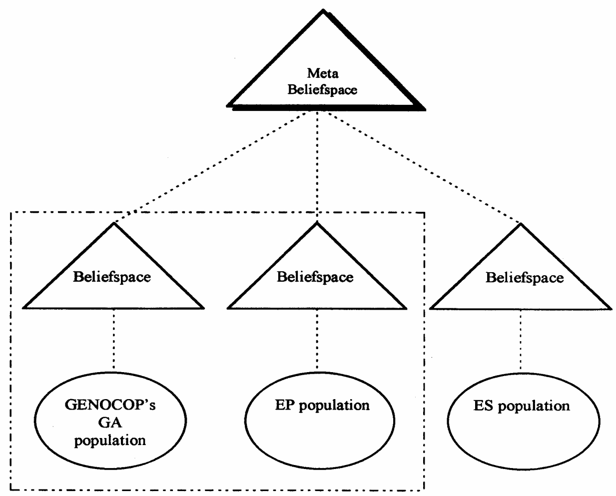
- Dual Inheritance (at population and knowledge levels)
- Knowledge are “beacons” that guide evolution of the population
- Supports hierarchical structuring of population and belief spaces.
- Domain knowledge separated from individuals(e.g. ontologies)
- Supports self adaptation at various levels
- Evolution can take place at different rates at different levels (“Culture evolves 10 times faster than the biological component”).
- Supports hybrid approaches to problem solving.
- A computational framework within which many all of the different models of cultural change can be expressed.

Hybrid System:

Weak Search Method
+
Knowledge-based Method



Can support the emergence of hierarchical structures in both the belief and population spaces



Suitable Problems

- Significant amount of domain knowledge (e.g. constrained optimization problems).
- Complex Systems where adaptation can take place at various levels at various rates in the population and belief space.
- Knowledge is in different forms and needs to be reasoned about in different ways.
- Hybrid systems that require a combination of search and knowledge based frameworks.
- Problem solution requires multiple populations and multiple belief spaces and their interaction.
- Hierarchically structured problem environments where hierarchically structured population and knowledge elements can emerge.

II. Designing Cultural Algorithms

- 1. Design of the knowledge component
- A. Ontological knowledge (shared common concepts for a domain) representation
- B. Constraint knowledge representation
- C. Solution representation
- D. Which will be modified? Update function for each modifiable component.
- E. Knowledge Maintenance
- 2. Design of the Population Component
- A. State variables that determine solution behavior
- B. How those variables are used to produce a problem solving strategy or behavior.
- C. How such behavior is evaluated?

Designing Cultural Algorithms: Embedding a Weak Method

- Use Genetic Algorithms as an example population model. Show how it can be embedded in the Cultural Framework for a sequence of increasingly complex problems.
- Whether you begin with the belief level or the population level depends on the problem. That is, which of the two is more constrained by the problem?
- Classification Problems Vs. Construction Problems. With former often start with the belief space, with the latter the population space. In real world situations may have both, select the most constrained of the two.
- In either case, iterate between the two adding detail as you go.

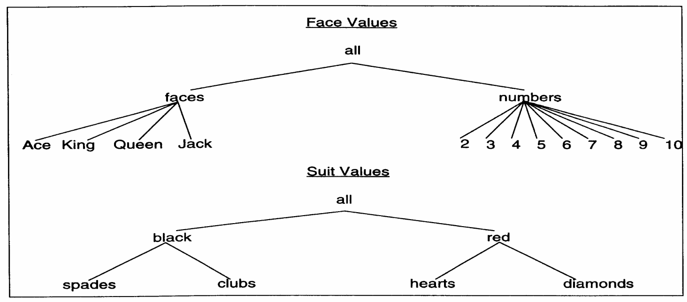
The Genetic Algorithm(Davis,1991)

- 1. Initialize a population of chromosomes
- 2. Evaluate each chromosome in the population
- 3. Create new chromosomes by mating current chromosomes: apply mutation and crossover as the parent chromosomes mate.
- 4. Delete members of the population to make room for the new chromosomes.
- 5. Evaluate the new chromosomes and insert them into the population.
- 6. If time is up, stop and return the best chromosome; if not go to 3.

A Classification Problem

- Mastermind problem.
- Guess the set of objects that the oracle has in mind.
- Can only get information about whether a specific object is included or not.
- Card Problem.

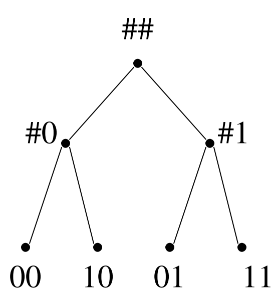
Cards are divided into two independent categories: suit and face.



Based upon this a possible population is
[Suit | Face]
Generate examples at random
Accept all examples
No influence (scorecard) until termination
Update using Mitchells Candidate Elimination Alg.
Focus on Suit {all=##, b=#0, r=#1,s=00,c=10,h=01,d=11}

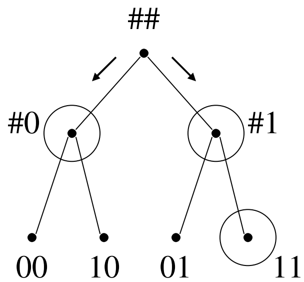
Static Version Spaces

- Use Mitchells candidate elimination search procedure



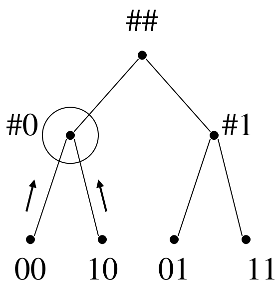
G set = { # # }

S set = { }



Negative examples pushes down G set

G set = { #0, #1 }



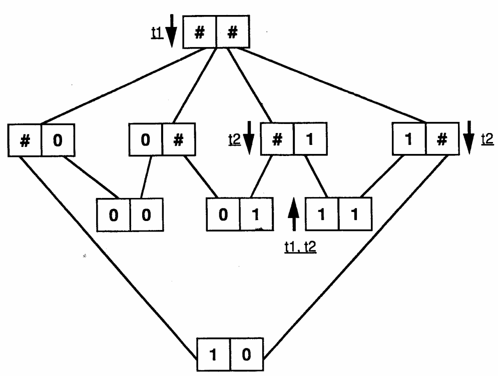
Positive examples push up

S set = { 00, 10 }

G set = { #0, #1 }

S set = { #0 }

If an individual observes another individual, information is recorded in the graph.



Individual observed: $\boxed{0\,|\,0}$ **Negative**

($\boxed{\#\,|\,\#}$ **) = ↓ at t1** **(** $\boxed{\#\,|\,1}$, $\boxed{1\,|\,\#}$ **) = ↓ at t2** **(** $\boxed{1\,|\,1}$ **) = ↑ at t2 = ↑ at t1**

Classification Example

- Generalize on positive examples and specialize with negative examples. When the arrows overlap then a maximally specific concept is identified. The most general concept or set description that is consistent with the negative examples.
- Here factored the space into two independent subspaces. Information about guesses is used to update each space independently.
- Then select a population representation to generate the guesses.
- Suit|Card Suit = {club,spades, hearts, diamonds} Card = {2,..J,Q,K,A}
- Performance function = oracle {right, or wrong}
- Acceptance function all guesses made this generation.
- Influence Function, generate only guesses consistent with the current S and G sets.
- Reproduction and modification, mutate each parent to values within within the intersection of the S and G sets.

