An evaluation of methods for estimating the number of local optima in combinatorial optimization problems

Leticia Hernando[1], Alexander Mendiburu[2], and Jose A. Lozano[1]

[1]Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, 20018 San Sebastián, Spain
[2]Intelligent Systems Group, Department of Computer Architecture and Technology, University of the Basque Country UPV/EHU, 20018 San Sebastián, Spain

Abstract

Nowadays, the solution of many combinatorial optimization problems is carried out by metaheuristics, which generally, make use of local search algorithms. These algorithms use some kind of neighborhood structure over the search space. The performance of the algorithms strongly depends on the properties that the neighborhood imposes on the search space. One of these properties is the number of local optima. Given an instance of a combinatorial optimization problem and a neighborhood, the estimation of the number of local optima can help, not only to measure the complexity of the instance, but also to choose the most convenient neighborhood to solve it. In this paper we review and evaluate several methods to estimate the number of local optima in combinatorial optimization problems. The methods reviewed not only come from the combinatorial optimization literature, but also from the statistical literature. A thorough evaluation in synthetic as well as real problems is given. We conclude by providing recommendations of methods for several scenarios.

Keywords—
Combinatorial Optimization Problems, Local Search, Estimation, Number of local optima, Species estimation problem, Traveling Salesman Problem, Flow Shop Scheduling Problem.

1 Introduction

Metaheuristic algorithms have been proved as efficient methods for solving hard combinatorial optimization problems. Most of these methods are based on, or use a kind of local search that relies on a neighborhood structure over the search space. The properties of this neighborhood can cause dramatical differences in the performance of the local search methods (Mattfeld and Bierwirth, 1999; Reeves and Eremeev, 2004; Kirkpatrick and Toulouse, 1985; Hertz et al., 1994; Fonlupt et al., 1999; Tomassini et al., 2008). One of the characteristics that the neighborhood imposes on the search space is the number of local optima. This property has attracted much attention because it seems to be related to the difficulty of finding the global optima (Eremeev and Reeves, 2002; Albrecht et al., 2008, 2010; Reeves and Aupetit-Bélaidouni, 2004; Reeves, 2001; Grundel et al., 2007; Garnier and Kallel, 2001; Caruana and Mullin, 1999; Eremeev and Reeves, 2003). Therefore, the number of local optima has been considered as an indirect complexity measure of an instance when solving it with a local search algorithm. Moreover, as the number of local optima depends on the neighborhood, it would be useful to take it into account in order to choose the most suitable neighborhood to solve a particular problem instance efficiently.

Unfortunately, in practice, given an instance of a combinatorial optimization problem and a neighborhood, we do not know in advance the number of local optima. Thus, the development of methods that efficiently estimate the number of local optima seems to be a requirement in order to design algorithms that work in the right neighborhood. This work is connected to the area of evolutionary computation as the analysis of the set of local optima in a landscape associated to the problem can be related to the investigation of some properties of evolutionary algorithms, such as properties of the stable steady-state points in genetic algorithms (Vose and Wright, 1995; Reeves, 2002).

While several approaches have been proposed to estimate and bound the expected number of local optima for combinatorial optimization problems (Grundel et al., 2007; Albrecht et al., 2008, 2010), the literature is not so extensive when it is about estimating the number of local optima of a particular instance. Despite this, it is possible to find works on this topic. A key aspect to take into account when designing a method to estimate the number of local optima of an instance is the distribution of the sizes of the attraction basins (informally, an attraction basin of a local optimum is the set of initial solutions that, after applying a local search algorithm to them, end at that local optimum). Most of these approaches assume that the attraction basins are equally-sized (Caruana and Mullin, 1999; Eremeev and Reeves, 2003), and they propose methods to obtain lower bounds for the number of local optima. Under this assumption on the attraction basins, there are works where biased estimators are obtained, and they try to correct this bias to provide an unbiased estimator (Eremeev and Reeves, 2002; Reeves, 2001; Reeves and Aupetit-Bélaidouni, 2004). On the other hand, there are papers where the sizes of the attraction basins are assumed to fit a certain type of parametric distribution, such as gamma or lognormal. For example, in Garnier and Kallel (2001), the authors assume a gamma distribution for the relative sizes of the attraction basins.

