Crossover

```
/*-------------------------------------------------------------------------
| Order1 (Starkweather, et. al., 1991 GA Conf.)
|
| "The offspring inherits the elements between the two crossover points,
| inclusive, from the selected parent in the same order and position as they
| appeared in that parent.  The remaining elements are inherited from the
| alternate parent in the order in which they appear in that parent, beginning
| with the first position following the second crossover point and skipping
| over all elements already present in the offspring."
|
| L. Davis. "Applying Adaptive Algorithms to Epistatic Domains". In Proc.
|    International Joint Conference on Artificial Intelligence (1985).
-------------------------------------------------------------------------*/
```

Crossover

```
/*-------------------------------------------------------------------------
| Order2 (Starkweather, et. al., 1991 GA Conf.)
|
| "...several key positions are chosen randomly and the order in which these
| elements appear in one parent is imposed on the other parent to produce
| two offspring..."
|
| G. Syswerda.  "Schedule Optimization Using Genetic Algorithms".  In Handbook
|    of Genetic Algorithms.  L. Davis ed.  Van Nostrand Reinhold: NY (1990).
-------------------------------------------------------------------------*/
```

Crossover

```
/*-------------------------------------------------------------------------
| Position (Starkweather, et. al., 1991 GA Conf.)
|
| "Several random locations in the sequence are selected along with one
| parent; the elements in those positions are inherited from that parent.
| The remaining elements are inherited in the order in which they appear in
| the alternate parent, skipping over all elements which have already been
| included in the offspring."
|
| G. Syswerda.  "Schedule Optimization Using Genetic Algorithms".  In Handbook
|    of Genetic Algorithms.  L. Davis ed.  Van Nostrand Reinhold: NY (1990).
-------------------------------------------------------------------------*/
```

Crossover

```
/*-------------------------------------------------------------------------
| Cycle (Starkweather, et. al., 1991 GA Conf.)
|
| "A parent sequence and a cycle starting point are randomly selected.  The
| element at the cycle starting point of the selected parent is inherited by
| the child.  The element which is in the same position in the other parent
| cannot then be placed in this position so its position is found in the
| selected parent and is inherited from that position by the child.  This
| continues until the cycle is completed by encountering the initial item in
| the unselected parent.  Any elements which are not yet present in the
| offspring are inherited from the unselected parent."
|
| I. Oliver, D. Smith, and J. Holland.  "A Study of Permutation Crossover
|   Operators on the Traveling Salesman Problem."  In Proc. Second Interna-
|   tional Conference on Genetic Algorithms and their Applications (1987).
-------------------------------------------------------------------------*/
```

Crossover

```
/*-------------------------------------------------------------------------
| PMX (Starkweather, et. al., 1991 GA Conf.)
|
| "A parent and two crossover sites are selected randomly and the elements
| between the two starting positions in one of the parents are directly
| inherited by the offspring.  Each element between the two crossover points
| in the alternate parent are mapped to the position held by this element in
| the first parent.  Then the remaining elements are inherited from the
| alternate parent.
|
| D. Goldberg and R. Lingle.  "Alleles, loci, and the Traveling Salesman
| Problem".  In Proc. International Conference on Genetic Algorithms and their
| Applications (1985).
-------------------------------------------------------------------------*/
```

GA Run: Data

```
#-------------------------------------------------------------------------
# The data type of the allele
#
# Usage: datatype [bit | int | int_perm | real]
#
#    bit      = bit string
#    int      = integers
#    int_perm = permutation of integers
#    real     = real numbers
#
# DEFAULT: int_perm
#-------------------------------------------------------------------------
# datatype bit
# datatype int
# datatype int_perm
# datatype real
#-------------------------------------------------------------------------
# Chromosome length, needed when "initpool random" selected
#
# Usage: chrom_len length
#
#    length = chromosome length, a positive integer
#
# DEFAULT: chrom_len 10
#-------------------------------------------------------------------------
# chrom_len 25
```

GA Run: Initialization

```
#------------------------------------------------------------------------
# How to initialize the pool
#
# Usage: initpool [random | from_file filename | interactive]
#
#    random      = generate at random based on
#                      datatype, chrom_len, & pool_size
#    from_file   = read from a file
#       filename = the name of the file to read from
#    interactive = read from stdin
#
# DEFAULT: initpool random
#------------------------------------------------------------------------
# initpool random
# initpool from_file   initpool.dat
# initpool interactive
```

