

---

**Algorithm 1** Combined Genetic Algorithm

---

```
procedure cGA(vessels, nGA, RunTime, nGen, nImpr)
  population  $\leftarrow$  INITGA(nGA, vessels)
  popID  $\leftarrow$  1
  while SessionTime  $\leq$  RunTime  $\wedge$  popID  $\leq$  nGen do
    tempPop1  $\leftarrow$  MUTATE(population)
    CALCULATECOST(tempPop1)
    UPDATE(population, tempPop1)
    statInd  $\leftarrow$  CREATESTATISTICALINDIVIDUAL()
    UPDATE(population, statInd)
    tempPop2  $\leftarrow$  IMPROVE(population, nImpr)
    CALCULATECOST(tempPop2)
    UPDATE(population, tempPop2)
    population  $\leftarrow$  SELECTION(population)
    popID  $\leftarrow$  popID + 1
  end while
end procedure
```

---

---

**Algorithm 2** Generating initial solutions for cGA

---

```
procedure INITGA(nGA, vessels)
  i  $\leftarrow$  1
  while i  $\leq$  nGA do
     $\Psi(i)$   $\leftarrow$  start $\Psi$ 
    UnusedVessels  $\leftarrow$  vessels
    individual(i)  $\leftarrow$  l empty lists
    while UnusedVessels  $\neq$   $\emptyset$  do
      v  $\leftarrow$  ROULETTE(UnusedVessels, selectionCriteria)
      p  $\leftarrow$  RANDOMSUBLIST(1, l)
      individual(i, p)  $\leftarrow$  APPEND(individual(i, p), v)
      UnusedVessels  $\leftarrow$  UnusedVessels  $\setminus$  {v}
    end while
    if FEASIBLE(individual(i)) then
      DECODE(individual(i),  $\Psi(i)$ )
      i  $\leftarrow$  i + 1
    end if
  end while
end procedure
```

---

---

**Algorithm 3** Mutation phase for cGA

---

```
procedure MUTATE(population)
  newPop  $\leftarrow$  population
  if iterNo  $\leq$   $\frac{1}{3}nGen$  then
    noGenes  $\leftarrow$  l
  else
    if iter  $\leq$   $\frac{2}{3}nGen$  then
      noGenes  $\leftarrow$  RANDOMINTEGER(1, l)
    else
      noGenes  $\leftarrow$  1
    end if
  end if
  for ind  $\in$  population do
     $\mu GA \leftarrow 0.9 \cdot popID/nGen$ 
    newIndividual1  $\leftarrow$  MOVESUBLIST(ind, noGenes,  $\mu GA$ )
    newIndividual2  $\leftarrow$  CHANGESUBLIST(ind, noGenes,  $1 - \mu GA$ )
    UPDATE(newPop, newIndividual1, newIndividual2)
  end for
end procedure
```

---

---

**Algorithm 4** Generating statistical individual for cGA

---

```
procedure CREATESTATISTICALINDIVIDUAL()
  statInd  $\leftarrow$  l empty lists
  solved  $\leftarrow$  {}
  notSolved  $\leftarrow$  {}
  for v  $\in$  vessels do
    freq  $\leftarrow$  MOSTFREQUENTPOSITION(v)
    if freq  $<$  0.85 then
      notSolved  $\leftarrow$  APPEND(notSolved, v)
    else
      solved  $\leftarrow$  APPEND(solved, v)
    end if
  end for
  statInd(1)  $\leftarrow$  solved
  for v  $\in$  notSolved do
    p  $\leftarrow$  RANDOMSUBLIST(2, l)
    statInd(p)  $\leftarrow$  APPEND(statInd(p), v)
  end for
end procedure
```

---

---

**Algorithm 5** Optimization phase for cGA

---

```
procedure IMPROVE(population, nImpr)
  for i  $\leftarrow$  1, nImpr do
    ind  $\leftarrow$  TOURNAMENTFORIMPROVEMENT(population, size)
    vessels  $\leftarrow$  SORT(vessels, ind)
    for v  $\in$  vessels do
      pos  $\leftarrow$  FINDCHEAPERPOSITIONS(v,  $\xi$ )
      for p  $\in$  pos do
        V  $\leftarrow$  CONFLICTVESSELS(v, p)
        movingVessels  $\leftarrow$   $V \cup \{v\}$ 
         $\xi(movingVessels) \leftarrow start\Psi(movingVessels)$ 
        tempAllocation(movingVessels)  $\leftarrow$  SOLVE(movingVessels,  $\Psi$ )
      end for
    end for
    vesselGroups  $\leftarrow$  MAXSAVINGS(tempAllocation)
    ind  $\leftarrow$  UPDATE(ind, vesselGroups)
    ind  $\leftarrow$  BERTHLOCALSEARCH(ind)
  end for
end procedure
```

---

---

**Algorithm 6** Selection operator for cGA

---

```
procedure SELECTION(population)  
  newPop  $\leftarrow$  {}  
  noElite  $\leftarrow$  ROUND( $\frac{1}{3}nGA$ )  
  newPop  $\leftarrow$  ELITE(population, noElite)  
  noSize1  $\leftarrow$  ROUND( $0.4 \cdot \frac{2}{3}nGA$ )  
  t  $\leftarrow$  TOURNAMENTSELECTION(size1, noSize1)  
  newPop  $\leftarrow$  APPEND(newPop, t)  
  noSize2  $\leftarrow$  ROUND( $nGA - noElite - noSize1$ )  
  t  $\leftarrow$  TOURNAMENTSELECTION(size2, noSize2)  
  newPop  $\leftarrow$  APPEND(newPop, t)  
end procedure
```

---