A Construction Problem

- In a construction problems the state variables are often not independent.
- This means that the lattice may not be easily factored into sub-lattices and updated in parallel. Theoretically all parameter values can be used to organize the set.
- The fan-out at a given level can be an exponential function of the problem size in the worst case.
- Can also be multiple solutions.
- Add operations in the belief space to compensate.
- E.G. Merge , and stable classes. Can prove properties about the operators (e.g. merge does not lose information Sverdlik)

Boole Problem:

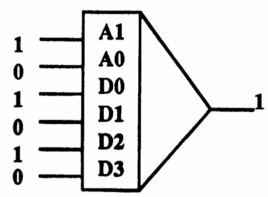
Infer the characteristic function for a unknown boolean multiplexer.

Example:

Characteristic function:

$F6 = A'0A'1D0 + A0A'1D1 + A'0A1D2 + A0A1D3.$

For F6 (2 address lines, 4 data lines).

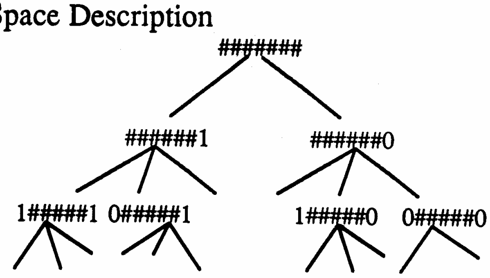


Problem Representation

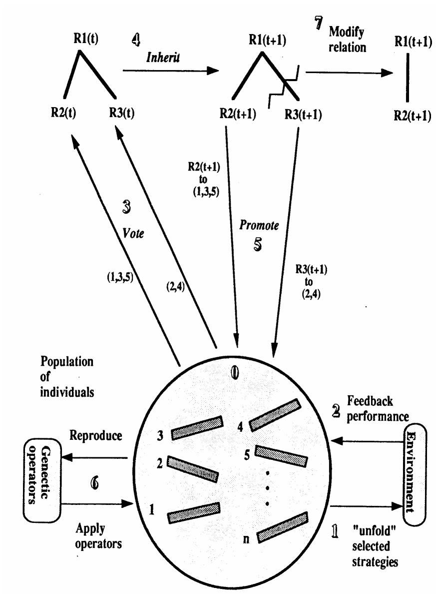
Chromosome Description

A1	A0	D0	D1	D2	D3	F6
1	0	1	0	1	0	1

Version Space Description

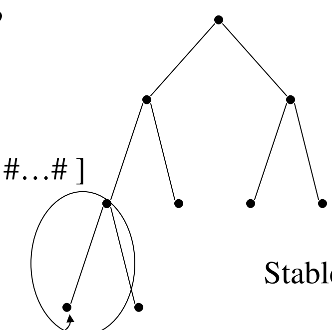


Schematic Description of Cultural Algorithm



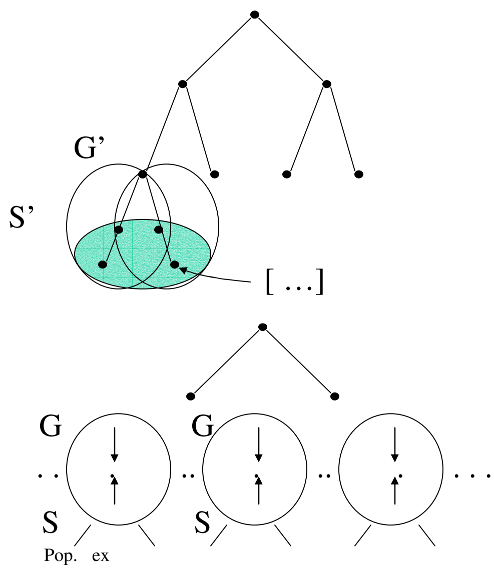
• VIP Protocol interconnects the biological and cultural components

"Segmentation"



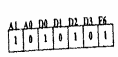
• Generating a homogeneous region with respect to the acceptance function.

“Merging”

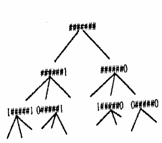


- Maximally Specific Generalization

TRAIT POPULATION SPACE



BELIEF SPACE



F6

Positive Instances	Stable Schema
(a) 111111	##1###
(b) 111110	##1###
(c) 111010	#11###
(d) 111011	#11###

A Stable Class is comprised of:

1. G set
2. S set
3. Population

Initially form stable classes from individual stable schema

- The G set of an instance is the stable class
- The S set of an instance is the instance
- The population of an instance is the instance

EXAMPLE: For the instance (a)

Sa.Gset = ##1###
Sa.Sset = 111111
Sa.Pop = {111111}

Stable classes are combined in 2 steps:

1.
Stable classes Sx and Sy are combined IFF
Sx.Gset = Sy.Gset

If Sz is the resultant stable class, then:

Sz.Gset = Sx.Gset = Sy.Gset
Sz.Sset = Generalize(Sx. Sset, Sy.Sset)
Sz.Pop = Sx.Pop ∪ Sy.Pop

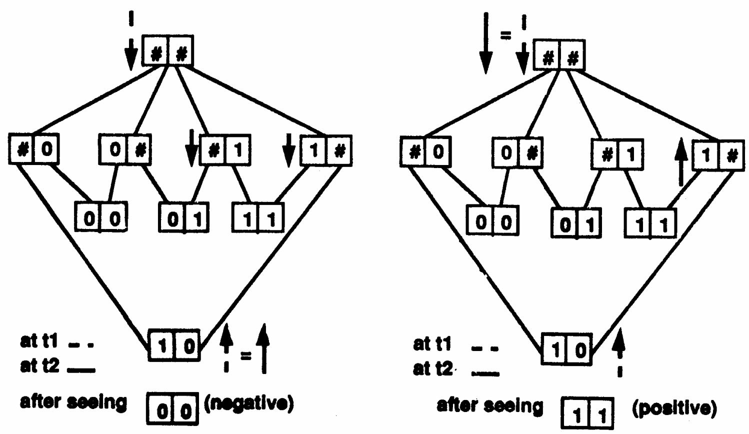
From previous example

Sa and Sb may be combined, as well as Sc and Sd.

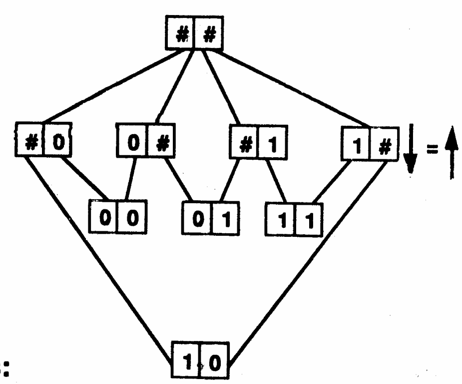
Sab.Gset = ##1###
Sab.Sset = 11111#
Sab.Pop = {111111, 111110}

Scd.Gset = #11###
Scd.Sset = 11101#
Scd.Pop = {111010, 111011}

Schema can be merged to share experiences. This can produce group schema.



merge produces:



- **By making the relations between schemata explicit one can exploit nested collections of high performance**
- **E.G. Clustering of successful cases in circuit design problem [Louis et al., FLAIRS 92]**

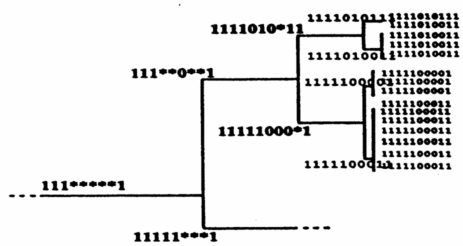


Figure 3: A closer look at the clustered cases reveals nested schemata.

- **Cultural Algorithms can exploit collections of nested schemata which is necessary when dealing with complex non-linear systems.**

VGA Symbiosis

The Version Spaces approach is now feasible for large problems. Since example generation is done automatically by the GA. THe Version Space guides the generation process using the VIP relation.