The problem of estimating the number of local optima can be compared with a well-known problem in biology: Estimating the number of different species in a population. In the Statistics field, we can find plenty of algorithms and methods used to estimate the number of species in a population. In fact, Eremeev and Reeves (2003) noticed the connection between these two problems, and they applied the Schnabel-Census method, already used for estimating species, for estimating the number of local optima. In Bunge and Fitzpatrick (1993), as well as in Seber (1986, 1992); Schwarz and Seber (1999), an exhaustive classification of methods for estimating the number of species is given. In those works the literature is organized depending on the sampling model, population specification, and philosophy of the estimation. We can also find recent papers (Chao and Bunge, 2002; Wang, 2010) that give estimators for the number of classes by assuming that the species abundance follows a Poisson-Gamma model, and others (Wang and Lindsay, 2005, 2008) in which estimators are based on the conditional likelihood of a Poisson mixture model. Unfortunately, there are extreme difficulties associated with estimating the population size (Link, 2003).

In this work, we present an evaluation of methods for estimating the number of local optima, that not only use the methods proposed in the combinatorial optimization field, but also those developed for the species problem in the statistics arena. After describing them in detail, we test their performance under three different scenarios:

1. Simulated instances of combinatorial optimization problems.

2. Random instances of the Traveling Salesman Problem.

3. Instances of the Traveling Salesman Problem with real distances between cities, and instances of the Flowshop Scheduling Problem obtained from the well-known Taillard's benchmark.

The rest of the paper is organized as follows. The preliminary mathematical background is given in Section 2. In Section 3 we explain in detail the selected estimate methods. Section 4 shows the experimental results when applying the methods to synthetic data as well as to real instances, and discusses the results observed for the different methods, providing clues to help select the most suitable estimation algorithm for a given instance. Finally, the conclusions are presented in Section 5.

2 Preliminaries

A combinatorial optimization problem (COP) consists of finding an optimal solution of (from now on, minimizing) a function

$$\begin{array}{rccc} f: & \Omega & \longrightarrow & \mathbb{R} \\ & x & \longmapsto & f(x) \end{array}$$

where the search space Ω is a finite or countable infinite set.

A common way of solving these problems is by means of metaheuristic algorithms, most of which use a kind of local search that is based on a neighborhood structure. A neighborhood N in a search space Ω is a mapping that assigns a set of neighbor solutions $N(x) \in \mathcal{P}(\Omega)$ to each solution $x \in \Omega$:

$$\begin{array}{rccc} N: & \Omega & \longrightarrow & \mathcal{P}(\Omega) \\ & x & \longmapsto & N(x) \end{array}$$

Based on the definition of neighborhood, we say that a solution $x^* \in \Omega$ is a local optimum if $f(x^*) \leq f(x),\ \forall x \in N(x^*)$ (local minimum) or if $f(x^*) \geq f(x),\ \forall x \in N(x^*)$ (local maximum). Obviously, one solution $x^* \in \Omega$ can be a local optimum under a neighborhood N_1, but not when considering a different neighborhood N_2. If $y^* \in \Omega$ is a solution such that $f(y^*) \leq f(y)$ (or $f(y^*) \geq f(y)$) , $\forall y \in \Omega$, then y^* is a global optimum. Clearly, the global optima are local optima for any neighborhood.

Algorithm 1 Deterministic Best-Improvement Local Search Algorithm

```
Choose an initial solution x ∈ Ω
repeat
  x* = x
  for i = 1 → |N(x*)| do
    Choose y_i ∈ N(x*)
    if f(y_i) < f(x) then
      x = y_i
    end if
  end for
until x = x*
```

A concept that plays an important role in this paper is that of basin of attraction $\mathcal{B}(x^*)$ of a local optimum x^*. The basin of attraction of a local optimum depends on the neighborhood, however, we will omit the neighborhood from the notation in order to simplify it. Roughly speaking, an attraction basin $\mathcal{B}(x^*)$ is composed of all the solutions that, after applying a local search algorithm starting with those solutions, finishes in x^*. In our case, we use a deterministic best-improvement local search (see Algorithm 1). It is important to notice that the neighbors are evaluated in a specific order, so that, in the case of two neighbors having the same function value, the algorithm will always choose that which was encountered first. We denote by $\mathcal{H}$ the operator that associates, to each solution x, the local optimum obtained after applying the algorithm. So, the attraction basin $\mathcal{B}(x^*)$ of a local optimum x^* is the set that can be defined in the following way:

$$\mathcal{B}(x^*) = \{x \in \Omega \mid \mathcal{H}(x) = x^*\}\,.$$

The relative size of the attraction basin $\mathcal{B}(x^*)$ related to the search space Ω, i.e., the proportion of solutions of the whole search space that belong to the basin $\mathcal{B}(x^*)$, denoted as $p = \frac{|\mathcal{B}(x^*)|}{|\Omega|}$, is of special relevance for the estimation of the number of local optima. Given the deterministic nature of $\mathcal{H}$, an important property of this concept is that the attraction basins of the local optima define a partition of Ω.

The problem we are considering in this paper, that is, estimating the number of local optima, can be considered as a classical estimation problem in statistics. Therefore, the most common way of solving it is by constructing an estimator by collecting a set of statistics from a sample. Due to the fact that we want to estimate the number of local optima, our sample has to be related to the local optima. Basically, most of the presented methods start by taking a uniformly distributed random sample $S = \{x_1, x_2, ..., x_M\} \subseteq \Omega$ of size M. The local search algorithm is applied to each solution in S, and from the M local optima obtained we get the r ($\leq M$) different ones into the set $S^* = \{x_1^*, ..., x_r^*\}$.

Two important statistics that will be used for the estimation methods are α_i and β_i. We denote by α_i ($i \in \{1, 2, ..., r\}$) the number of initial solutions of the sample that belong to the attraction basin of the local optimum x_i^*:

$$\alpha_i = \left| \{x \in S \mid \mathcal{H}(x) = x_i^*\} \right| = \left| \{x \in S \mid x \in \mathcal{B}(x_i^*)\} \right| \leq \left|\mathcal{B}(x_i^*)\right|.$$

With this information, the following statistic is calculated:

$$\beta_j = \left| \{\alpha_i \mid \alpha_i = j, i \in \{1, 2, ..., r\}\} \right| \ , \quad \forall j \geq 1.$$

So, β_j is the number of local optima that have been seen exactly j times in the sample. In the following section, we call β_0 the number of local optima in the search space that have not been found in the sample. Notice that $\beta_j = 0$, $\forall j > M$. Two interesting relations are the following:

$$\sum_{j=1}^{M} \beta_j = r \qquad \text{and} \qquad \textstyle\sum_{j=1}^{M} j * \beta_j = M.$$

3 Estimation methods

In this section we present a review of the methods proposed in the literatire for estimating the number of local optima, as well as the most widespread methods for estimating the number of species in a population. Table 1 shows the methods collected from the combinatorial optimization and also from the statistics area. Inside the combinatorial optimization field we find the following methods: a method based on the Birthday problem (Caruana and Mullin, 1999), First Repetition Time, Maximal First Repetition Time and Schnabel-Census (Eremeev and Reeves, 2003), Jackknife and ?Bootstrap (Eremeev and Reeves, 2002), and a method based on Gamma distributions (Garnier and Kallel, 2001). In the statistics field, the selected methods are: Chao 1984 (Chao, 1984), Chao & Bunge (Chao and Bunge, 2002), Chao & Lee 1 and Chao & Lee 2 (Chao and Lee, 1992), a Poisson-Compound Gamma Model (Wang, 2010) and a Penalized Nonparametric Maximum Likelihood Approach (Wang and Lindsay, 2005).

We discarded some of the methods shown in Table 1, due to their poor performance (Caruana and Mullin, 1999), high dependence with the sample size (Garnier and Kallel, 2001), or the high computation time (Wang and Lindsay, 2005, 2008; Wang, 2010) observed in preliminary experiments. The methods analyzed in this paper are highlighted in bold.

Table 1: A classification of the estimate methods selected on both the combinatorial optimization field and the statistics area, according to the sampling model. The methods analyzed in this paper are in bold and their abbreviations are indicated between brackets.

