




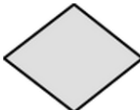




Διαγράμματα ροής και Ψευδοκώδικας

Διαγράμματα ροής

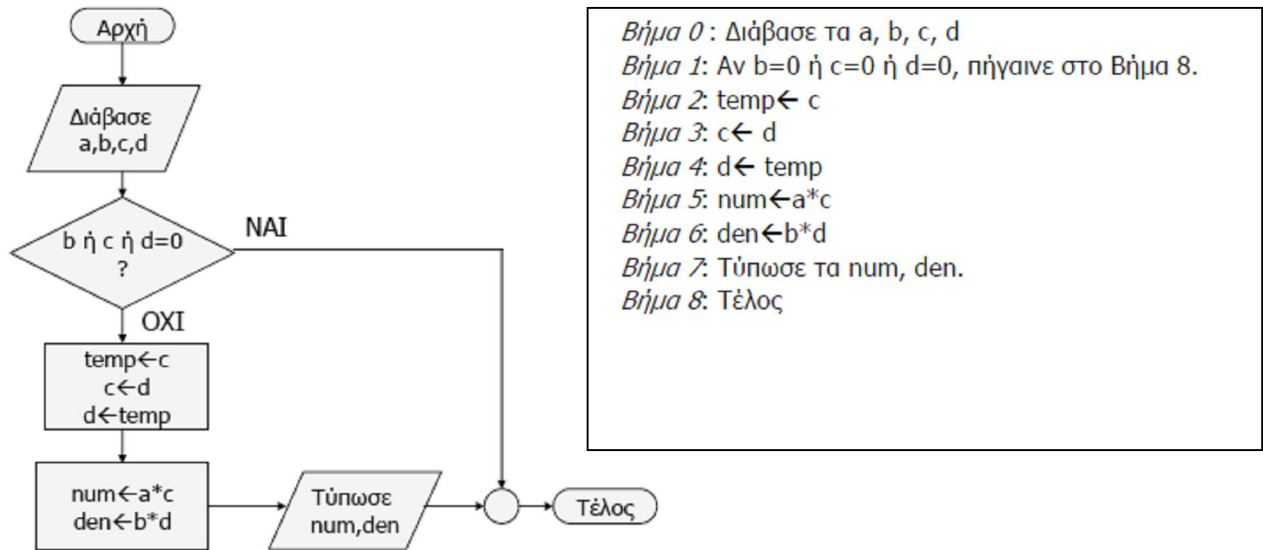
Ένα διάγραμμα ροής αποτελείται από σύμβολα και λέξεις οι οποίες περιγράφουν με λεπτομέρεια κάθε διαδικασία επίλυσης οποιουδήποτε προβλήματος.