Schema Theorem:

$$m(H,t+1) >= m(H,t)\frac{f(H)}{\bar{f}}\left[1 - p_c \frac{dlen(H)}{len - 1} - p_m\, o(H)\right]$$

- The presence of the version space allows the GA system to retain experience outside of its own knowledge base and explore the space at a high rate, even in localized search.
- In addition, the population size needed can be reduced markedly.
- Interpretation of the results can be done at "high level", relative to accepted hypotheses in the version space.

Hyperschema Theorem:

$$m(H,t+1 \mid HS \in PATHS(H,t+1)) >=$$
$$m(H,t \mid HS \in PATHS(H,t)) \times$$
$$\frac{avg(f''(H,t) \mid HS \in PATHS(H,t))}{\overline{f''}} \times$$
$$\frac{avg(f'(H,t) \mid HS \in PATHS(H,t))}{\overline{f'}} \times$$
$$[\, 1 - [\, p_m \times avg(o(H) \mid HS \in PATHS(H,t)] -$$
$$\left[p_c \times \frac{avg(dlen(H) \mid HS \in PATHS(H,t))}{len - 1} \right] \,]$$

Comparison of VGA on Boole with other systems.

Wilson's Boole Classifier System (1988).

Leaning Task	Number of Instances Seen		Accuracy of Test Results	
	Boole	SVGA	Boole	SVGA
F_6	15,000	1500	97.3%	100 %
F_{11}	30,000	3920	97.5%	100 %

Quinlan's C4 System (1988).

Learning Tasks	Training Set (C_4)	Initial Population (SVGA)	Accuracy of Test Results	
			C_4	SVGA
F_6	50	48	85.1%	90.91%
F_{11}	200	220	98.3%	100%

Performance as a function of Genetic Operator Probability.

Mutation.

Probability of Mutation	Average Number of Reproductions	Marginal Accuracy of the Test Result
0.1	16.8	96.4%
0.2	14.8	100.0%
0.3	13.2	98.7%

Crossover.

Probability of Crossover	Average Number of Reproductions	Marginal Accuracy of The Test Result
0.2	16	91.95%
0.5	16.8	96.38%
0.8	15.2	90.24%

Experimental Results for F_6 as a function of population size.

Initial Population Size	Average Number of Reproductions	Average Number of Patterns in Following Sets			CPU Time in Seconds	Marginal Accuracy of the Test Results
		Solution	Overlapping	Incorrect		
12	35.4	8	8	20.4	1.9	44.0%
24	25.0	8	8	5.8	1.8	73.4%
36	22.2	8	8	2.6	2.2	86.0%
48	18.6	8	8	1.6	2.5	90.9%
60	19.0	8	8	0.4	2.9	97.6%
72	17.2	8	8	0.4	3.1	97.6%
84	14.8	8	8	0.4	4.0	97.6%
96	13.0	8	8	0.4	4.2	97.6%
108	13.4	8	8	0.4	4.3	97.6%
120	12.6	8	8	0.0	5.3	100%

Experimental Results of F_{11} as a function of population size.

Initial Population Size	Average Number of Reproductions	Average Number of Patterns in the following sets			CPU Time in Seconds	Marginal Accuracy of the Test Results
		Solution	Overlapping	Incorrect		
22	80	16	24	9.4	2385.7	80.9%
44	48.6	15.8	24	9.6	1172.2	80.6%
66	37.0	15.8	23.8	12.6	785.3	75.9%
88	31.6	16	24	1.2	727.9	97.1%
110	26.0	16	24	0.0	725.5	100%
132	23.8	16	24	1.0	769.8	97.6%
154	22.2	16	24	0.2	695.5	99.5%
176	20.2	16	23.8	0.2	757.2	99.5%
198	19.8	16	24	0.0	791.1	100%
220	19.0	16	24	0.0	805.2	100%

Comparison:

- The VGA performs as well as C4 but does not need to generate the 200 examples by hand.
- The VGA requires an order of magnitude fewer trials to solve the problem relative to the Classifier approach.
- The VGA is much less sensitive to genetic operator probabilities which corresponds with behavior predicted by the Hyperschema Theorem.
- Therefore the attention paid to possible symbiotic relationships among components in a hybrid learning system may result in a system capable of outperforming that of its components.

Population Component

- Genetic Algorithms
- Often population model has an inherent knowledge structure associated with it.
- Genetic Algorithms exploit schemata. The VGA model described earlier is nothing more than the explicit use of binary schemata to guide the generation of examples by the Genetic Algorithm population.
- Exploits building blocks. In hierarchical problems building blocks at one level can be exploited and combined at the next level.
- Need to allow our representation scheme to emerge based upon the level of complexity achieved in the mined building blocks.

ROYAL ROAD PROBLEM

- **ROYAL ROAD FUNCTION**

```
function rr
    var
        i;               { number of target schemata }
        j;               { number of levels in hierarchy }
        n_j;             { number of target schemata found at level j }
        m_i;             { number of correct bits in a target schema }
        b;               { number of bits in a target schema }
        u, u*, v, m*;    { parameters }
        part_i;          { points for number of correct bits }
        bonus_j;         { points for correct target schemata }
        score;           { part_i + bonus_j }

begin
    for each target schema i at level 1
    begin
        if ( m_i < m* + 1 ) then
            part_i = ( m_i )v;
        else if ( m* < m_i < b ) then
            part_i = -( m_i - m* )v;
        else
            part_i = 0;
    end

    for each level j in hierarchy
    begin
        if ( n_j > 0 ) then
            bonus_j = u* + ( n_j - 1 )u;
        else
            bonus_j = 0;
    end

    score=0;
    for each target schema i at level 1
        score = score + part_i;
    for each level j in hierarchy
        score = score + bonus_j;
    return(score);
end;
```

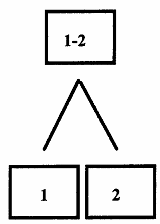
ROYAL ROAD PROBLEM

- **A SIMPLE EXAMPLE**

Parameters: i = 2, j = 2, u = .3, u* = 1, v = .02, m* = 4, b = 8

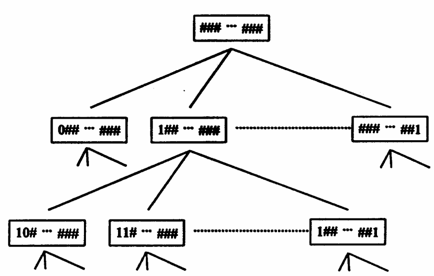
Goal: 0 0 0 0 0 0 0 0 b b 0 0 0 0 0 0 0 0

Individual$_1$:	0 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1	score = .08
Individual$_2$:	0 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1	score = -.02
Individual$_3$:	0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0	score = 2.3



PATHFINDER

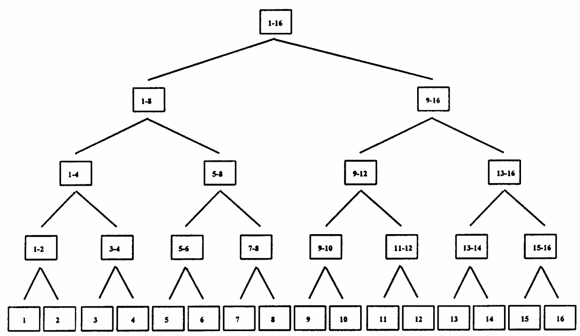
- **LOWEST LEVEL OF BELIEF SPACE**



Once we acquire building blocks at one level we can
Re-size the version space to exploit them

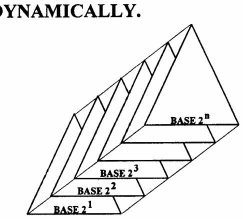
EXPERIMENTS AND RESULTS

- HIERARCHY USING HOLLAND'S SUGGESTED PARAMETERS



PATHFINDER

- MULTILEVEL BELIEF SPACE
  - IT IS POSSIBLE TO CONVERT A NUMBER FROM ONE BASE TO A DIFFERENT BASE.
  - THE REPRESENTATION SPACE IS HIERARCHICAL, AND CONSTRUCTED DYNAMICALLY.



