A Tunable Generator of Instances of Permutation-based Combinatorial Optimization Problems: Algorithms to choose the Local Optima according the three configurations

Leticia Hernando, Alexander Mendiburu and
Jose A. Lozano

October 8, 2014

1 Introduction

In this document we specify the three algorithms used to choose the local optima for the experiments shown in Section VII-B of the paper:

L. Hernando, A. Mendiburu and J.A. Lozano, A Tunable Generator of Instances of Permutation-based Combinatorial Optimization Problems, IEEE Transactions on Evolutionary Computation, 2014

2 Configurations

1st CONFIGURATION

Algorithm 1 Algorithm to choose the global optimum surrounded by all the local optima, as close as possible.

```
Choose at random $\sigma_1 \in \Omega$
$k = 2$
$dist = 2$
while $k \leq m$ do
   $t = 0$
   while $t < 50$ do
      Choose at random $\sigma_k \in \Omega$ such that $d(\sigma_k, \sigma_1) = dist$
      if $\exists i < k$ such that $\sigma_k = \sigma_i$ then
         $t = t + 1$
      else
         $t = 0$
         $k = k + 1$
      end if
   end while
   $dist = dist + 2$
end while
```

2nd CONFIGURATION

Algorithm 2 Algorithm to choose all the local optima close except the global optimum that is as far of them as possible.

```
Choose at random $\sigma_1 \in \Omega$
$k = 2$
$dist = 2$
while $k \leq m - 1$ do
   $t = 0$
   while $t < 50$ do
      Choose at random $\sigma_k \in \Omega$ such that $d(\sigma_k, \sigma_1) = dist$
      if $\exists i < k$ such that $\sigma_k = \sigma_i$ then
         $t = t + 1$
      else
         $t = 0$
         $k = k + 1$
      end if
   end while
   $dist = dist + 2$
end while
Choose $\sigma_m \in \Omega$ such that $d(\sigma_m, \sigma_1) = maximum\_distance$
```

3^{rd} CONFIGURATION

Algorithm 3 Algorithm to choose all the local optima, including the global optimum, uniformly spread along the search space.

```
Choose at random $\sigma_1 \in \Omega$
$k = 2$
$dist = 2$
while $k \leq m$ do
    Choose $\sigma_k \neq \sigma_i, \forall i < k$, such that $d(\sigma_k, \sigma_1) = dist$
    if $dist + 2 \leq maximum\_distance$ then
        $dist = dist + 2$
    else
        $dist = 2$
    end if
    $k = k + 1$
end while
```