Σύμβολο Διαγράμματος ροής

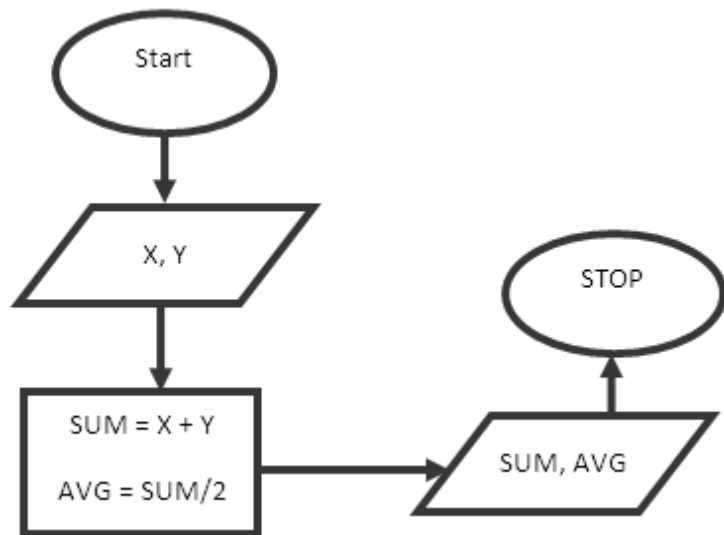
Επεξήγηση

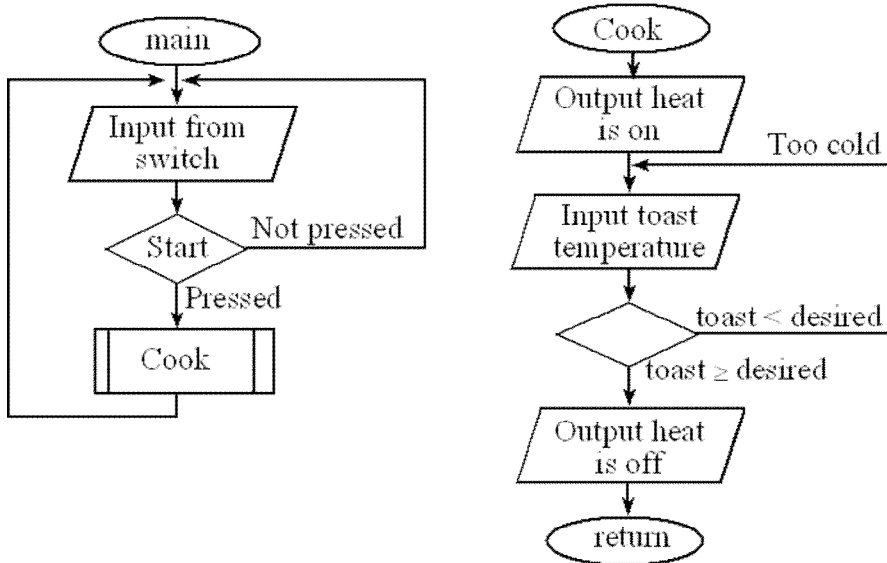
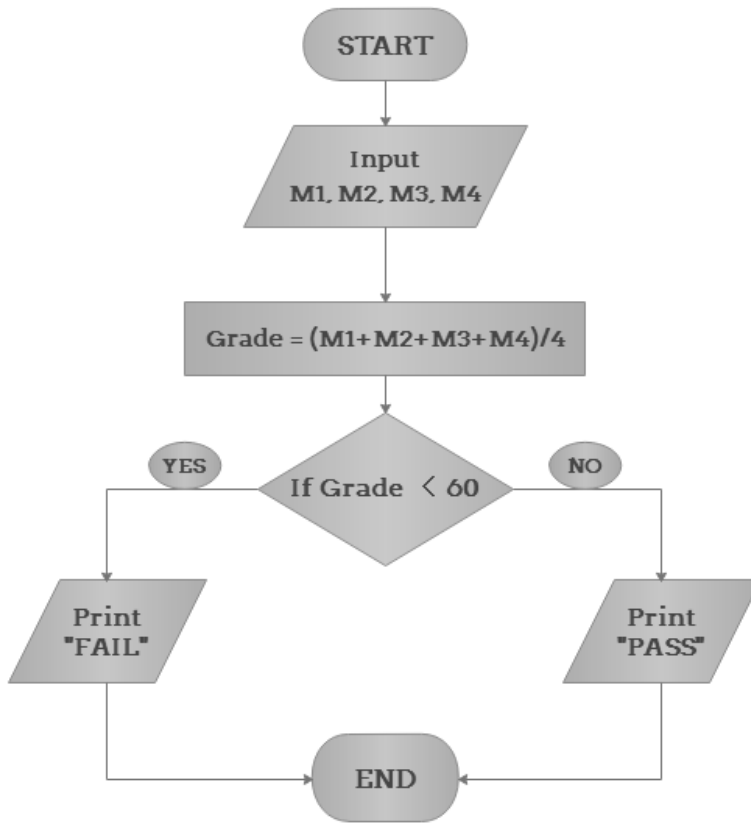
	Αρχή ή τέλος των ενεργειών του προγράμματος
	Είσοδος ή έξοδος δεδομένων από οποιοδήποτε μέσο
	Επεξεργασία στοιχείων, υπολογισμοί
	Απόφαση με 2 ή περισσότερες εξόδους
	Έξοδος αποτελεσμάτων σε αρχείο στον σκληρό δίσκο
	Υποδηλώνει αλγόριθμο εκτός του προγράμματος (subroutine, module, συνάρτηση)
	Σύνδεσμοι (ροή) μεταξύ των συμβόλων, οι οποίοι δηλώνουν την κατεύθυνση εκτέλεσης του πρόγραμματος.
	Ένωση

