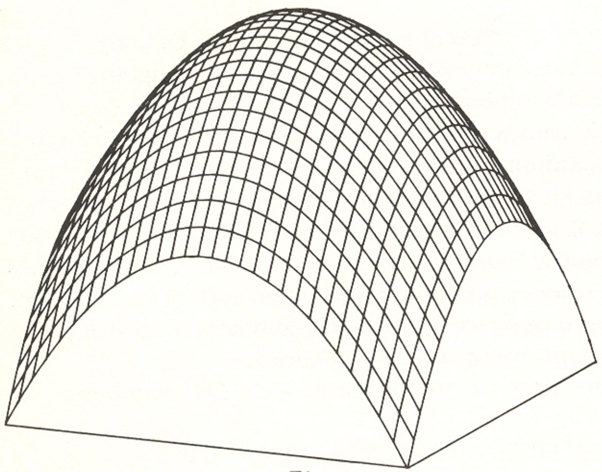
Applications of GAs

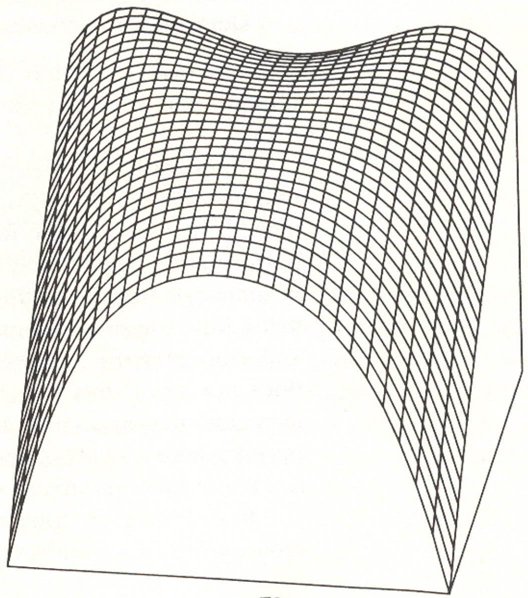
De Jong and Function Optimization

Five test problems in function minimizations:

