

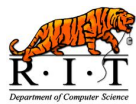


Department of Computer Science

Permutation Based GAs and Ordered Greed

Peter G. Anderson, Rochester Institute of Technology
Daniel Ashlock, University of Guelph

November 2004



ANNIE Outline

- Permutation problems for GA's to solve.
- The N Queens problem.
- Introducing Ordered Greed.
- Creating permutations.
- Representing permutations with signatures.

ANNIE Outline

- Crossovers with signatures.
- MOX: merging crossover.
- Even preceed odds: a toy problem to compare crossovers.
- Comparing crossovers with N Queens.
- Coloring random planar Hamiltonian graphs.
- Ordered Greed application areas.



N Queens—an Illustration of Permutation-Based GA

Place N mutually un-attacking Queens on an $N \times N$ chess board.

(Queens attack on rows, columns, and diagonals.)

((Noise on the Internet: \mathcal{NP} complete, indeed!))



Permutation Is Placement

Let the GA creatures be permutations of $(0, 1, \dots, N - 1)$:

Individual = $(c_0, c_1, \dots, c_{N-1})$.

Interpret this as a Queens placement:

The Queen in row k is in column c_k .

Fitness: number of Queens unattacked by Queens on previous rows.



How to Make a Permutation

```
for  $i = 0$  to  $N-1$  do  
     $P_i = i$   
end for  
for  $i = 0$  to  $N-1$  do  
     $k = \text{random\_int}(N - i)$   
    Interchange  $P_i$  with  $P_{i+k}$   
end for
```

This uniformly generates permutations of $\{0, \dots, N - 1\}$

First Example of Ordered Greed (OG)

Permutations Order Placement

The GA creatures are permutation of $(0, 1, \dots, N - 1)$:

individual = $(c_0, c_1, \dots, c_{N-1})$.

Interpret this as a placement ordering:

```
for  $i = 0$  to  $N-1$  do  
    Place the Queen in row  $c_j$   
    in the left-most safe column  
end for
```

Fitness: the number of successfully placed Queens.



Ordered Greed in Action

Permutation individual: 3 4 1 5 7 0 6 2

Row 3, col 0

Row 4, col 2

Row 1, col 1

Row 5, col 4

Row 7, col 3

Row 0, col 5

Row 6, col 7

Row 2, col 6

Fitness = 8 = 100%

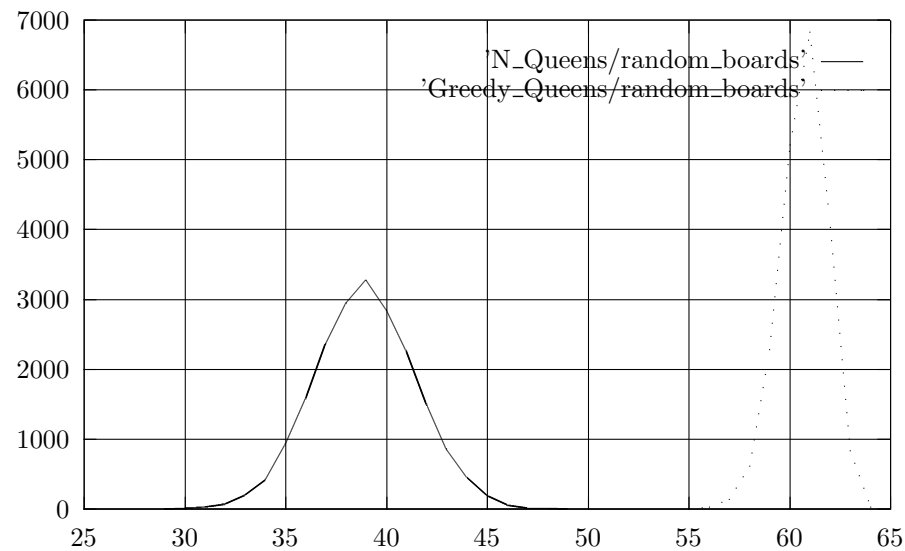


Greed Ordered by 3 4 1 5 7 0 6 2

	0	1	2	3	4	5	6	7
0						6		
1		3						
2							8	
3	1							
4			2					
5					4			
6								7
7				5				