	SAMPLING MODEL	METHOD (ABBREVIATION)		REFERENCE
COMBINATORIAL OPTIMIZATION	Multinomial	Method based on the Birthday problem		Caruana and Mullin (1999)
		Confidence Intervals	**First Repetition Time (*FRT*)** **Maximal First Repetition Time (*MFRT*)** **Schnabel-Census (*Sch-Cen*)**	Eremeev and Reeves (2003)
		Bias Correction	**Jackknife (*Jckk*)** **Bootstrap (*Boots*)**	Eremeev and Reeves (2002)
	Gamma	Method based on a Gamma model		Garnier and Kallel (2001)
STATISTICS	Multinomial	**Chao 1984 (*Chao1984*)**		Chao (1984)
		Chao & Lee 1 (*ChaoLee1*) **Chao & Lee 2 (*ChaoLee2*)**		Chao and Lee (1992)
	Poisson - Gamma	**Chao & Bunge (*ChaoBunge*)**		Chao and Bunge (2002)
		Poisson-Compound Gamma Model		Wang (2010)
	Mixed Poisson	Penalized Nonparametric Maximum Likelihood Approach		Wang and Lindsay (2005)

The five methods proposed in the field of optimization are explained in detail. First Repetition Time (*FRT*), Maximal First Repetition Time (*MFRT*) and Schnabel-Census are methods that provide lower bounds and that can be used for computing confidence intervals, while Jackknife and Bootstrap are bias correcting non-parametric methods. *FRT* has attracted our interest because it is a parameter-less method, whereas *MFRT* and *Sch-Cen* only depend on one parameter which is the sample size. So, these three methods do not require too much computation time. Jackknife is a method that also depends on the sample size and it is very fast. Bootstrap not only needs the sample size, but also another parameter: the number of repetitions inside the method. The fact of carrying out repetitions causes the method to take more time than the other methods in providing the estimated value.

In a second step, we present methods proposed in the field of statistics used by biologists and ecologists when determining how many different classes of species are in a population of plants or animals. They are non-parametric methods, but they are based on particular sampling models. *Chao1984*, *ChaoLee1* and *ChaoLee2* are based on multinomial sampling, while *ChaoBunge* is based on a mixed Poisson sampling model. The main reason for choosing these methods is that, according to preliminary experiments, they do not require too much computation time and, in general, they give very good estimates.

3.1 Methods proposed in the field of optimization

3.1.1 Methods used for computing confidence intervals

In this section we describe the methods proposed in Reeves and Eremeev (2004): First Repetition Time, Maximal First Repetition Time and Schnabel-Census Procedure. These three methods assume that all the attraction basins of the local optima are equal in size. Under this assumption, and supposing a finite multinomial model, they give (with a high probability) lower bounds for the number of local optima. First, we explain the common aspects of these three methods, and then, the particular details of each of them are given.

Let us start by considering that the continuous distribution function $F_v(t)$ of a random variable T that depends on a parameter v is known, and this distribution function is a strictly monotonically decreasing function on this parameter v, that is:

$$\text{if} \quad v_2 > v_1 \quad \text{then} \quad F_{v_2}(t) < F_{v_1}(t)\ , \quad \forall t.$$

Now, the (1 - ϵ_1 - ϵ_2)*100% confidence interval for the parameter v is calculated. So, the goal is to find [v_1 , v_2], with $v_1 < v_2$, and such that P($v_1 \leq v \leq v_2$) = 1 - ϵ_1 - ϵ_2, from the known distribution function $F_\mu(t) = P(T \leq t \mid v = \mu)$. Taking into account that $\tau \in \mathbb{N}$ is an observed value sampled from this distribution, we obtain:

$$v_1 = min\left\{\mu \mid 1 - F_\mu(\tau - 1) \geq \epsilon_1\right\} \quad , \quad v_2 = max\left\{\mu \mid F_\mu(\tau) \geq \epsilon_2\right\}.$$

Following the framework of Reeves and Eremeev (2004), we work with the fixed value $\epsilon_1 = 0.05$. So, these methods will define v_1 as a lower bound with probability 0.95 when $\epsilon_2 = 0$, and consequently v_2 is infinity.

Particularly, in *FRT*, *MFRT*, and Schanbel-Census methods, we will denote v as the number of local optima. Let $\mathbf{p} = (p_1, p_2, ..., p_v)$ be the vector of probabilities of finding the corresponding local optima, that is, the relative sizes of the attraction basins of the local optima. If $\mathbf{p} = \bar{\mathbf{p}} = (\frac{1}{v}, \frac{1}{v}, ..., \frac{1}{v})$ then all the local optima have the same probability of being found. These methods calculate the distribution function $F_{\bar{\mathbf{p}}}(t)$ according to a random variable T, which in each case will determine different concepts.