GA Run: Crossover Type

```
#-------------------------------------------------------------------------
# Crossover method:
#
# Usage: crossover [simple | uniform | order1 | order2 | position | cycle |
#                   pmx | uox | rox | asexual]
#
#    simple    = children get alternate "halves" of parents
#    uniform   = alleles swapped uniformly
#    order1    = order based
#    order2    = order based
#    position  = order based
#    cycle     = order based
#    pmx       = order based
#    uox       = uniform order
#    rox       = relative order
#    asexual   = swap two alleles
#
# DEFAULT: crossover order1
#-------------------------------------------------------------------------
# crossover simple
# crossover uniform
# crossover order1            # use ony with integer permutations
# crossover order2            # use ony with integer permutations
# crossover position          # use ony with integer permutations
# crossover cycle             # use ony with integer permutations
# crossover pmx               # use ony with integer permutations
# crossover uox               # use ony with integer permutations
# crossover rox               # use ony with integer permutations
# crossover asexual
```

GA Run: Population Management

```
#-------------------------------------------------------------------------
# Replacement method:
#
# Usage: replacement [append | by_rank | first_weaker | weakest]
#
#    append       = append to new pool, as in generational GA
#    by_rank      = insert in sorted order, as in Genitor
#    first_weaker = replace first weaker found in linear scan of pool
#    weakest      = replace weakest member of the pool
#
# DEFAULT: replacement append
#-------------------------------------------------------------------------
# replacement append           # use with roulette (generational GA)
# replacement by_rank          # use with rank_biased (steady-state GA)
# replacement first_weaker     # experimental
# replacement weakest          # experimental
```

GA Run: Fitness Function

```
/*-------------------------------------------------------------------------
| obj_fun() - user specified objective function
-------------------------------------------------------------------------*/
int obj_fun(chrom)
   Chrom_Ptr chrom;

   int i, how_far_off;
   double val = 0.0, penalty, fudge_factor;

   /*--- Penalty for not being in correct position ---*/
   penalty = 1.0;

   /*--- Fudge factor for variance from optimal
         Ensure this is never more than penalty ---*/
   fudge_factor = (1.0 / (double)(chrom->length * 10.0));

   /*--- Fitness is number of genes out of place for sorted order ---*/
   for(i = 0; i < chrom->length; i++)
      if(chrom->gene[i] != i+1)

         how_far_off = chrom->gene[i] - (i+1);
         if(how_far_off < 0) how_far_off = -how_far_off;

         val += penalty + how_far_off * fudge_factor;



   chrom->fitness = val;
```

The Chromosome

```
/*--- A Chromosome ---*/
typedef struct
   long       magic_cookie;         /* For validation */
   Gene_Ptr   gene;                 /* Encoding */
   int        length;               /* Length of gene */
   double     fitness;              /* Fitness value of chromosome */
   float      ptf;                  /* Percent of total fitness */
   int        index;                /* My index */
   int        idx_min, idx_max;     /* Reserved */
   int        parent_1, parent_2;   /* Indices of parents */
   int        xp1, xp2;             /* Crossover points */
 Chrom_Type, *Chrom_Ptr;
```

The Main Program

```
/*-------------------------------------------------------------------------
| main()
-------------------------------------------------------------------------*/
main(argc, argv)
   int  argc;
   char *argv[];

   GA_Info_Ptr ga_info;

   /*--- Initialize the genetic algorithm ---*/
   ga_info = GA_config("ga-test.cfg", obj_fun);

   /*--- Select crossover ---*/
   if(argc > 1)
      X_select(ga_info, argv[1]);
   ;

   /*--- Run the GA ---*/
   GA_run(ga_info);
```

GA Run: Compiling and Running

```
Library of Functions:
   /software/expert/genetic/libGA.a

Compiling:
gcc -O -Wall -L/software/expert/genetic -o ga-test ga-test.c -lGA -lm

Running:  (Uses: ga-test.cfg)
ga-test
```

GA Run: Output

```
LibGA Version 1.00
(c) Copyright Arthur L. Corcoran, 1992, 1993.  All rights reserved.

GA Configuration Information:
-----------------------------
Basic Info
   Random Seed       : 1
   Data Type         : Integer Permutation
   Init Pool Entered : Randomly
   Chromosome Length : 10
   Pool Size         : 200
   Number of Trials  : Run until convergence
   Minimize          : Yes
   Elitism           : Yes
   Scale Factor      : 0

Functions
   GA          : generational (Gap = 0)
   Selection   : roulette
   Crossover   : order1 (Rate = 1)
   Replacement : append

Reports
   Type     : Short
   Interval : 1
-----------------------------
```