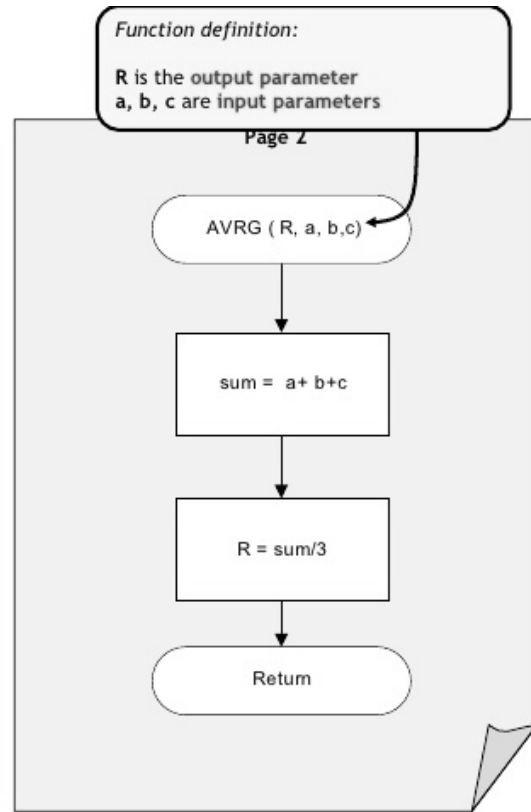
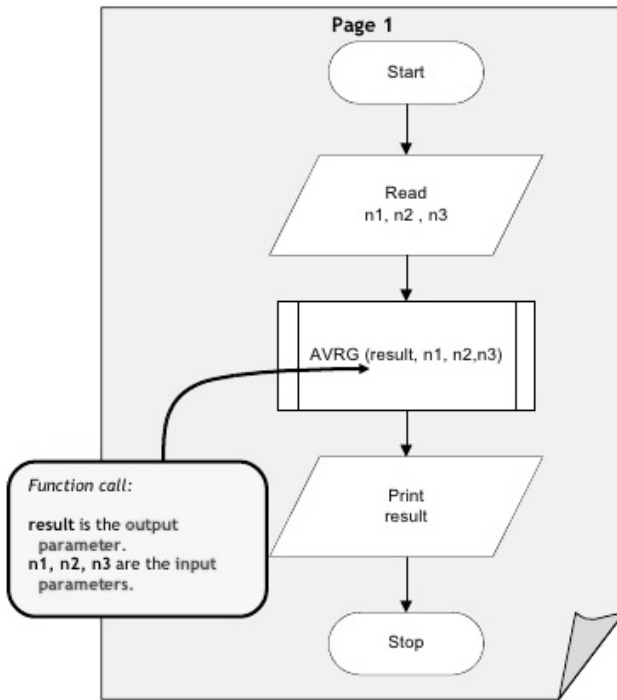
Τα βασικότερα σύμβολα που χρησιμοποιούνται σε ένα διάγραμμα ροής είναι η *έλλειψη*, το *ορθογώνιο*, το *πλάγιο παραλληλόγραμμα*, ο *ρόμβος* και τα *βέλη*. Τα *ορθογώνια παραλληλόγραμμα* συμβολίζουν επεξεργασία. Τα *πλάγια παραλληλόγραμμα* είσοδο και έξοδο δεδομένων, ενώ οι *ρόμβοι* αποτελούν σημεία στα οποία πρέπει να ληφθεί κάποια απόφαση. Με τα *βέλη* απεικονίζουμε τη ροή του προγράμματος. Το *ορθογώνιο παραλληλόγραμμα* χρησιμοποιείται για να περιγραφεί κάποια διαδικασία.