Fitnesses of 20,000 Random 64^2 Boards

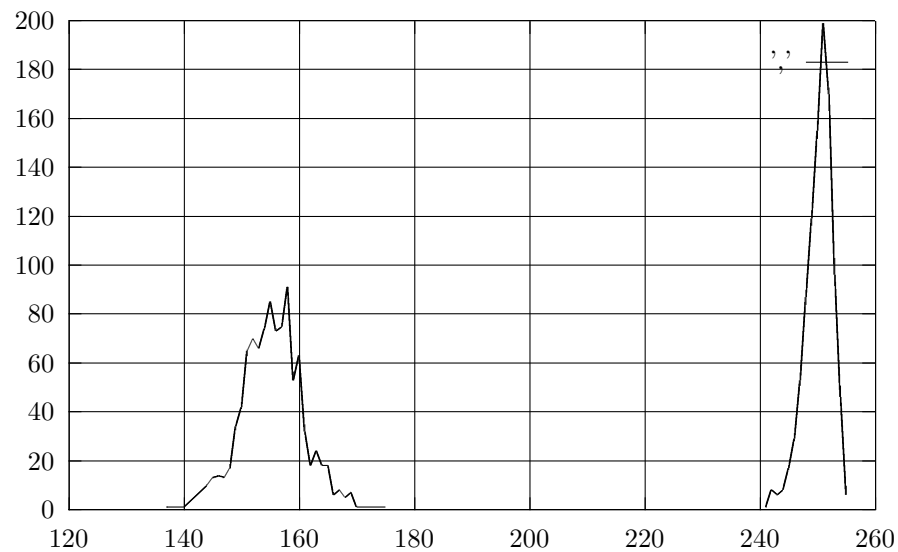
Successfully placed Queens' histograms.



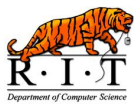
Note the efficacy of Ordered Greed vs. random placement.



Fitnesses of Some Random 256^2 Boards



Note the efficacy of Ordered Greed vs. random placement.



Representing Permutations

A permutation's *signature* is a list of places.

$\{p_0, p_1, p_2, \dots, p_{N-1}\}$ satisfying: $0 \leq p_k < N - k$

Meaning: “Try to put k in position p_k ”

(There are exactly $N!$ lists.)

Creating a Signature

```
for  $i = 0$  to  $N-1$  do  
     $S_i = \text{random\_int}(N - i)$   
end for
```

This procedure uniformly generates permutation signatures.

Manipulating Signatures

Preserve the rule: $0 \leq s_k < N - k$

Mutate:

- Increment or decrement $s_k \bmod N - k$
- Replace s_k with random value mod $N - k$

Crossover, as usual:

- One point.
- Two point.
- Uniform.



Converting a Signature to a Permutation

```
for  $i = 0$  to  $N-1$  do  
     $P_i = i$   
end for  
for  $i = 0$  to  $N-1$  do  
    Interchange  $P_i$  with  $P_{i+S_k}$   
end for
```

We require that P_{i+S_i} does not precede P_i . i.e.: $0 \leq S_i < N - i$

This decoding only costs $\mathcal{O}(N)$.

Inverting: Given a Permutation, What is Its Signature?

A problem for the interested student.

This will demonstrate the 1-1 correspondence between signatures and permutations.



Merging Crossover: MOX

Randomly merge two parents into a $2N$ -element list, L (this operation is similar to a riffle shuffle of cards).

The first instance of each value in L gives the first child, and the second instance gives the second child.

Example parents: $p_1 = \{3\ 9\ 0\ 1\ 2\ 4\ 6\ 8\ 7\ 5\}$, $p_2 = \{2\ 6\ 7\ 1\ 4\ 8\ 0\ 3\ 5\ 9\}$

Merge p_1 and p_2 : $L = \{2\ \mathbf{3}\ 6\ 7\ 1\ 4\ \mathbf{9}\ \mathbf{0}\ 8\ 0\ \mathbf{1}\ 2\ 3\ \mathbf{4}\ \mathbf{6}\ 5\ \mathbf{8}\ \mathbf{7}\ \mathbf{5}\ 9\}$ (the elements from p_1 are shown in bold).

Extract children: $c_1 = \{2\ \mathbf{3}\ 6\ 7\ 1\ 4\ \mathbf{9}\ \mathbf{0}\ 8\ 5\}$, $c_2 = \{0\ \mathbf{1}\ 2\ 3\ \mathbf{4}\ \mathbf{6}\ \mathbf{8}\ \mathbf{7}\ \mathbf{5}\ 9\}$ (the p_1 contribution is still shown in bold).



Notes and Properties of MOX

The intermediate list, L , is not needed, except conceptually.

All we need is a one-element buffer, X , that is filled from the initial elements of the two parents, treated as queues, chosen at random.

X is appended to the first child, if X is not already present, otherwise it is appended to the second child.