GA Run: Output

Gener	Min	Max	Ave	Variance	Std Dev	Tot Fit	Best
0	4.16	10.46	9.29	1.3	1.14	1858.3	4.16
1	4.1	10.44	8.36	1.79	1.34	1671.8	4.1
2	4.1	10.34	7.38	2.02	1.42	1475.96	4.1
3	4.1	9.38	6.54	2.29	1.51	1307.3	4.1
4	3.08	9.28	5.79	2.15	1.47	1158.64	3.08
5	3.08	8.24	5.06	1.41	1.19	1012.64	3.08
6	2.14	7.32	4.55	1.19	1.09	909.38	2.14
7	2.14	7.2	4.12	1.06	1.03	824.68	2.14
8	2.14	6.26	3.55	0.606	0.779	710.22	2.14
9	2.14	6.22	3.24	0.452	0.672	648.8	2.14
10	2.02	6.16	3.02	0.286	0.535	603.88	2.02
11	2.02	4.16	2.87	0.231	0.481	573.22	2.02
12	2.02	4.16	2.72	0.252	0.502	544.62	2.02
13	2.02	4.16	2.56	0.234	0.484	511.76	2.02
14	0	4.14	2.4	0.231	0.481	480.2	0
New scale factor = 1							
15	0	3.08	2.17	0.403	0.635	433.68	0
16	0	3.08	1.71	0.952	0.976	342.86	0
17	0	3.08	0.699	1.06	1.03	139.88	0
18	0	2.14	0.127	0.255	0.505	25.44	0
19	0	2.14	0.0422	0.0878	0.296	8.44	0
20	0	0	0	0	0	0	0

```
The GA has converged after 20 iterations.

Best: 1 2 3 4 5 6 7 8 9 10  (0)
```

GA Run: 0th Generation

```
================================================================================
Generation 0: Mutations = 0 (0 total)

 # Parents  XP   Fitness  String
-- ------- ----- -------  ------
 1 ( 0, 0)  0  0    7.26  2 9 7 4 3 6 5 8 10 1
 2 ( 0, 0)  0  0     9.4  7 8 5 10 4 2 6 3 9 1
 3 ( 0, 0)  0  0    7.22  4 1 7 8 3 6 2 5 9 10
 4 ( 0, 0)  0  0    8.24  1 4 3 10 7 5 9 2 8 6
 5 ( 0, 0)  0  0     9.3  7 10 2 3 5 4 8 1 6 9
 6 ( 0, 0)  0  0    10.2  4 1 5 2 8 7 3 6 10 9
 7 ( 0, 0)  0  0   10.44  5 6 7 10 3 8 9 4 1 2
 8 ( 0, 0)  0  0    9.36  4 5 3 9 10 8 2 1 6 7
 9 ( 0, 0)  0  0    10.3  3 5 1 6 10 9 2 7 4 8
10 ( 0, 0)  0  0    7.18  1 9 3 5 2 4 7 6 10 8
.
.
.
198 ( 0, 0)  0  0     5.1  1 2 3 8 4 6 7 5 10 9
199 ( 0, 0)  0  0    10.3  10 6 1 2 3 8 4 7 5 9
200 ( 0, 0)  0  0    8.28  8 6 3 5 2 7 1 9 4 10

Min= 4.16   Max= 10.46   Ave= 9.3   Tot= 1858.3   Var= 1.3   SD= 1.1

Best: 5 2 3 8 4 6 7 1 9 10 (4.16)
================================================================================
```