Can move up and down the hierarchy of bases depending Upon how well two adjacent bases do.

REAL-VALUED SCHEMA IN THE BELIEF SPACE

ESCHELMAN AND SCHAEFFER PROPOSED INTERVAL SCHEMATA FOR REAL-VALUED VARIABLES.

```
                X                        Y
|--------------------------|     |----------------|
```

CAEP USED THIS AS BELIEF SPACE KNOWLEDGE TO GUIDE SEARCH USING AN EP POPULATION TO SOLVE UNCONSTRAINED REAL-VALUED FUNCTION OPTIMIZATION PROBLEMS. (CHUNG AND REYNOLDS 1994)

FOR PROBLEMS WITH LARGE BASINS AND OR VALLEYS, LESS INFORMATION WAS GAINED FROM EACH INDIVIDUAL DURING A GENERATION. FUZZY SCHEMATA USED FUZZY INTERVALS TO DIRECT SEARCH IN THESE INSTANCES (ZHU AND REYNOLDS, 1998).

```
            X                              Y
  |--------------------|          |--------------|
```

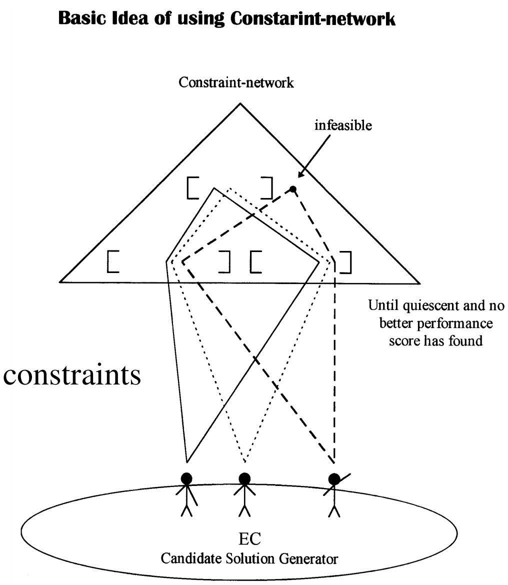
TWO BASIC TYPES OF KNOWLEDGE IN THE BELIEF SPACE:

NORMATIVE KNOWLEDGE: STANDARDS OF BEHAVIOR

(E.G. 10 > x > 2) ACCEPTABLE RANGE OF VALUES FOR PARAMTER X IN A PARAMETER OPTIMIZATION PROBLEM

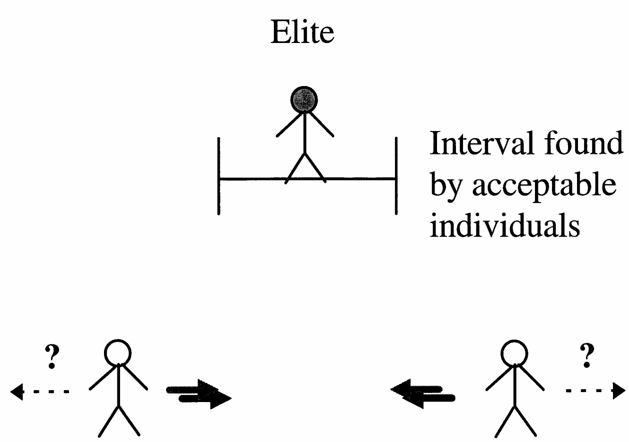
SITUATIONAL KNOWLEDGE: INDIVIDUAL EXAMPLES OF PROBLEM SOLVING SUCCESS AND OR FAILURE.

(E.G. F(0,1,0) HAS THE BEST OBSERVED PERFORMANCE SO FAR.



Domain Range constraints

Cultural Influence



	1	2	$f(x_i)$	*accept*	*update*E	*update*N
1	0.01	0.01	0.0001	1	1	1
2	0.0	0.1	0.01	1	0	1
3	-0.1	0.2	0.05	0	0	0
4	-0.11	0.22	0.0605	0	0	0
5	-0.1	0.59	0.3581	0	0	0

Figure 3.8 Individuals in a population for updating Belief Space

Figure 3.9 shows a result of adjusting situational knowledge from the population in the figure 3.8. Since the best individual has better performance value (0.0001) than that of the current exemplar, the current exemplar is replaced with the current best, <0.01, 0.01>, in the population space.

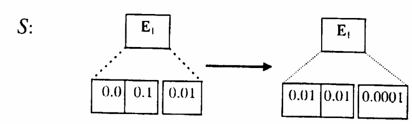


Figure 3.9 An example result of Adjusting Situational Knowledge

Figure 3.10 shows a result of adjusting normative knowledge according to the adjustment rules from the population in the figure 3.8. The top 2 individuals, <0.01, 0.01> with performance score 0.0001 and < 0.0, 0.1> with performance score 0.01 are used to adjust the current normative knowledge from the population.

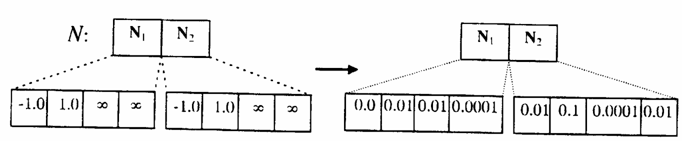


Figure 3.10 An example result of Adjusting Normative Knowledge

The individuals in figure 3.8 are then become the parents for the next generation of the CAEP system and the process begins anew.

Influence Function for Interval Schemata

Use Cultural Algorithms
as a framework in which to perform
knowledge-based evolutionary learning

Replace σ_i with empirical generalizations
produced in the belief space.

Mutation $$x_i^{'} = x_i + \sigma_i \cdot N_i(0,1)$$

① *Interval size information*

② *Directional knowledge*

How is this done?

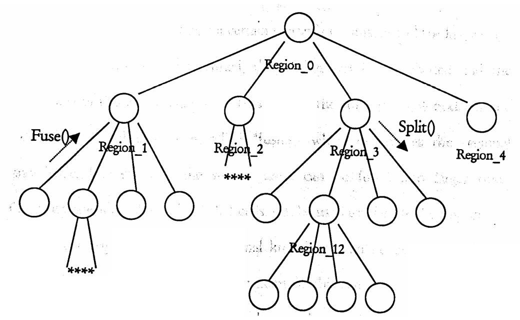
Adding Constraint Knowledge

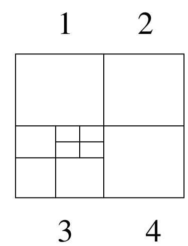
- With the addition of constraint knowledge, n one dimensional interval schemata are combined to produce an n-dimensional region.
- Regional schemata result from imposing a grid system of a certain granularity on the space.
- Grid squares are sampled by scouts. They can be classified based upon the problem characteristics they exhibit: e.g. feasible, infeasible, partially feasible, etc.
- The influence function here cause individuals to migrate to or from cells as a function of their characteristics.
- New cells are broken down into subregions, explored and exploited.
- Knowledge base operations allow the fissioning and fusioning of cells.

Regional Schema: an n-dimensional region defined as a combination of intervals that circumscribe a portion of n-dimensional space

NOW WE EXTEND THIS BY ALLOWING

1. **MULTIPLE M-DIMENSIONAL REGIONAL SCHEMATA**
2. **THE ORGANIZATION OF THESE SCHEMATA INTO A HIERARCHICAL STRUCTURE.**





<u>ACCEPTANCE FUNCTION :</u>

HERE, ALL INDIVIDUALS ARE USED TO UPDATE CONSTRAINT KNOWLEDGE. THE TOP 20% ARE USED TO UPDATE THE NORAMTIVE KNOWLEDGE.