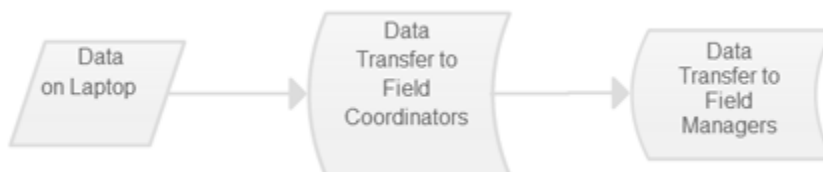
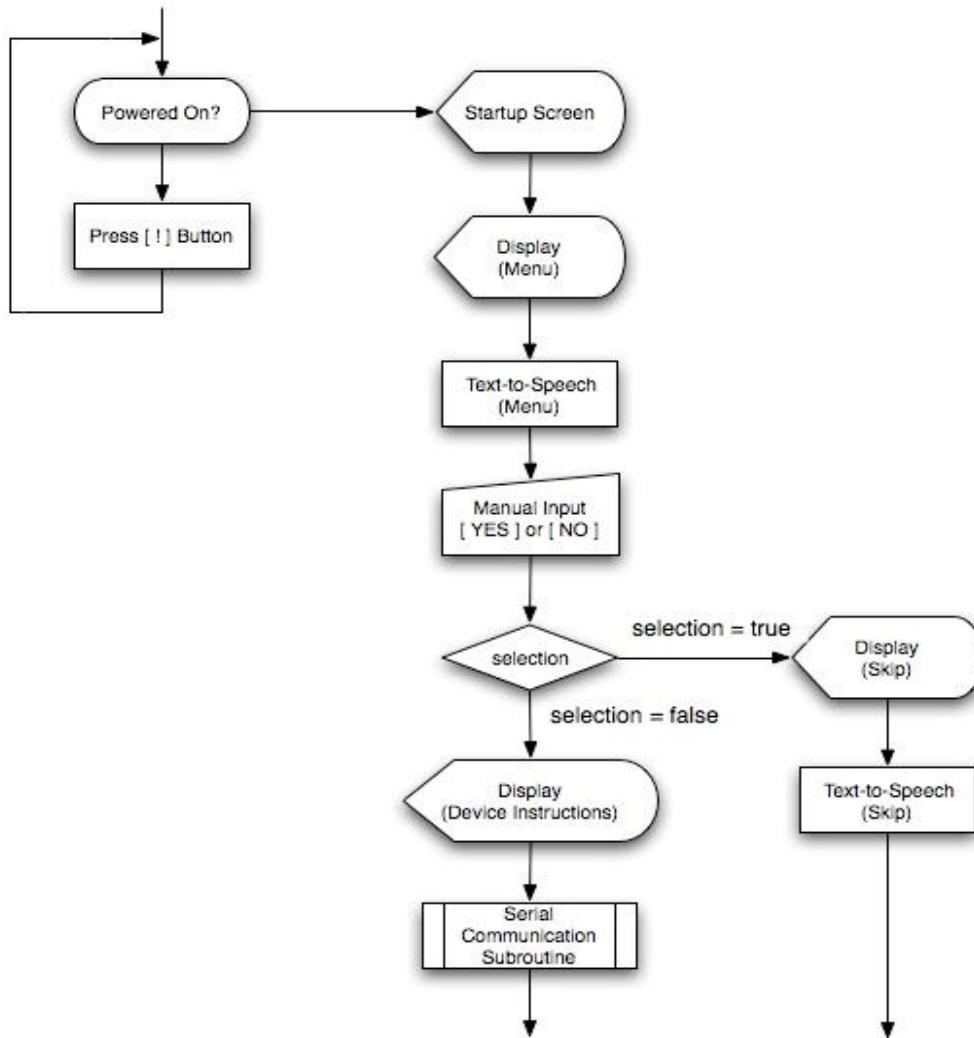