1. **The First Repetition Time method**

   This method starts taking uniformly at random a solution x_1 from the search space Ω, and a local search algorithm (in our case, algorithm $\mathcal{H}$)

is applied to x_1, ending at a local optimum x_1^*. This process is repeated until a local optimum is seen twice.

The random variable T denotes, in this case, the number of initial solutions x_i taken until a local optimum is repeated. The distribution function of T corresponding to the vector $\mathbf{p}$ is $F_{\mathbf{p}}(t)$. It can be proved (Reeves and Eremeev, 2004) that for any $t \geq 2$, $F_{\mathbf{p}}(t)$ is minimal only at $\mathbf{p} = \bar{\mathbf{p}} = (\frac{1}{v}, \frac{1}{v}, ..., \frac{1}{v})$, where v is the number of local optima.

Now, the distribution function $F_{\bar{\mathbf{p}}}(t)$ is calculated for the variable T.

$$F_{\bar{\mathbf{p}}}(t) = P(T \leq t \mid \mathbf{p} = \bar{\mathbf{p}}) = 1 - P(T > t \mid \mathbf{p} = \bar{\mathbf{p}}),$$

where $P(T > t \mid \mathbf{p} = \bar{\mathbf{p}})$ is the probability of finding none of them repeated in the t first optima: $P(T > t \mid \mathbf{p} = \bar{\mathbf{p}}) = \frac{v}{v} \cdot \frac{v-1}{v} \dots \frac{v-t+1}{v} = \left(\frac{1}{v}\right)^t \binom{v}{t} t!$.

So,

$$F_{\bar{\mathbf{p}}}(t) = 1 - \left(\frac{1}{v}\right)^t \binom{v}{t} t!.$$

Assuming that τ is the value obtained from the sample for the variable T, the estimate for the number of local optima v_1 is given by the following formula:

$$\hat{v}_{FRT} = v_1 = min \left\{ \mu \;\Big|\; \left(\frac{1}{\mu}\right)^{\tau-1} \binom{\mu}{\tau - 1} (\tau - 1)! \geq 0.05 \right\}.$$

2. **The Maximal First Repetition Time method**

   In this case a uniformly distributed random sample S of size M is taken from the search space: $S = \{x_1, x_2, ..., x_M\} \subseteq \Omega$. Then, a local search is applied to each solution of the sample, so that M local optima $\{x_1^*, x_2^*, ..., x_M^*\}$ are obtained. Notice that not all of them have to be different. Then, starting from x_1^* and taking the local optima in order of appearance, subsequences $S_i \subseteq S$ are created, where each of them ends with its first reoccurrence of a local optimum. It is as if we were repeating the First Repetition Time procedure many times. If there is no local optima repeated, then the unique subsequence obtained is $S_1 = S$ of size M.

   The number of subsequences obtained is denoted by s. The variable that denotes the length of the j-th subsequence is T_j, and $T^{(s)} = max_j T_j$ represents the maximum length of all the subsequences.

   The distribution function of the variable $T^{(s)}$ corresponding to the vector of probabilities of the local optima $\mathbf{p}$ is

$$F_{\mathbf{p}}^{(s)}(t) = P(T^{(s)} \leq t \mid \mathbf{p}) = P(T_j \leq t, j = 1, ..., s \mid \mathbf{p}).$$

   As in the previous case, for a fixed value of t and a fixed value of s, it can be proved (Reeves and Eremeev, 2004) that $F_{\mathbf{p}}^{(s)}$ is minimal when $\mathbf{p} = \bar{\mathbf{p}} = (\frac{1}{v}, \frac{1}{v}, ..., \frac{1}{v})$.