See Figures 4.9-4.11



F1

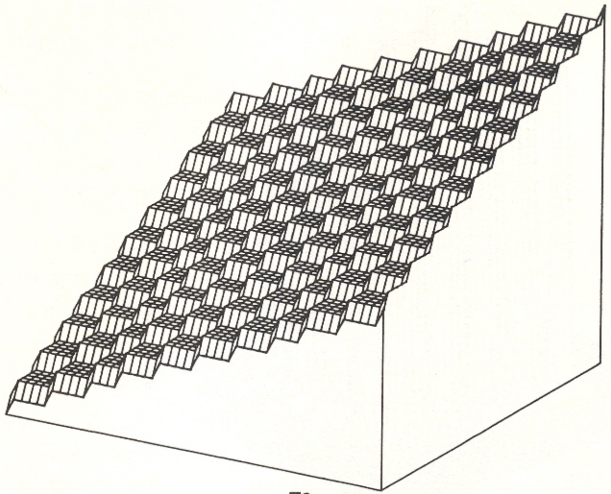


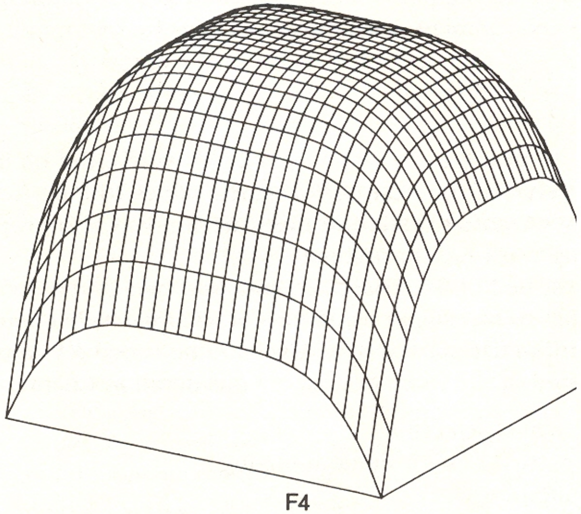
F2

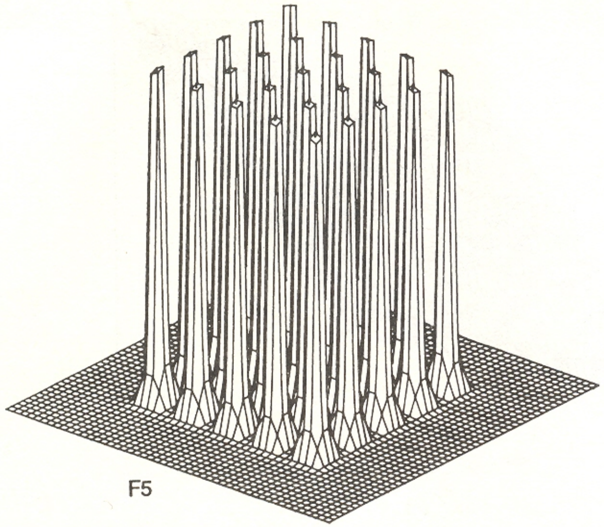
De Jong and Function Optimization

Five test problems in function minimizations:

See Figures 4.9-4.11


F3


F4


F5

De Jong and Function Optimization

Five test problems in function minimizations:

See Figures 4.9-4.11

Two measures to quantify the effectiveness of different GAs

On-line performance: Average of all function evaluations

$x_e(s)=1/T\ \Sigma f_e(t)$

$f_e(t)$: objective function value for environment e on trial t

Off-line performance: running average of the best

$x_e^*(s) = 1/T\Sigma f_e^*(t)$

$f_e^*(t) = \text{best}\{f_e(1), f_e(2),...,f_e(t)\}$

Variations of Simple GA

R1 (reproductive plan 1):

Roulette wheel selection

Simple crossover (with random mate)

Simple mutation

R1 depends on four parameters

n = population size

p_c = crossover probability

p_m = mutation probability

G = generation gap

$G=1$	nonoverlapping populations
$0<G<1$	overlapping population

Variations of Simple GA

Overlapping population

nG individuals are selected by roulette wheel for further GA operation.

The remaining positions in A(t+1) are filled with individuals from A(t) without replacement using a uniform distribution

Simple GA (R1)

Results: See Figures 4.12-4.17.

1. Larger populations lead to better ultimate off-line performance, even if the initial on-line performance is poor.
2. Increased mutation rate decreases the number of lost alleles at the expense of degraded off-line and on-line performance.
3. p_c =0.6 is a reasonable compromise between good on-line and off-line performance.
4. Higher crossover rates (p_c=1.0) are better when the stochastic errors of sampling are reduced through the use of more accurate selection procedures.
5. Nonoverlapping population model is best in most optimization studies, where off-line performance tends to be the overriding concern.

Simple GA (R1)

Results: See Figures 4.12-4.17.

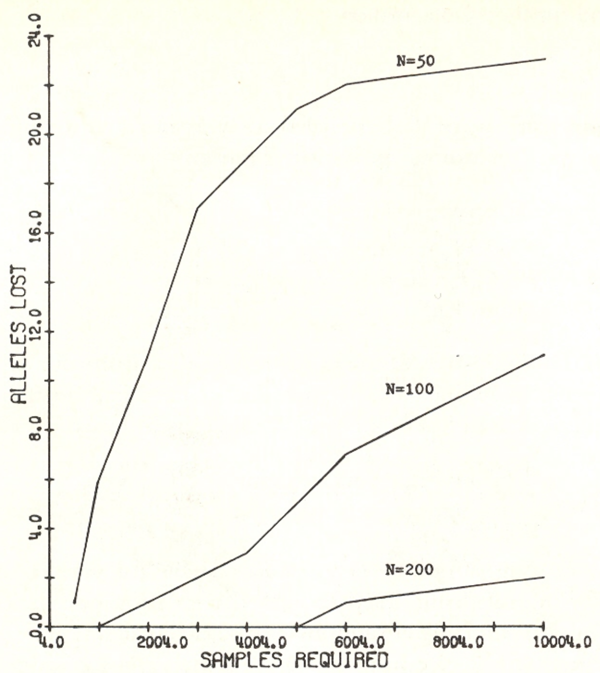


FIGURE 4.12 The effects of population size on allele loss for plan *R*1 on f
tion *F*1 (De Jong, 1975). Reprinted by permission.