THESE 20% ARE CALLED THE EMINENT INDIVIDUALS.

<u>UPDATE:</u>

USE INFERENCE RULES TO ADJUST THE CLASSIFICATION OF ACTIVE CELLS. E.G. FEASIBLE, INFEASIBLE, SEMI-FEASIBLE.

ADUST THE HIERARCHICAL STRUCTURE BASED UPON THIS INFORMATION. E.G.

FISSION: SPLIT A SEMI-FEASIBLE CELL INTO SMALLER CELLS WHEN THE NUMBER OF INDIVIDUALS BECOMES TOO HIGH.

FUSION: MERGE , RECOMBINE CHILDREN INTO THE ORIGINAL PARENT. THEN CAN DECOMPOSE THE PARENT IN A DIFFERENT WAY. E.G. CURRENT DECMPOSITION IS UNATTRACTIVE. E.G. INFEASIBLE CELL BECOMES SEMI-FEASIBLE.

INFLUENCE FUNCTION:

GUIDE THE MIGRATION OF INDIVIDUALS FROM LESS PRODUCTIVE CELLS, INFEASIBLE, TO ONES THAT ARE MORE PRODUCTIVE, SEMI-FEASIBLE AND FEASIBLE CELLS. SEMI-FEASIBLE AND FEASIBLE CELLS WITH EMINENT INDIVIDUALS ARE CALLED EMINENT. HIGHLIGHT THE MIGRATION TO EMINENT CELLS FROM ORDINARY ONES.

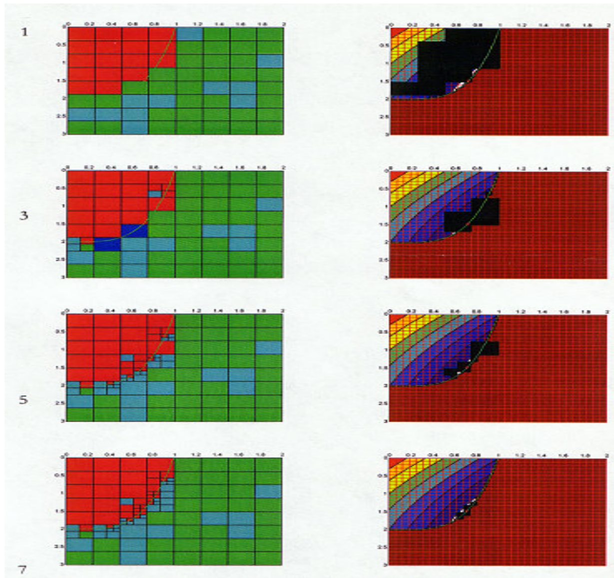
1. PERTURB INDIVIDUALS A LITTLE IN EMINENT CELLS.
2. MOVE INDIVIDUALS IN INFEASIBLE CELLS TO FEASIBLE ONES.
3. MOVE INDIVIDUALS FROM ORDINARY TO EMINENT CELLS.

Implementation and test results

To access the approaches, we used a nonlinear constrained optimization problem [Floudas 1990], which is given below:

Problem Description

Min $-12x-7y+y^2$
Domain constraints: $0 \leq x \leq 2,\ 0 \leq y \leq 3$
Problem constraints: $y \leq -2x^4+2$
Global best point: $x^*=0.71751,$
$y^*=1.470$
Global best value: -16.73889
Optimization goal: < -16.70



Evolution of Constraint Knowledge   Evolution of Normative Knowledge

Cultural Algorithm Configuration: Embedding Other Methods

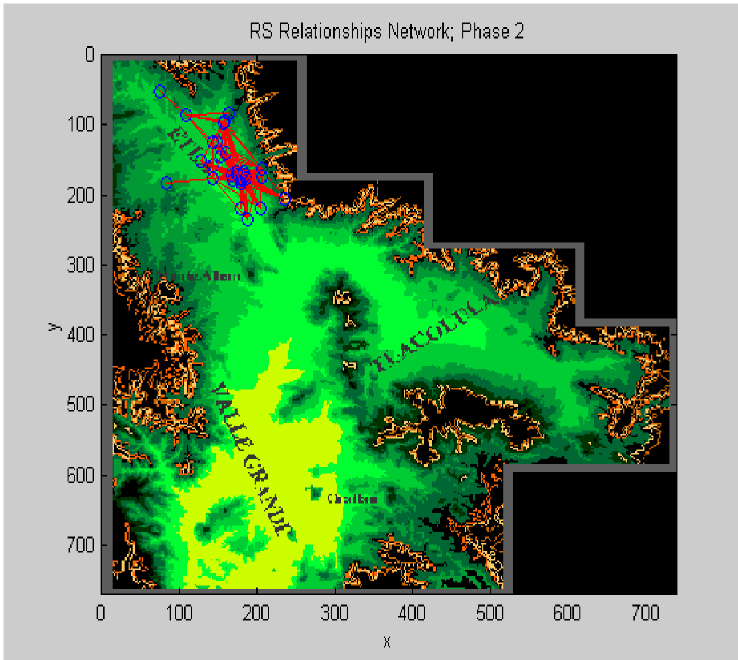
- Population models used
  - Genetic Algorithms (Concept learning, optimization)
  - Genetic Programming (Evolving agent strategies)
  - Evolutionary Programming (Real valued function optimization)
  - Evolution Strategies (Robot soccer plays)
  - Memetic models (Evolution of agriculture)
  - Agent based modeling (Evolution of the state, Environmental Impact)

Knowledge Models Used

- Schemata
- Binary valued (Maleticconcept learning, Boole problem, data mining)
- Real-valued interval schemata (Chang:unconstrained optimization)
- Fuzzy Real-valued schemata
- Regional Schemata ((Xidong Jin)constrained optimization)
- Semantic Networks (DLMS:Rychtyckyj)
- Graphical Models (GP:Zannoni, Ostrowski)
- Logical and Rule Based models (HYBAL(Sverdlik), Fraud Detection (Sternberg), Lazar (Data mining)

Evolution of the State

- Evolution of Complex Social Systems
- Valley of Oaxaca, Mexico
- Implement Marcus and Flannery's Model of State Formation and Observe the Social Networks that form as a result.
- Compare to the Archaeological data for the Valley



Future Directions

- Integrating Multiple Representations and Population Models
- Parallelization
- Belief Space Evolution
- Designing Cultural Systems
- How does a Culture’s structure and content reflect its problem solving environment (Saleem)

A Selected Bibliography of Cultural Algorithms

Book Chapters:

Reynolds, R.G., "The Impact of Raiding on Settlement Patterns in the Northern Valley of Oaxaca: An Approach Using Decision Trees, Dynamics in Human and Primate Societies: Agent-Based Modelling of Social and Spatial Processes, T. Kohler and G. Gummerman, Editors, Oxford University Press, 1999.

Reynolds, R.G., "An Overview of Cultural Algorithms", Advances in Evolutionary Computation, McGraw Hill Press, 1999.

Reynolds, R. G., "Why Does Cultural Evolution Proceed at a Faster Rate Than Biological Evolution?", in Time, Process, and Structured Transformation in Archaeology, Sander van der Leeuw and James McGlade Editors, Routledge Press, New York, NY, 1997, pp. 269-282.

Reynolds, R. G., "Introduction to Cultural Algorithms", in Proceedings of the Third Annual Conference on Evolutionary Programming, Anthony V. Sebald and Lawrence J. Fogel, Editors, World Scientific Press, Singapore, 1994, pp.131-139.