MOX Pseudocode

```
for  $i = 1$  to  $2N$  do  
  if random choice = 1 and  $p_1$  not depleted then  
     $X \leftarrow$  next element of  $p_1$   
  else  
     $X \leftarrow$  next element of  $p_2$   
  end if  
  if  $X$  is not already in  $c_1$  then  
    add  $X$  to  $c_1$   
  else  
    add  $X$  to  $c_2$   
  end if  
end for
```



Notes and Properties of MOX

OG seeks good precedence orders among permutation elements.

Let “ $a \prec b$ ” denote that “ a precedes b ” in a given permutation.

MOX seems particularly suitable for OG because:

$$a \prec b \text{ in both parents} \Rightarrow a \prec b \text{ in both children}$$

If $a \prec b$ in one parent and $b \prec a$ in the other, then both children can have $a \prec b$, both can have $b \prec a$, or the two children can be mixed.

Other permutation crossovers (signatures, CX, OX, PMX) can fail to preserve two parents' $a \prec b$.



Adam and Eve

Any permutation and its reverse can produce *any* permutation as an eventual descendant.

(Binary string one-point crossover can do this starting with any string and its complement.)

MOX and Generalizations of Permutations

OG can deal with permutations of multisets

i.e., sets in which some elements appear more than once.

Application: assign several workers (e.g., faculty) to several jobs (classes).

In advance, determine how many jobs each person will do.

A person who must get k jobs appears k times in a list.

MOX can breed such lists.



Even Precedes Odds: A Toy Problem

A permutation-based analogy to the problem
maximize the number of 1's in a binary string.

A perfect string has all the even numbers in the left half.

Partial credit (gives a pretty good gradient towards solutions):

if $k < N/2$ and P_k with the right parity contributes $N/2 - k$

if $k > N/2$ and P_k with the right parity contributes $1 + k - N/2$

Use this problem with $N = 100$, population=100, mutation rate=0.001
to compare MOX, PMX, and signatures.

Each problem ran 100 times with different random seeds.



Even Precedes Odd Results: MOX Wins!

xover	min	Q1	median	Q3	max
MOX	3,769	5,647	6,392	7,347	11,028
PMX	5,239	17,069	30,012	59,263	—
Sig. 1 Pt.	15,824	31,325	44,534	70,175	—
Sig. 2 Pt.	12,514	29,298	45,928	68,969	—
Sig. Unif.	8,648	23,764	38,847	61,547	—

Fitness evaluation counts to solve “evens precede odds” for five crossovers.

100 tries for each crossover. We report minimum, first quartile, median, third quartile, and maximum number of fitness evaluations.

“—”: Max > 100,000, so the process was stopped.



500 Queens Results: MOX Wins! (Mostly)

xover	min	Q1	median	Q3	max
MOX	162	974	1,444	2,049	5,005
PMX	184	988	1,722	2,573	7,862
Sig. 1 Pt.	128	766	1,144	1,627	29,706
Sig. 2 Pt.	30	763	1,142	1,726	10,885
Sig. Unif.	204	1,079	1,551	2,539	25,423

The number of fitness evaluations needed to solve the 500-Queens problem for five crossover techniques.

Coloring Hamiltonian Planar Graphs

The vertices are on a sphere's equator.

The edges are randomly drawn to triangulate each hemisphere.

Type 1 graphs: only use the edges in the northern hemisphere.
These are 3-colorable, uniquely.

Type 2 graphs: use the edges in both hemispheres.
These are 4-colorable, right?

Type 1 graphs proved harder to color!



Experiments & Results

190 random graphs of each type, $11 \leq N \leq 200$.

21 attempts to color each.

MOX, pop=20, mut-rate=0.01, max 100,000 fitness evaluations.

Type 1 graphs

$11 \leq N \leq 30$	all success
$31 \leq N \leq 64$	most success
$65 \leq N \leq 110$	< half success
$111 \leq N \leq 200$	all failed

Type 2 graphs

$11 \leq N \leq 50$	all success
$59 \leq N \leq 171$	most success
$172 \leq N \leq 200$	< half success

Every type 2 graph was colored.



Using Warnsdorff to Help Color Graphs

OG + Warnsdorff: color the hardest-to-color first.

Break ties using the permutation.

Result for type 1 graphs: The first attempt to color each one worked.

Type 2 graphs: work in progress.



An OG Applications Sampler

O. G. is natural & appropriate for a variety of problems.

A necessary condition: The optimal solution can be found by a greedy algorithm and the right permutation.

Many permutations will produce the same, or equivalent, answer.



An OG Applications Sampler

- N Queens
- Graph vertex & edge coloring
- Scheduling in general
- Exam scheduling
- Sports tournaments scheduling
- Multiprocessors scheduling
- Faculty teaching assignments
- Job assignments
- Matching
- Traveling salesman
- Bin packing
- 2D board cutting
- Pentominos
- SAT, 3SAT