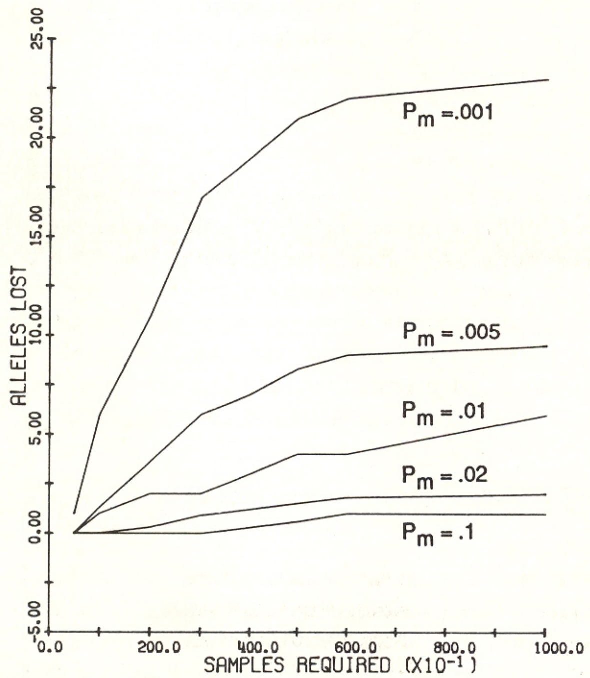


FIGURE 4.15 The effects of mutation rate on allele loss for plan *R*1 on function *F*1 (De Jong, 1975). Reprinted by permission.

Simple GA (R1)

Results: See Figures 4.12-4.17.

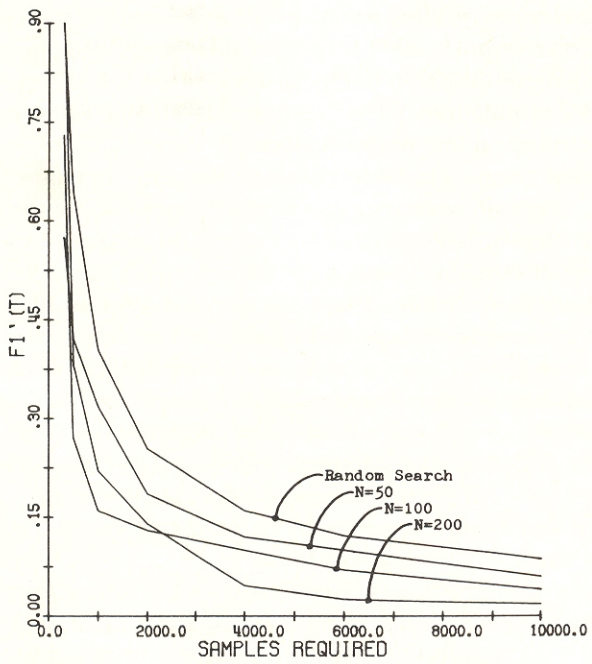


FIGURE 4.13 The effects of population size on off-line performance of pla
on function *F*1 (De Jong, 1975). Reprinted by permission.

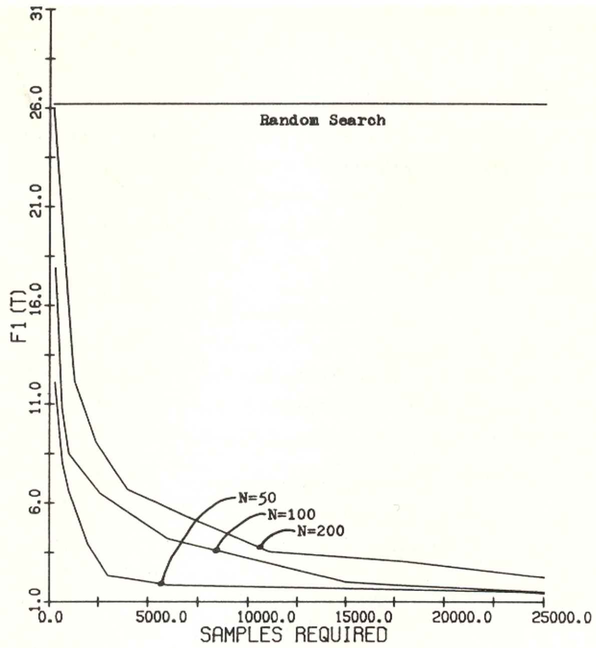


FIGURE 4.14 The effects of population size on on-line performance of plan *R*1 on function *F*1 (De Jong, 1975). Reprinted by permission.