The distribution function $F_{\bar{\mathbf{p}}}^{(s)}(t)$ for the variable $T^{(s)}$ is:

$$F_{\bar{\mathbf{p}}}^{(s)}(t) = \prod_{i=1}^{s} P(T_i \le t \mid \mathbf{p} = \bar{\mathbf{p}}) = \left[1 - \left(\frac{1}{v}\right)^t \binom{v}{t} t!\right]^s .$$

If τ is the value obtained from the sample for the variable $T^{(s)}$, then the estimate for the number of local optima v_1 is given by the following formula:

$$\hat{v}_{MFRT} = v_1 = min\left\{\mu \;\middle|\; 1 - \left[1 - \left(\frac{1}{\mu}\right)^{\tau-1} \binom{\mu}{\tau-1}(\tau-1)!\right]^s \ge 0.05\right\}.$$

3. **Schnabel Census Procedure**

This method has also been used in ecology to provide an estimate of the size of a population of animals. It consists of taking a sample of size M and counting the number of distinct animals seen. However, we include it in this section because there are already works (Eremeev and Reeves, 2002; Reeves and Eremeev, 2004) that have used this method to estimate the number of local optima in COP. The way to proceed in the case of estimating the number of local optima is similar to the problem of estimating the number of animals.

Firstly, a uniformly distributed random sample $S = \{x_1, x_2, ..., x_M\} \subseteq \Omega$ of size M is taken from the search space Ω, and a local search algorithm is applied to each solution in S, obtaining r different local optima $\{x_1^*, x_2^*, ..., x_r^*\}$, with $r \le M$.

Let R be the random variable that represents the number of different local optima found. The distribution function for the variable R, when a sample of size M has been taken from Ω and when it corresponds to the vector of probabilities of the local optima $\mathbf{p}$ is:

$$F_{\mathbf{p}}(r, v, M) = P(R \le r \mid M, v, \mathbf{p}) = \sum_{i=1}^{r} P(R = i \mid M, v, \mathbf{p}).$$

If $\mathbf{p} = \bar{\mathbf{p}}$, then $P(R = i \mid M, v) = \frac{v!S(M,i)}{(v-i)!v^M}$, where $S(M, i)$ is the Stirling number of the second kind, that is, the number of all possible partitions of an M-element set into i non-empty subsets.

Then,

$$F_{\bar{\mathbf{p}}}(r, v, M) = \sum_{i=1}^{r} \frac{v!S(M,i)}{(v-i)!v^M}.$$

The estimate for the number of local optima v_1 is given by the following formula:

$$\hat{v}_{Sch-Cen} = v_1 = min\left\{\mu \;\middle|\; 1 - \sum_{i=1}^{r-1} \frac{\mu!S(M,i)}{(\mu-i)!\mu^M} \ge 0.05\right\}.$$

3.1.2 Bias-correcting nonparametric methods

In this section we describe the application of two commonly used bias-correcting methods to the problem of estimating the number of local optima. These methods are Jackknife and Bootstrap, and their specific use in this context was proposed in Reeves (2001); Eremeev and Reeves (2002, 2003); Reeves and Aupetit-Bélaidouni (2004). While in the original papers only the mechanic of the algorithm is provided, we have also added the assumptions of the methods, as we consider them relevant for our work.

Jackknife and Bootstrap are nonparametric methods based on ideas of resampling. Moreover, they use the concept of bias of an estimator (the difference between the estimated value and the real value) to improve an initial estimate. There is an important difference that the authors make between the application of Jackknife and the application of Bootstrap. In the Jackknife method, no assumptions about the sampling model are made when calculating the initial biased estimate. However, the Bootstrap method has an underlying assumption about the sampling model because it uses the maximum likelihood estimator as an initial biased estimator.

1. **Jackknife**

   The Jackknife method starts from a biased estimator $\hat{v}$, and assumes that the bias decreases asymptotically as the size of the sample increases. The underlying resampling technique consists of leaving the different points x_i of the initial sample out, and finding estimators that reduce the bias of $\hat{v}$. The mean of these estimates is considered the Jackknife estimator.

   This method, in the same way as Schnabel Census Procedure, was previously proposed for the estimation of population sizes (Burnham and Overton, 1978). In the context of estimating the number of local optima (Eremeev and Reeves, 2002), a uniformly distributed random sample $S = \{x_1, ..., x_M\}$ of size M is taken from the search space. After applying a local search algorithm to each solution $x_i \in S$, the set of local optima $\mathcal{L}^* = \{x_1^*, x_2^*, ..., x_M^*\}$ is obtained, with $r \leq M$ different local optima.