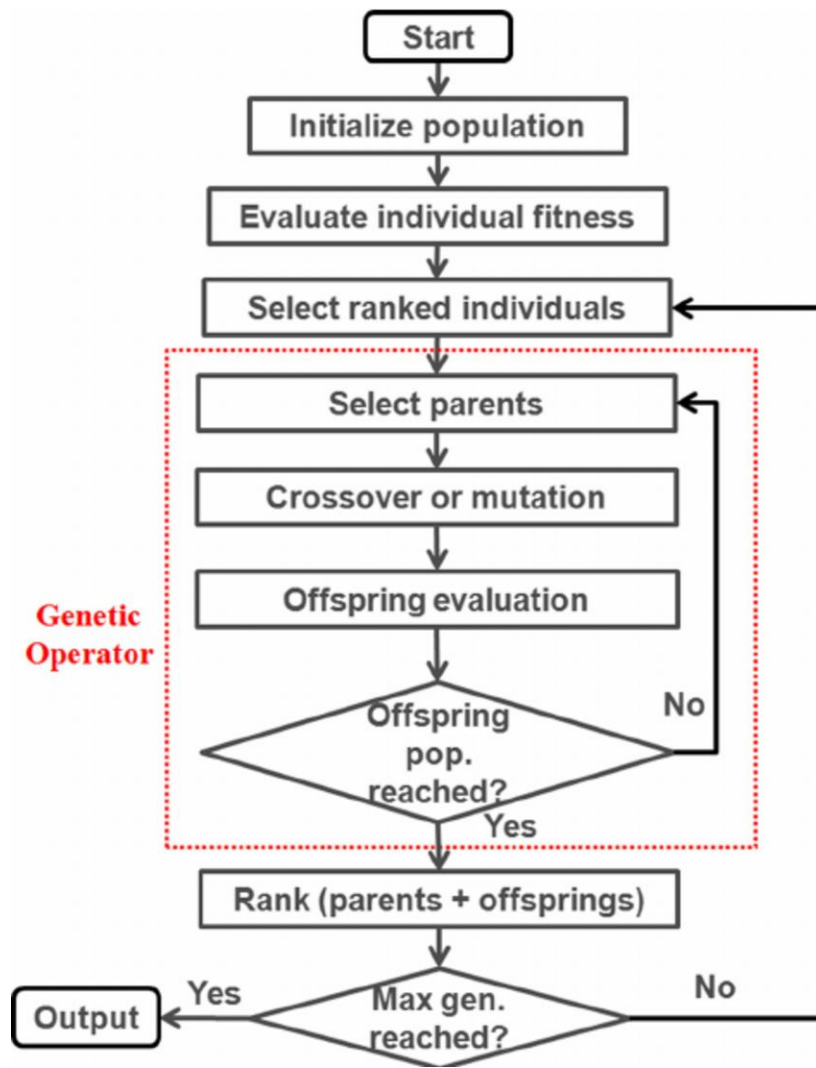
Παραδείγματα flowchart





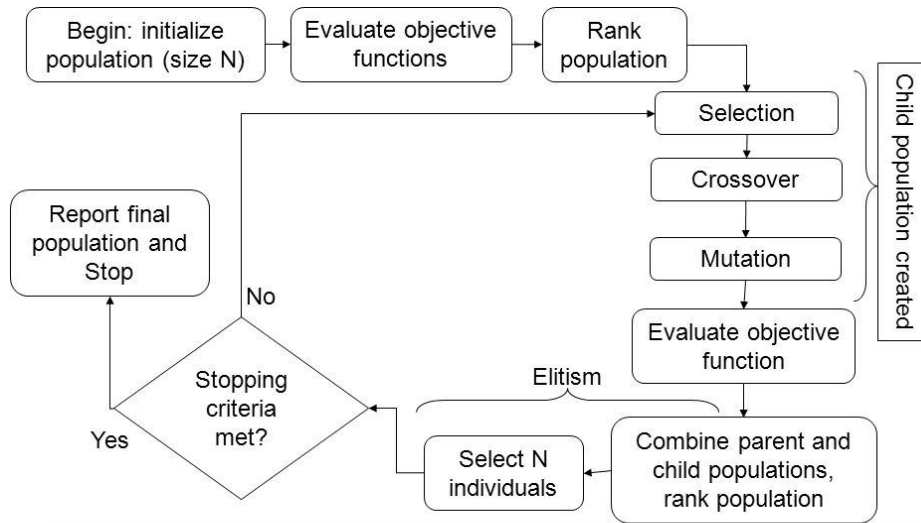






Flowchart NSGAII

Flowchart NSGAI

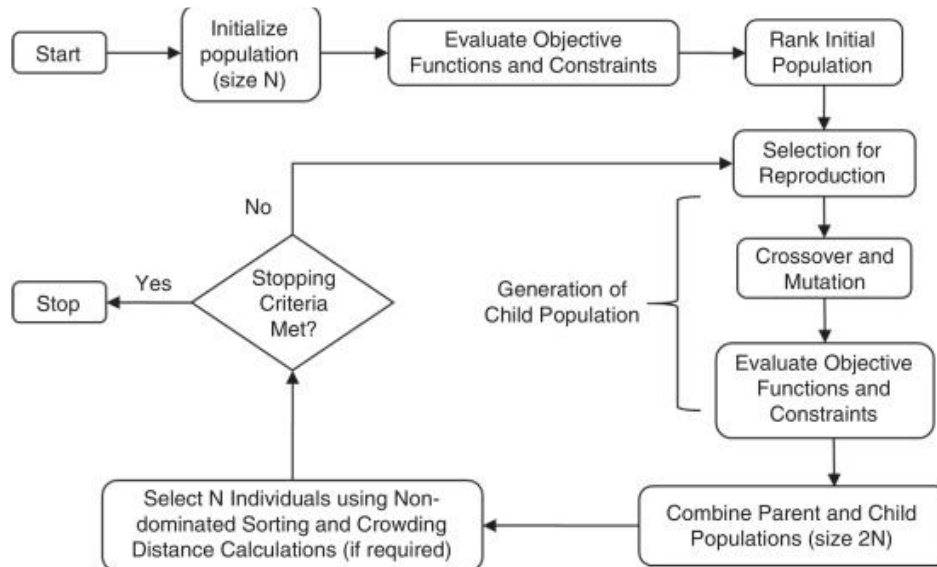


Pseudocode NSGAI

Initialize Population
 Generate N random solutions and insert into Population

for ($i = 1$ to MaxGenerations) **do**
 Generate ChildPopulation of size N
 Select Parents from Population
 Create Children from Parents
 Mutate Children
 Combine Population and ChildPopulations into CurrentPopulation with size $2N$
 for each individual in CurrentPopulation do
 Assign rank based on Pareto – Fast non-dominated sort
 end for
 Generate sets of non-dominated vectors along PF_{known}
 Loop (inside) by adding solutions to next generation of Population starting from the best front
 until N solutions found and determine crowding distance between points on each front
 end for
 Present results

Flowchart NSGAI



Pseudocode NSGAI

```

procedure NSGA-II
  Generate a random initial population ( $N$ -size)
  Evaluate objectives
  Rank initial population based on Pareto dominance
  Compute the crowding distance for each solution in the initial population
  for  $i = 1$  to Number of Epochs
    Generate offspring (through binary tournament, crossover and mutation)
    Mix current population and offspring
    Rank mixed population based on Pareto dominance
    Compute the crowding distance for each solution in the mixed population
    Select the  $N$  best individuals from the mixed population to form the new one
  end for