Reynolds, R. G., "Learning to Cooperate Using Cultural Algorithms", in Simulating Societies, Nigel Gilbert and J. Doran, Editors, University College of London Press, 1994, pp. 223-244.

Reynolds, R. G., "An Adaptive Computer Model for the Evolution of Plant Collecting and Early Agriculture in the Eastern Valley of Oaxaca", in Guila Naquitz: Archaic Foraging and Early Agriculture in Oaxaca, Mexico, K. V. Flannery, Editor, Academic Press, 1986. pp. 439-500.

Reynolds, R. G., "Multidimensional Scaling of Four Guila Naquitz Living Floors", in Guila Naquitz: Archaic Foraging and Early Agriculture in Oaxaca, Mexico, K. V. Flannery, Editor, Academic Press, 1986.

Book Chapters Co-Authored:

Reynolds, R.G., and Chung, Chan-Jin, "Function Optimization using Evolutionary Programming with Self-Adaptive Cultural Algorithms", Lecture Notes on Artificial Intelligence, Springer-Verlag Press, 1997, pp. 184-198.

Reynolds, R.G., and Chung, Chan-Jin, "A Cultural Algorithm to Evolve Multi-Agent Cooperation Using Cultural Algorithms", in Evolutionary Programming VI, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Editors, Springer-Verlag Press, New York, NY, 1997, pp. 323-334.

Reynolds, R.G., and Nazzal, Ayman, "Using Cultural Algorithms with Evolutionary Computing to Extract Site location Decisions From Spatio-Temporal Databases, in Evolutionary Programming VI, P. J. Angeline, R. G. Reynolds, J. R. McDonnell, and R. Eberhart, Editors, Springer-Verlag Press, New York, NY, 1997, pp. 323-334.

Reynolds, R. G., and Chung, Chan-Jin, "A Test Bed for Solving Optimization Problems Using Cultural Algorithms", in Evolutionary Programming V, John R. McDonnell, and Peter Angeline, Editors, A Bradford Book, MIT Press, Cambridge Massachusetts, 1996, pp. 225-236.

Reynolds, R. G., and Zannoni, Elena, "Extracting Design Knowledge from Genetic Programs Using Cultural Algorithms", in Evolutionary Programming V, Peter Angeline, Editor, A Bradford Book, MIT Press, Cambridge Massachusetts, 1996, pp. 217-224.

Reynolds, R.G., Michalewicz Z., and Cavaretta M. J., "Using Cultural Algorithms for Constraint Handling in Genocop", in Evolutionary Programming IV, J. R. McDonnell, R.G. Reynolds, and David B. Fogel, Editors, a Bradford Book, MIT Press, Cambridge, Massachusetts, 1995.

Reynolds, R.G., and Maletic J. I., "The Evolution of Cooperate using Cultural Algorithms", in Proceedings of the Third Annual Conference on Evolutionary Programming, Anthony V. Sebald and Lawrence J. Fogel, Editors, World Scientific Press, Singapore, 1994, pp.141-149.

Reynolds R. G., Zannoni, E., and Posner, R. M., "Learning to Understand Software using Cultural Algorithms", in Proceedings of the Third Annual Conference on Evolutionary Programming, Anthony V. Sebald and Lawrence J. Fogel, Editors, World Scientific Press, Singapore, 1994, pp.150-157.

Reynolds, R. G. , Brown, W., and Abinoja, E., "Guiding Parallel Bidirectional Search with Cultural Algorithms, in Proceedings of the Third Annual Conference on Evolutionary Programming, Anthony V. Sebald and Lawrence J. Fogel, Editors, World Scientific Press, Singapore, 1994, pp.167-174.

Reynolds, R. G. and Zeigler, B.,"Information Processing Models for Hunter-Gatherer Decision Making", in Mathematical Models of Cultural Change, Colin Renfrew and Kenneth Cooke, Editors, Academic Press, December 1978. pp. 485-418.

Journal Articles:

Reynolds, R.G., Jin, X.*, “Regional Schemata for Real-Valued Constrained Function Optimization Using Cultural Algorithms, Journal of Natural Computing, T. Back, Editor, in press, to appear 2002.

Reynolds, R.G., Goodhall, S.,and Whallon, R., “Transmission of Cultural Traits by Emulation: An Agent Based Model of Group Foraging Behavior”, Journal of Memetics, March, 2001.

Reynolds, R. G., and Zhu, Shinin, “Fuzzy Cultural Algorithms with Evolutionary Programming for Real-Valued Function Optimization”, IEEE Transactions on Systems, Man, and Cybernetics, Part B:Cybernetics, Vol. 31, No. 1, February, 2001, pp. 1-18.

Reynolds, R. G., and Chung, Chan-Jin*, “Knowledge-Based Self-Adaptation in Evolutionary Search”, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 1, 2000.

Reynolds, R.G., and Chung, Chan Jin*, "CAEP: An Evolution-Based Tool for real-Valued Function Optimization Using Cultural Algorithms", International Journal on Artificial Intelligence Tools, Vol. 7, No. 3, September, 1998, pp. 239-293.

Reynolds, R. G., and Sternberg, Michael*, "Using Cultural Algorithms to Support the Re-Engineering of Rule-Based Expert Systems in Dynamic Performance Environments: A Fraud Detection Example", IEEE Transactions on Evolutionary Computation, Vol.1, No. 4, November, 1997, pp. 225-243.

Reynolds, R. G., and Zannoni, E.*, "Learning to Control the Program Evolution Process in Genetic Programming Systems Using Cultural Algorithms", Journal of Evolutionary Computation, Vol. 5, No. 2, October, 1997, pp. 181-211.

Reynolds, R. G., "Evolution-Based Approaches to Software Engineering: An Introduction", International Journal of Software Engineering and Knowledge Engineering, Vol. 5, No.2, June, 1995, pp. 161-164.

Reynolds, R.G., and Sverdlik, W., "An Evolution-Based Approach to Program Understanding Using Cultural Algorithms", International Journal of Software Engineering and Knowledge Engineering, Vol. 5, No.2, June, 1995, pp. 211-226.

Reynolds, R. G., and Maletic, J., "The Use of Version Space Controlled Genetic Algorithms to Solve the Boole Problem" International Journal on Artificial Intelligence Tools, Vol. 2, No. 2, June, 1993, pp. 219-234.

Reynolds, R. G., and Savatsky, K.*, "A Computer Model of the Evolution of Cooperation", Biosystems, Vol. 23, 1989, pp. 261-279.

Reynolds, R. G., " A Computational Model of Hierarchical Decision Systems", Journal of Anthropological Archaeology, Academic Press, Vol. 3, September, 1984. pp. 159-189.

Reynolds, R. G., "On Modeling the Evolution of Hunter-Gatherer Decision-Making Systems", Geographical Analysis, Vol. X, No. 1, January, 1978. pp. 31-46.

Papers Published in Conference Proceedings:

Reynolds, R., Tassier, T., Everson, M., and Ostrowski, D.*, Using Cultural Algorithms to Evolve Strategies in Agent-Based Models", Proceedings of World Congress on Computational Intelligence, May 12-19, 2002, Honolulu, Hawaii.

Reynolds, R., Rychtyckyj,N.*,"Knowledge Base Maintenance Using Cultural Algorithms: Application to the DLMS Manufacturing Process System", Proceedings of World Congress on Computational Intelligence, May 12-19, 2002, Honolulu, Hawaii.

Reynolds, R., and Lazar, A.,"Simulating the Evolution of the Archaic State", Proceedings of World Congress on Computational Intelligence, May 12-19, 2002, Honolulu, Hawaii.

Reynolds, R, Whallon, R., and Goodhall, S*. "The Impact of Resource Access on Learning by Emulation in Hunter-Gatherer Foraging Systems: A Multi-Agent Model", Proceedings of World Congress on Computational Intelligence, May 12-19, 2002, Honolulu, Hawaii.