   Next, one point x_i is left out from the sample S. The subset $\mathcal{L}_i^* \subseteq \mathcal{L}^*$ that contains the local optima that correspond to all the solutions in $S - \{x_i\}$ is considered: $\mathcal{L}_i^* = \mathcal{L}^* - \mathcal{H}(x_i)$. If this idea is repeated leaving each of the points out from the original sample once each time, we obtain M subsets $\mathcal{L}_1^*, \mathcal{L}_2^*, ..., \mathcal{L}_M^* \subseteq \mathcal{L}^*$, with $r_{-1}, r_{-2}, ..., r_{-M} \leq r$ different local optima.

   The biased estimator $\hat{v} = r$ of v is assumed to be of the form $r = v + \frac{a_1}{M} + \frac{a_2}{M^2} + \frac{a_3}{M^3} + ...$ Thus, the bias is of order $\frac{1}{M}$. The purpose is to find an estimator that reduces the bias to $O(\frac{1}{M^2})$. So, for each $i \in \{1, 2, ..., M\}$, the following estimator is defined:

   $$r_i = Mr - (M-1)r_{-i} \,, \tag{1}$$

   so that $r_i = v + O\left(\frac{1}{M^2}\right)$.

Since

$$r_{-i} = r - j_i \ \ , \quad \text{where } j_i = \begin{cases} 0 & if \quad \exists\, x_k \neq x_i \in S \ s.t.\ \mathcal{H}(x_k) = x_i^* \\ 1 & otherwise \end{cases} \tag{2}$$

then, from (1) and (2) the estimator is:

$$r_i = r + (M-1)j_i \ \ , \quad \forall i \in \{1, 2, ..., M\}.$$

The Jackknife estimator for the number of local optima is the mean value of r_i:

$$\hat{v}_{Jckk} \ = \ \frac{1}{M}\sum_{i=1}^{M} r_i = r + \frac{M-1}{M}\beta_1,$$

where β_1 is the number of local optima which have only been seen once.

Notice that when the sample size tends to infinity, β_1 tends to 0, and therefore, the estimate for the number of local optima tends to the real number of local optima. So, this is an unbiased estimator.

2. **Bootstrap**

   The Bootstrap method starts from a biased estimator $\hat{v}$, obtained from a sample S. Resamples from S are taken and the biased estimate is calculated in each case. With this information, an estimator of the bias is provided. So, the result of adding the estimated bias to the initial $\hat{v}$ is the Bootstrap estimator.

   In the application of this method to the estimation of the number of local optima (Eremeev and Reeves, 2002), we start from the set $S^* = \{x_1^*, x_2^*, ..., x_r^*\}$ of r different local optima. Assuming a multinomial model (Reeves and Aupetit-Bélaidouni, 2004), with equally sized attraction basins, the probability distribution of the random variable R that represents the number of different local optima found in the sample S is given by

   $$P(R=r) = \tfrac{v!}{(v-r)!}\tfrac{S(M,r)}{v^M}, ? \quad 1 \leq r \leq min\{M, v\} \ ,$$

   where $S(M,r)$ is again the Stirling number of the second kind. From this, the maximum likelihood estimate $\hat{v}_r^{ML}$ of v is obtained by solving the equation:

   $$M * log\left(1 - \frac{1}{v}\right) - log\left(1 - \frac{r}{v}\right) = 0.$$

   If r/M is small (lower than 0.3) the best estimate for the number of local optima (Reeves and Aupetit-Bélaidouni, 2004) is actually r, because it is assumed that with small values it is likely that all local optima have been found. So, in this case it is considered that $\hat{v}_r^{ML} = r$.

Afterwards, a resample with replacement of the same size M from S is taken, obtaining r_1 different local optima, and the maximum likelihood estimate of r_1 is considered: $\hat{v}_{r_1}^{ML}$. The same procedure is repeated b times, that is, b resamples with replacement from S are taken, obtaining $\{r_1, r_2, ..., r_b\}$ different local optima, and the maximum likelihood estimate for each r_i is calculated: $\{\hat{v}_{r_1}^{ML}, \hat{v}_{r_2}^{ML}, ..., \hat{v}_{r_b}^{ML}\}$.

With the maximum likelihood estimates for the number of local optima of the resamples $\{\hat{v}_{r_1}^{ML}, \hat{v}_{r_2}^{ML}, ..., \hat{v}_{r_b}^{ML}\}$ and the maximum likelihood estimate for the number of local optima obtained in the original sample $\hat{v}_r^{ML}$, the bias can be estimated and used as a bias correction for $\hat{v}_r^{ML}$. The bias can be calculated as the difference between the maximum likelihood estimate of the number of local optima found with the original sample and the average maximum likelihood estimates of the number of local optima from the resamples: $bias = \hat{v}_r^{ML} - \frac{1}{b}\sum_{i=1}^{b} \hat{v}_{r_i}^{ML}$.