end procedure
  
```


Παραδείγματα Pseudocode

```

1:  $P = \text{solution.population}()$ ; // Population of candidate solutions
2:   for  $i=0$  to  $\text{solution.population}()$ ;
3:     for  $j=0$  to  $\text{solution.length}()$ ; // length of solution vector:  $K_{max}$  genes
4:       if  $\text{rand}(0, 1) \leq P_m$  then //  $P_m$  is a user-specified parameter named mutation probability
5:          $\text{solution.set}(j, \text{random\_asset}(1, N))$ ; //  $N$  is the number of assets available
6:       endif
7:     endfor
8:   endfor

```

Fig. 3. Pseudo code of the proposed mutation operator.



Fig. 4. The proposed mutation operator.

Procedure NSGA-II

Input: $N', g, f_k(X) \triangleright N'$ members evolved g generations to solve $f_k(X)$

- 1 Initialize Population P' ;
- 2 Generate random population - size N' ;
- 3 Evaluate Objectives Values;
- 4 Assign Rank (level) based on Pareto - *sort*;
- 5 Generate Child Population;
- 6 Binary Tournament Selection;
- 7 Recombination and Mutation;
- 8 **for** $i = 1$ to g **do**
 - 9 **for each** Parent and Child in Population **do**
 - 10 Assign Rank (level) based on Pareto - *sort*;
 - 11 Generate sets of nondominated solutions;
 - 12 Determine Crowding distance;
 - 13 Loop (inside) by adding solutions to next generation starting from the *first* front until N' individuals;
 - 14 **end**
 - 15 Select points on the lower front with high crowding distance;
 - 16 Create next generation;
 - 17 Binary Tournament Selection;
 - 18 Recombination and Mutation;
- 19 **end**