Reynolds, R., and Lazar, A., "A Computational Framework for Modelling the Dynamic Evolution of Large-Scale Multi-Agent Organizations" Proceedings SPIE Conference on Enabling Technologies for Simulation Science, April 1-5, 2002.

Reynolds, R. and Lazar, A.*, "Evolution-Based Learning of Ontological Knowledge for a Large-Scale Multi-Agent Simulation", Proceedings of 4th International Workshop on Frontiers in Evolutionary Computation, Duke University, March 11-13, 2002.

Reynolds, R.G., "Knowledge Swarms and Cultural Evolution", Proceedings of American Anthropological Association Annual Meeting, November 28-31, 2001, Washington, D.C.

Reynolds, R.G., and Rychtyckyj, N.*, "Bottom-Up Re-Engineering of Semantic Networks using Cultural Algorithms", Proceedings of GECCO 2001, San Francisco, California, July 7-11, 2001.

Reynolds, R.G., and Saleem, S.*, "Cultural and Social Evolution in Dynamic Environments", CASOS 2001, Carnegie-Mellon University, July 5-7, 2001.

Reynolds, R.G., and Saleem, S.*, "Knowledge-Based Function Optimization in Dynamic Environments Using Cultural Algorithms", 2001 International Conference on Artificial Intelligence, Las Vegas, Nevada, June 25-28, 2001.

Reynolds, R.G., and Saleem, S.*, "Evolutionary Learning in Dynamic Environments Using Cultural Algorithms",Workshop on Emergence, Transformation, and Decay in Socio-Natural Systems, Abisko, Sweden, May 19-23, 2001.

Reynolds, R.G., and Saleem, S.*, "Function Optimization with Cultural Algorithms in Dynamic Environments, Proceedings of the Particle Swarm Optimization Workshop, Indianapolis, Indiana, April 6-7, 2001, pp: 63-79.

Reynolds, R.G., Goodhall, S., Whallon, R., "Modeling Imitative Learning in a Multi-Agent System Using Cultural Algorithms and Swarm", Proceedings of Agent Simulation 2000: Applications, Models, and Tools", Chicago, Illinois, October 5-7, 2000.

Reynolds, R.G., Goodhall, S., Whallon, R., "Modeling Imitative Learning : A Hunter-Gatherer Model", Proceeding of Sienna Workshop on Cultural Evolution, Sienna, Italy, September 2-4, 2000

Reynolds, R.G., and Rychtyckyj, N.*, "Assessing the Performance of Cultural Algorithms for Semantic Network Re-engineering", Proceedings of the Congress on Evolutionary Computation, San Diego, California, July 16-19, 2000, Vol. 2, pp: 1482-1491.

Reynolds, R.G., and Jin, X.*, "Mining Knowledge in Large-Scale Databases Using Cultural Algorithms with constraint handling Mechanisms", Proceedings of the Congress on Evolutionary Computation, San Diego, California, July 16-19, 2000, Vol. 2, pp: 1498-1506.

Reynolds, R.G., and Saleem, S.*, "Cultural Algorithms in Dynamic Environments", Proceedings of the Congress on Evolutionary Computation, San Diego, California, July 16-19, 2000, Vol. 2, pp: 1513-1520.

Reynolds, R.G., and Jin, X.*, "Using Knowledge-Based System with a Heirarchical Architecture to guide the Search of Evolutionary Computation", Proceedings of the Eleventh IEEE Conference Tools with Artificial Intelligence, Chicago, Il, Nov. 10-12, 1999.

Reynolds, R.G., and Jin, X.*, "Solving Constrained Real-Valued Function Optimization Problems using a Cultural Algorithm", Proceedings ANNIE 1999, St. Louis, Mo., Nov. 7-9, 1999.

Reynolds, R.G., and Rychtyckyj, N.*, "Using Cultural Algorithms to Improve Performance in Semantic Networks", in Proceedings 1999 IEEE Congress on Evolutionary Computation, Washington, D. C., July 6-9, 1999, pp. 1651-1656.

Reynolds, R.G., and Ostrowski, D.*, "Knowledge-Based Software Testing Agent Using Evolutionary Learning with Cultural Algorithms", in Proceedings 1999 IEEE Congress on Evolutionary Computation, Washington, D. C., July 6-9, 1999, pp. 1657-1663.

Reynolds, R.G., and Cowan, G.*, "The Metrics Apprentice: Using Cultural Algorithms to Formulate Quality Metrics for Software Systems", in Proceedings 1999 IEEE Congress on Evolutionary Computation, Washington, D. C., July 6-9, 1999, pp. 1664-1671.

Reynolds, R.G., and Jin, X.*, "Using Knowledge-Based Evolutionary Computation to Solve Non-Linear Optimization Problems: A Cultural Algorithm Approach", in Proceedings 1999 IEEE Congress on Evolutionary Computation, Washington, D. C., July 6-9, 1999, pp. 1672-1678.

Reynolds, R.G., and Cowan, G.*, "Learning to Assess the Quality of Genetic Programs Using Cultural Algorithms", in Proceedings 1999 IEEE Congress on Evolutionary Computation, Washington, D. C., July 6-9, 1999, pp. 1679-1686.

Reynolds, R. G., "On the Evolution of Schemata for Function Optimization", in Holland Fest: New Directions in Evolutionary Computation Inspired by the Work of John Holland, Ann Arbor, Michigan, May 16-18, 1999

Reynolds, R.G., and Chung, Chan-Jin, “A Knowledge-Based Approach to Self-Adaptation in Evolutionary Search Using Cultural Algorithms”, in Proceedings of the 12th International FLAIRS Conference, Orlando, Florida, May 3-6, 1999.

Reynolds, R.G., and Cowan, G*, “Evolving Distributed Software Engineering Environments”, in Proceedings 17th IEEE Symposium on Reliable Distributed Systems, West Lafayette, Indiana, October 20-23, 1998, pp: 151-160.

Reynolds, R.G., and Zhu, S., “The Impact of Fuzzy Knowledge Representation on Problem Solving in Fuzzy Cultural Algorithms with Evolutionary Programming”, Proceedings of Genetic Programming Conference, Madison, Wisconsin, July 22-25,1998, Morgan Kaufmann Press.

Reynolds, R.G., and Zhu, S., “The Design of Fully Fuzzy Cultural Algorithms with Evolutionary Programming for Real-Valued Function Optimization”, Proceedings of Genetic Programming Conference, Madison, Wisconsin, July 22-25, 1998, Morgan Kaufmann Press.

Reynolds, R.G., and Al-Shehri, H., “Data Mining of Large-Scale Spatio-Temporal Databases Using Cultural Algorithms”, Proceedings of 1998 IEEE World Congress on Computational Intelligence, Anchorage, Alaska, May 4-9, 1998.

Reynolds, R.G., and Rychtychyj, N.*, “Learning to Re-Engineer Semantic Networks Using Cultural Algorithms”, Proceedings of Seventh Annual Conference on Evolutionary Programming, San Diego, California, March 26-29, 1998.

Reynolds, R.G., and Ostrowski, D*., “Developing Software Engineering Environments for Genetic Programming Systems Using Cultural Algorithms”, Proceedings of Seventh Annual Conference on Evolutionary Programming, San Diego, California, March 26-29, 1998.

Reynolds, R.G., and Chung, C*., “Culturing Evolution Startegies to Support the Exploration of Novel Environments by an Intelligent Robotic Agent”, Proceedings of Seventh Annual Conference on Evolutionary Programming, San Diego, California, March 26-29, 1998.

