Algorithm 2 GAFP.

```
population = InitializePopulation(populationSize, function);
chromosomeLength=GetNumberOfInputs(function);
encodedZeroPolarity=” ”;
for $(i = 0; i < chromosomeLength; i++)$ do
    encodedZeroPolarity+=”0”;
zeroPolarityExpansion=GenerateFixedPolarity(function,encodedZeroPolarity);
for $(i = 0; i < populationSize; i++)$ do
    population[i].Expansion=GenerateFixedPolarity(
        zeroPolarityExpansion,
        population[i].Chromosome);
    population[i].Fitness=Evaluate(population[i].Expansion);
for $(i = 0; i < NumberOfIterations; i++)$ do
    offsprings=$\phi$
    parents = RouletteWheelSelection(population);
    for $parent_1, parent_2 \in parents$ do
        $(offspring_1, offspring_2) = Crossover(parent_1, parent_2)$;
        $offspring_1 = Mutate(offspring_1)$;
        $offspring_2 = Mutate(offspring_2)$;
        $offspring_1.Expansion = GenerateFixedPolarity($
          $zeroPolarityExpansion,$
          $offspring_1.Chromosome)$;
        $offspring_1.Fitness = Evaluate(offspring_1.Expansion)$;
        $offspring_2.Expansion = GenerateFixedPolarity($
          $zeroPolarityExpansion,$
          $offspring_2.Chromosome)$;
        $offspring_2.Fitness = Evaluate(offspring_2.Expansion)$;
        $offspring = \{offspring\} \cup offspring_1 \cup offspring_2$;
    bestSolution = GetBestSolution(offspring);
    for $(i = 0; i < ElitismCount; i++)$ do
        $offspring = \{offspring\} \cup bestSolution$;
    population = replace(population, offspring);
$Return(bestSolution)$;
```

Algorithm 3 GAMP.

```
population = InitializePopulation(populationSize, function);
zeroPolarityExpansion=GenerateMixedPolarity(function,encodedZeroPolarity);
chromosomeLength=
        GetNumberOfInputs(function)*GetNumberOfProductTerms(zeroPolarityExpansion);
encodedZeroPolarity=" ";
for $(i = 0; i < chromosomeLength; i++)$ do
    encodedZeroPolarity+="0";
for $(i = 0; i < populationSize; i++)$ do
    population[i].Expansion=GenerateMixedPolarity(
        zeroPolarityExpansion,
        population[i].Chromosome);
    population[i].Fitness=Evaluate(population[i].Expansion);
for $(i = 0; i < NumberOfIterations; i++)$ do
    offsprings=$\phi$
    parents = RouletteWheelSelection(population);
    for $parent_1, parent_2 \in parents$ do
        $(offspring_1, offspring_2) = Crossover(parent_1, parent_2)$;
        $offspring_1 = Mutate(offspring_1)$;
        $offspring_2 = Mutate(offspring_2)$;
        $offspring_1.Expansion = GenerateMixedPolarity($
         $zeroPolarityExpansion,$
         $offspring_1.Chromosome)$;
        $offspring_1.Fitness = Evaluate(offspring_1.Expansion)$;
        $offspring_2.Expansion = GenerateMixedPolarity($
         $zeroPolarityExpansion,$
         $offspring_2.Chromosome)$;
        $offspring_2.Fitness = Evaluate(offspring_2.Expansion)$;
        $offspring = \{offspring\} \cup offspring_1 \cup offspring_2$;
    bestSolution = GetBestSolution(offspring);
    for $(i = 0; i < ElitismCount; i++)$ do
        $offspring = \{offspring\} \cup bestSolution$;
    population = replace(population, offspring);
$Return(bestSolution)$;
```