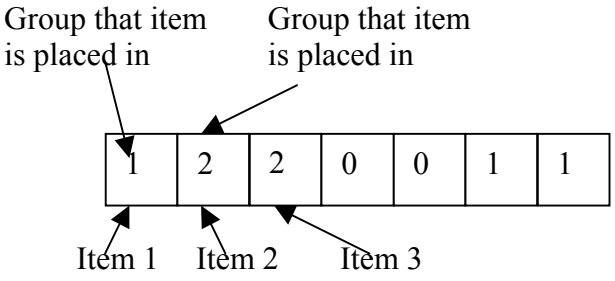
Notes on Program Code

Representation

In the code you are given, a direct representation is used to represent a candidate solution to a problem. If there are **n** items to be grouped into **g** groups, then a solution is represented as below:

There are **n** "genes", one for each item. Each value of each gene is the group that the item is to be placed in. Therefore genes take values between 0 and (g-1).



So, in the above example, there are 7 items, which must be placed into 3 groups. Item 1 goes in group 1, item, item 2 goes in group 2, item 3 goes in group 2, …, and item 7 goes in group 1. It doesn't matter how items are distributed amongst groups, and how many items go in each group – any solution described in this way is valid (though not necessarily very good!)

For example:

If you use an input problem files with **500 items** (e.g. input1n.epp*) in which you must group the items into **10 groups**, a 2D array is set up with the 1st dimension equal to the size of the population, and the second dimension size 500 (corresponding to the number of items). Each gene has a value between 0 and 9 corresponding to the group it is placed in.

Items are numbered from 0 to 499

Groups are numbered from 0 to 9

For example:

- population[2][6] = 9; the 2nd member of the population has item 6 placed in group 9
- population[45][32] = 0; the 45th member of the population has item 32 placed in group 0
- population[99][467] = 4; the 99th member of the population has item 467 placed in group 4

Selection:

Tournament – in this method, n chromosomes are randomly selected and placed in a pool. The selection method returns the best chromosome from the pool. The size of the tournament (the pool) can be altered in the code.

Mutation

Every gene in the chromosome is considered for mutation. The gene is mutated with a probability defined by the rate entered in the application (which should be between 0 and 1). For example, if the mutation rate is set to 0.01, there is a 1 in 100 chance each gene is mutated. If the rate is set to 1.0, then every gene will be mutated. The mutation operator randomly selects a new group for the item being mutated.

Crossover

Three crossover operators are available:

a) *Uniform crossover* – forms a child by choosing the group an item should be placed in from either parent 1 or parent at random.
b) *1-point* – a random position within the chromosome is selected, say position p. The child is formed by copying the groups for items 1 to (p-1) from parent 1, and the groups for items p to n from parent 2.
c) *2-point* – two random positions are chosen, p1 and p2. The child is formed by copying the groups for items 1 to (p1-1) from parent1, for items p1 to (p2-1) from parent2, and for items p2 to n from parent 1