Simple GA (R1)

Results: See Figures 4.12-4.17.

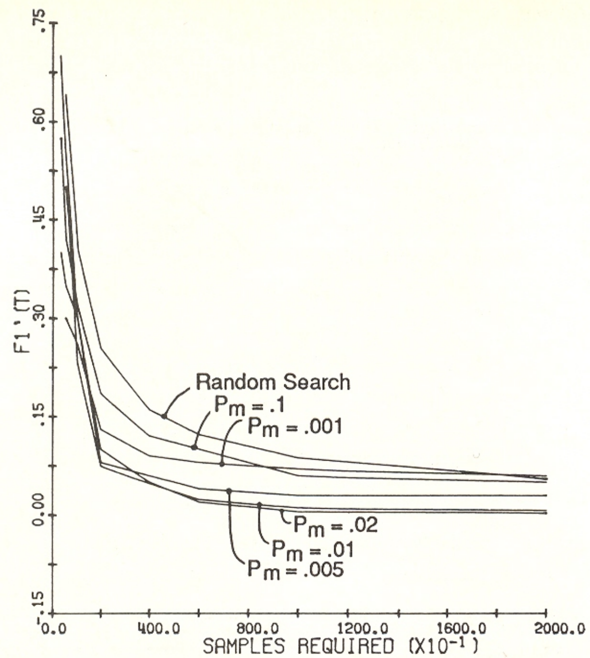


FIGURE 4.16 The effects of mutation rate on off-line performance function $F1$ (De Jong, 1975). Reprinted by permission.

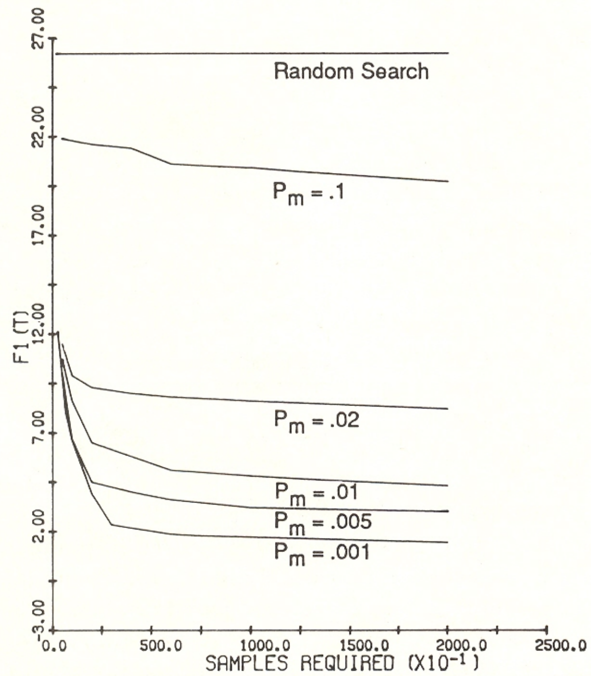


FIGURE 4.17 The effects of mutation rate on on-line performance of plan $R1$ on function $F1$ (De Jong, 1975). Reprinted by permission.

Variations of plan R1

R2: Elitist model

R3: Expected value model

R4: Elitist expected value model

R5: Crowding factor model

R6: Generalized crossover model

Elitist model (R2)

After generating A(t+1), if it does not have the best individual generated up to time t, then the best is included to A(t+1).

Significantly improve both on-line and off-line performance on unimodal surface.

On multimodal function F5, it degrades both performance measures.

Expected value model (R3)

Designed to reduce the stochastic errors of roulette wheel selection.

Compute the expected number of offspring for each string (f/f_{avg}):

Each time a string is selected for mating and crossover, its offspring count is decreased by 0.5.

When an individual string is selected for reproduction without mating and crossover, its offspring count is decreased by 1.0.

In either case an individual whose offspring count fell below zero is no longer available for selection.

In this model the actual number of offspring is generally less than $f/f_{avg}+ 0.5$.

R3 outperforms R1 and R2 in both on-line and off-line performance measures over the environment E (functions F1-F5).