GA Run: 1st Generation

```
================================================================================
Generation 1: Mutations = 0 (0 total)

 # Parents  XP   Fitness  String
-- ------- ----- -------  ------
 1 ( 0, 0)  0  0    4.16  5 2 3 8 4 6 7 1 9 10
 2 ( 0, 0)  0  0    4.16  5 2 3 8 4 6 7 1 9 10
 3 (90,187)  7  8    7.34  10 2 6 8 5 3 7 9 4 1
 4 (90,187)  7  8    9.28  2 6 8 5 3 9 4 7 1 10
 5 (52,10)  3 10    7.18  1 9 3 5 2 4 7 6 10 8
 6 (52,10)  3 10     8.2  9 1 3 5 2 4 7 6 10 8
 7 (69,72)  2  7    8.22  1 3 6 5 10 7 2 4 9 8
 8 (69,72)  2  7     9.3  2 7 6 3 9 8 1 4 5 10
 9 (98, 6)  7 10    9.36  4 7 3 6 10 9 5 2 8 1
10 (98, 6)  7 10    10.2  4 1 5 2 8 7 3 6 10 9
.
.
.
197 (12,128)  5  7    8.26  9 5 2 4 1 8 6 3 7 10
198 (12,128)  5  7     8.3  5 2 4 10 7 1 9 8 6 3
199 (103,73)  2  3    9.38  7 3 9 10 5 2 1 4 6 8
200 (103,73)  2  3    9.48  9 8 7 10 5 2 1 4 6 3

Min= 4.1   Max= 10.44   Ave= 8.4   Tot= 1671.8   Var= 1.8   SD= 1.3

Best: 1 2 3 8 4 6 7 9 5 10 (4.1)
================================================================================
```

```
 # Parents  XP   Fitness  String
-- ------- ----- -------  ------
 1 (189,50)  7 10    2.02  1 2 3 4 6 5 7 8 9 10
 2 (189,50)  7 10    2.02  1 2 3 4 6 5 7 8 9 10
 3 (105,161)  9 10    2.14  8 2 3 4 5 6 7 1 9 10
 4 (105,161)  9 10    2.14  8 2 3 4 5 6 7 1 9 10
 5 (115,143)  8 10    2.14  8 2 3 4 5 6 7 1 9 10
 6 (115,143)  8 10    2.14  8 2 3 4 5 6 7 1 9 10
 7 (113,13)  4  8    3.08  1 2 3 8 4 6 7 5 9 10
 8 (113,13)  4  8    3.08  1 2 3 8 4 6 7 5 9 10
 9 (178,120)  1  2    2.14  8 2 3 4 5 6 7 1 9 10
10 (178,120)  1  2    2.14  8 2 3 4 5 6 7 1 9 10

196 (84,128)  1  7    2.14  8 2 3 4 5 6 7 1 9 10
197 (149,187)  9 10    2.14  8 2 3 4 5 6 7 1 9 10
198 (149,187)  9 10    2.14  8 2 3 4 5 6 7 1 9 10
199 (137,152)  4  4    2.14  8 2 3 4 5 6 7 1 9 10
200 (137,152)  4  4    3.08  1 2 3 8 4 6 7 5 9 10

Min= 0   Max= 4.14   Ave= 2.4   Tot= 480.2   Var= 0.23   SD= 0.48

Best: 1 2 3 4 5 6 7 8 9 10 (0)
================================================================================
New scale factor = 1
```

GA Run: Best Element

```
Best: 5 2 3 8 4 6 7 1 9 10 (4.16)
Min= 4.16   Max= 10.46   Ave= 9.3   Tot= 1858.3   Var= 1.3   SD= 1.1
Best: 1 2 3 8 4 6 7 9 5 10 (4.1)
Min= 4.1   Max= 10.44   Ave= 8.4   Tot= 1671.8   Var= 1.8   SD= 1.3
Best: 1 2 3 8 4 6 7 9 5 10 (4.1)
Min= 4.1   Max= 10.34   Ave= 7.4   Tot= 1475.96   Var= 2   SD= 1.4
.
.
.
Best: 1 2 3 4 6 5 7 8 9 10 (2.02)
Min= 2.02   Max= 6.16   Ave= 3   Tot= 603.88   Var= 0.29   SD= 0.53
Best: 1 2 3 4 6 5 7 8 9 10 (2.02)
Min= 2.02   Max= 4.16   Ave= 2.9   Tot= 573.22   Var= 0.23   SD= 0.48
Best: 1 2 3 4 6 5 7 8 9 10 (2.02)
Min= 2.02   Max= 4.16   Ave= 2.7   Tot= 544.62   Var= 0.25   SD= 0.5
Best: 1 2 3 4 6 5 7 8 9 10 (2.02)
Min= 2.02   Max= 4.16   Ave= 2.6   Tot= 511.76   Var= 0.23   SD= 0.48
Best: 1 2 3 4 5 6 7 8 9 10 (0)
Min= 0   Max= 4.14   Ave= 2.4   Tot= 480.2   Var= 0.23   SD= 0.48
Best: 1 2 3 4 5 6 7 8 9 10 (0)
Min= 0   Max= 3.08   Ave= 2.2   Tot= 433.68   Var= 0.4   SD= 0.63
.
.
.
Best: 1 2 3 4 5 6 7 8 9 10 (0)
Min= 0   Max= 2.14   Ave= 0.042   Tot= 8.44   Var= 0.088   SD= 0.3
Best: 1 2 3 4 5 6 7 8 9 10  (0)
```