Reynolds, R.G., and Zhu, S.*., “Fuzzy Cultural Algorithms with Evolutionary Programming”, Proceedings of Seventh Annual Conference on Evolutionary Programming, San Diego, California, March 26-29, 1998.

Reynolds, R.G., and Chung, Chan Jin, "Knowledge-Based Self Adaptation in Evolutionary Search", Proceedings of 1997 IEEE International Conference on Artificial Intelligence Tools, Newport Beach, November 4-7, 1997.

Reynolds, R.G., and Al-Shehri, H., “The Use of Cultural Algorithms with Evolutionary Programming to Control the Data Mining of Large-Scale Spatio-Temporal Databases”, 1997 IEEE International Conference on Systems, man, and Cybernetics, Orlando, Florida, October 15, 1997

Reynolds, R. G., and Chung, Chan Jin, "The Importance of Functional Complexity in Regulating the Amount of Information Required to Guide Self-Adaptation in Cultural Algorithms, Proceedings 1997 International Conference on Genetic Algorithms, East Lansing, Michigan, July, 1997, pp. 401-408.

Reynolds, R.G., Chung, Chan Jin, "Knowledge-Based Self-Adaptation in Evolutionary Programming Using Cultural Algorithms", Proceedings of 1997 IEEE International Conference on Evolutionary Computation, Indianapolis, Indiana, April, 1997, pp. 71-76.

Reynolds, R. G., and Chung, Chan-Jin, "A Cultural Algorithm Framework for Evolving Multi-Agent Cooperation Using Evolutionary Programming", Proceedings of International Conference on Evolutionary Programming, Indianapolis, Indiana, 1997, pp. 323-334.

Reynolds, R. G., and Nazzal, Ayman, "Using Cultural Algorithms with Evolutionary Computing to Extract Site Location Decisions from Spatio-Temporal Databases", Proceedings of International Conference on Evolutionary Programming, Indianapolis, Indiana, 1997, pp. 443-456.

Reynolds, R.G., and Zannoni, E., "Evolving Software Design Methodologies in Automatic Programming Systems Using Cultural Algorithms", Proceedings of Second World Congress on Integrated Design and Process Technology", Austin, Texas, December 1-4, 1996.

Reynolds, R. G., and Chung, Chan-Jin, "Function Optimization Using Evolutionary Programming with Self-Adaptive Cultural Algorithms, Proceedings of First Asian-Pacific Conference on Simulated Evolution and Learning, Taejon, Korea, November 8 -12, 1996.

Reynolds, R.G. and Chung Chan-Jin, "The Use of Cultural Algorithms to Evolve Multiagent Cooperation", Proceedings of 1996 World Cup Soccer Tournament, Taejon, Korea, November 8 -12, 1996 .

Reynolds, R. G., and Chung, Chan-Jin, "The Use of Cultural Algorithms to Support Self-Adaptation in Evolutionary Programming", Proceedings of 1996 Adaptive Distributive Parallel Computing Symposium, Dayton, Ohio, August 8-9, 1996, pp. 260-271.

Reynolds, R.G., Chung, Chan Jin, "A Self-Adaptive Approach to Representation Shifts in Cultural Algorithms", Proceedings of 1996 IEEE International Conference on Evolutionary Computation, May 20-22, Nagoya, Japan, pp. 94-99.

Reynolds, R. G., and Chung, Chan Jin, "A Test Bed for Solving Optimization Problems Using Cultural Algorithms", Proceedings of Fifth Annual Conference on Evolutionary Programming, February 29-March 2, 1996, San Diego, California.

Reynolds, R. G., and Zannoni, Elena, "Extracting Design Knowledge from Genetic Programs Using Cultural Algorithms", Proceedings of Fifth Annual Conference on Evolutionary Programming, February 29-March 2, 1996, San Diego, California.

Reynolds, R. G., Rolnick, S. R., "Learning the Parameters for a Gradient-Based Approach to Image Segmentation from the Results of a Region Growing Approach Using Cultural Algorithms", 1995 IEEE International Conference on Evolutionary Computation, November 29-December 1, 1995, Perth, Australia, pp. 1135-1143.

Reynolds, R. G., Rolnick, S. R., "Learning the Parameters to a Gradient-Based Approach to Image Segmentation Using Cultural Algorithms", Proceedings International Symposium on Intelligence in Neural and Biological Systems, May 29-31, 1995, Herndon, Virginia, pp. 240-247.

Reynolds, R.G., "Solving Design Problems Using Cultural Algorithms", Proceedings of the Eighth Florida Artificial Intelligence Research Symposium, April 27-29, 1995, Melbourne, Florida, pp. 279-283.

Reynolds, R. G., Sverdlik, W., "Problem Solving Using Cultural Algorithms", Proceedings of 1st IEEE World Congress on Computational Intelligence, June 26-July 2, 1994, Orlando, Florida, pp. 1004-1008.

Reynolds, R. G., and Zannoni, E., "Learning to Understand Software From Examples using Cultural Algorithms", Proceedings of the 6th International Conference on Software Engineering and Knowledge Engineering, Riga, Latvia, June 21-23, 1994, pp. 188-192.

Reynolds, R. G., Cavaretta, M., "Discovering Search Heuristics for Concept Learning Using Version Space Guided Genetic Algorithms", Proceedings of Florida Artificial Intelligence Research Symposium, Pennsacola, Florida, May 5-7, 1994, pp. 183-192.

Reynolds, R. G., "An Introduction to Cultural Algorithms", Proceedings of the Third Annual Conference on Evolutionary Programming, February 24-26, 1994, San Diego, California, pp. 131-139.

Reynolds, R. G., Maletic, J., "Learning to Cooperate Using Cultural Algorithms", Proceedings of the Third Annual Conference on Evolutionary Programming, February 24-26, 1994, San Diego, California, pp. 140-149.

Reynolds, R. G., Brown W., Abinoja, E., "Guiding Parallel Bidirectional Search Using Cultural Algorithms", Proceedings of the Third Annual Conference on Evolutionary Programming, February 24-26, 1994, San Diego, California, pp. 167- 174.

Reynolds, R.G., Zannoni, E., Posner, R., "Learning to Understand Software Using Cultural Algorithms", Proceedings of the Third Annual Conference on Evolutionary Programming, February 24-26, 1994, San Diego, California, pp. 150-157.

Reynolds, R. G., and Sverdlik, W., "Incorporating Domain Specific Knowledge into Version Space Search", Proceedings of the Second World Congress on Expert Systems, Lisbon, Portugal, January 10-14, 1994.

Reynolds, R. G., and *Sverdlik, W., "Scaling Up Version Spaces by Using Domain Specific Algorithms", Fifth International Conference on Tools for Artificial Intelligence, November 8-11, 1993, pp. 216-223.

Reynolds, R. G., and Sverdlik, W., "Learning the Behavior of Boolean Circuits From Examples Using Cultural Algorithms", Proceedings of Second Adaptive Learning Systems Conference, SPIE International Symposium on Aerospace and Remote Sensing, Orlando, Florida, April 12-16, 1993, 177-188.

Reynolds, R. G., and Sverdlik, W., "Solving Problems in Hierarchical Systems Using Cultural Algorithms", Proceedings of Second Annual Conference on Evolutionary Programming, La Jolla, California, February 27 - 29, 1993, pp. 144-153.

Reynolds, R. G. and *Sverdlik, W., "Dynamic Version Spaces in Machine Learning", Proceedings of 1992 IEEE Conference on Tools for Artificial Intelligence, Arlington, Virginia, November 10-13, 1992.

Reynolds, R. G., and Zannoni, E, "Why Cultural Evolution Can Proceed Faster Than Biological Evolution", in Proceeding of International Symposium on Simulating Societies, Surrey, England, April 2-3, 1992, pp. 81-93.