Elitist expected value model (R4)

Considerable improvement is observed in unimodal function (F1-F4), while on the difficult foxhole function F5, performance is degraded over the expected value plan alone.

The global robustness measures are superior to R2 and R3.

Crowding factor model (R5)

To enforce crowding pressure in artificial GAs, newly generated offspring is forced to replace similar, older adults in the hope of maintaining more diversity in the population.

Produce A(t+1) by adopting overlapping population with generation gap G=0.1.

Produce nG offspring using roulette wheel selection and the genetic operators.

Insert the nG offspring into A(t) by selecting nG individuals to die.

The dying individual is selected from a subset of CF members chosen from A(t) at random.

The individual that most closely resembles the new offspring is chosen to die.

Results: See Figure 4.21.

Generalized crossover model (R6)

The best off-line performance is achieved with CP=1, 2.

Multiple-point crossover degrades off-line and on-line performance.

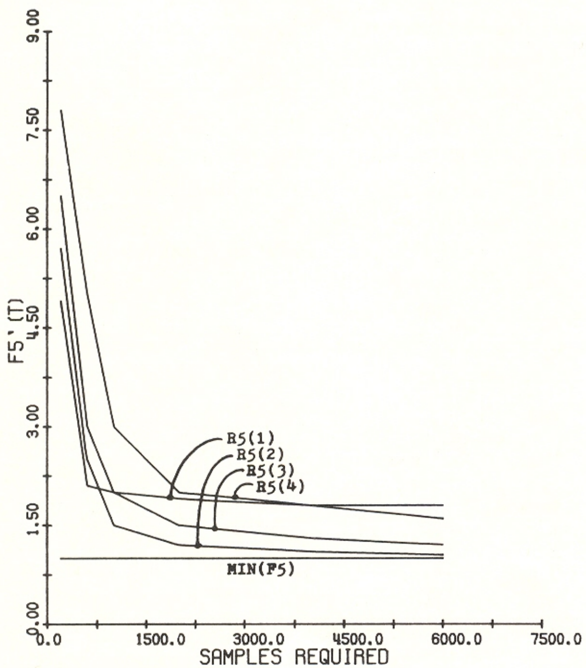


FIGURE 4.21 Effect of crowding factor on off-line performance, plan *R*5 on function *F*5 (De Jong, 1975). Reprinted by permission.

Alternate selection schemes

1. Stochastic sampling with replacement: Roulette wheel selection
2. Stochastic sampling without replacement: Expected value model (R3)
3. Deterministic sampling
   - Each string is copied according to the integer part of the expected number of offspring
   - The remainder of the population is filled with strings in the order of high fractional part
4. Remainder stochastic sampling with replacement
5. Remainder stochastic sampling without replacement

Roulette wheel << Expected value model

< Remainder stochastic without Replacement

Alternate selection schemes

6. Stochastic tournament: draw a pair of strings using roulette wheel
7. Tournament: draw a pair of strings uniform randomly

   GENITOR (Gordon and Whitley, 1993)

      Select the better s^1 among two strings s_1, s_2

      Select the better s^2 among two strings s_3, s_4

      Produce c^1, c^2 after crossover s^1, s^2

      Replace the looser of the tournament if c^1, c^2 has higher fitness

Linear Ranking Selection

(Step1) Sort individuals according to their fitness values. Rank 1 to the best and N to the worst.

Let selection probability

$p_i = \{\eta^+ + (\eta^- - \eta^+)(i\text{-}1)/(N\text{-}1)\}/N, \ i=1,\ldots,N$

η^+/N = Probability of best individual to be selected

η^-/N = Probability of worst individual to be selected

(Step2) $Sum_0 = 0$

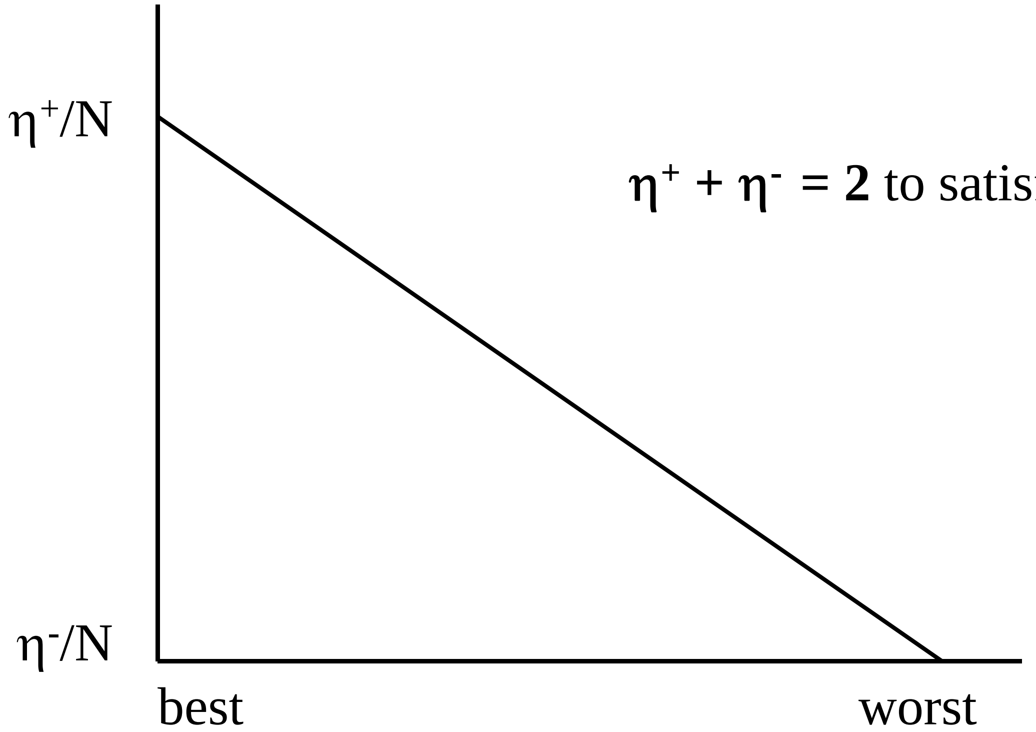
$Sum_i = Sum_{i-1} + p_i \quad i=1,\ldots,N$

(Step3) Generate uniform random number $r \in [0, 1]$.

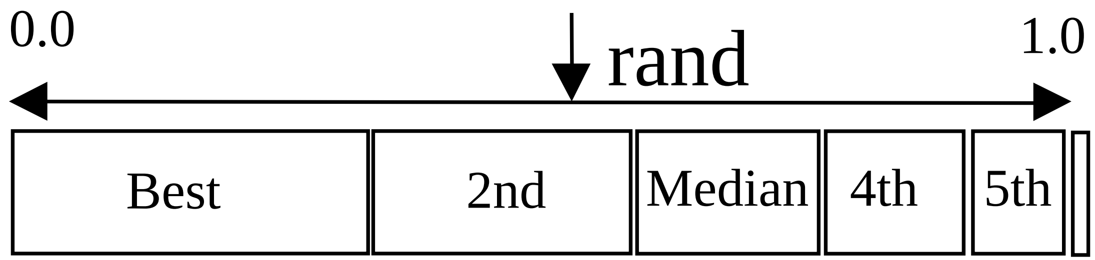
Select individual l, if $Sum_{l-1} \leq r < Sum_l$.

Repeat for N individuals.

Linear Ranking Selection



$\boldsymbol{\eta^+ + \eta^- = 2}$ to satisfy $\Sigma p_i = 1$



Linear Ranking Selection

GENITOR (Gordon and Whitley, 1993)

- Select two parents for crossover
- Produce one offspring and replace it with the worst in the population

Linear Normalization (fitness in Ranking Procedures)

Original Evaluation	200	9	8	8	4	1
Linear fitness with decrement = 1	100	99	98	97	96	95
Linear fitness with decrement = 20	100	80	60	40	20	1

Scaling Mechanism

Scaling mechanism keeps appropriate levels of competition through the generation.

In earlier generation, a few super-individuals tend to dominate the process.

Objective function values must be scaled back to prevent takeover of the population by the super-strings.

In later generation, when the population is largely converged, competition among population members is less strong and the process tends to wander.

Objective function values must be scaled up to accentuate differences among population members to continue to reward the best performers.

Linear scaling

$f' = af + b$

a and b are chosen to meet two things

1. raw average fitness = scaled average fitness
2. maximum scaled fitness = k * scaled average fitness (k=2)

Average population members receive one offspring copy on average and the best receive k multiple member of copies.