Hence, the Bootstrap estimator for the number of local optima is $\hat{v} = \hat{v}_r^{ML} + bias$, so:

$$\hat{v}_{Boots} = 2\hat{v}_r^{ML} - \frac{1}{b}\sum_{i=1}^{b} \hat{v}_{r_i}^{ML}.$$

A very important observation is that, as the sample size tends to infinity, the number of local optima found tends to be the real number of local optima. In addition, as r/M is very small (M is very large, and r is constant), the maximum likelihood estimate is the number of local optima found from the sample. Moreover, the bias tends to 0, so the estimate tends to be the real number of local optima.

3.2 Methods proposed in the field of statistics

In this section we present four nonparametric methods based on sampling models: *Chao1984* (Chao, 1984), *ChaoLee1*, *ChaoLee2* (Chao and Lee, 1992) and *ChaoBunge* (Chao and Bunge, 2002). Although they were proposed to estimate the number of species in a population, we explain here their specific application to our problem. An important consideration is that they assume an infinite population, while the common COP have a finite search space. However, we treat the search spaces as if they were infinite because of their large cardinality.

All of the methods presented below start from a sample $S = \{x_1, x_2, ..., x_M\} \subseteq \Omega$ of size M. A local search algorithm is applied to each solution $x_i \in S$ and r different local optima $\{x_1^*, x_2^*, ..., x_r^*\}$ are obtained.

1. **Chao 1984**

   This is a nonparametric method proposed in Chao (1984) based on multinomial sampling that has been used to estimate the number of classes in an infinite population.

The estimator given by this method is the result of adding to the number of local optima obtained from the sample a quantity that depends only on the number of local optima seen once and twice in the sample.

This method is based on the estimate of the expected value of the number of unobserved local optima E_{β_0}. Harris (1959) proved that if $j^2 = O(M)$, then $E_{\beta_j} \sim \sum_{i=1}^{v} (Mp_i)^j \frac{e^{-Mp_i}}{j!}$, where p_i is the relative size of the attraction basin of the local optimum x_i^*. The method considers the following distribution function:

$$F(x) = \frac{\sum_{Mp_i \leq x} (Mp_i) e^{-Mp_i}}{\sum_{i=1}^{v} (Mp_i) e^{-Mp_i}}$$

and it assumes that the number of unobserved local optima is of the following form: $E_{\beta_0} \sim \sum_{i=1}^{v} e^{-Mp_i} \sim (E_{\beta_1}) \int_0^M x^{-1}\, dF(x)$.

We want to obtain an estimator $\hat{F}(x)$ of $F(x)$ and thus, find an estimator $\hat{v}$ of v that is the sum of r and the estimated number of unobserved local optima. That is,

$$\hat{v} = r + \beta_1 \int_0^M x^{-1}\, d\hat{F}(x).$$

The moment estimates were proposed in Chao (1984) to obtain an estimator $\hat{F}(x)$ of $F(x)$, and once attained, a lower bound for v was found. As the sample size M tends to infinity, the lower bound tends to the *Chao1984* estimator:

$$\hat{v}_{chao1984} = r + \frac{\beta_1^2}{2\beta_2}.$$

It is very important to take into account that this estimator works when the information is concentrated on the low order occupancy numbers, that is, when β_1 and β_2 carry most of the information. If $\beta_2 = 0$, that is, if there is no local optima seen exactly twice from the sample, the method does not work. Moreover, β_1 is also very important in this estimator, because if $\beta_1 = 0$, the estimate is just the number of local optima obtained from the sample. We find these situations, for example, when M is much higher than the real number of local optima. In this case, it is unlikely that the local optima will be found only once or twice. Furthermore, we can also find $\beta_1 = 0$ and $\beta_2 = 0$ when the attraction basins are close in size, because the different local optima are probably found the same number of times. Notice that only if we force the method to return r as the estimate for the number of local optima when $\beta_1 = 0$ and $\beta_2 = 0$, we can ensure that when M tends to infinity we obtain the real number of local optima.

2. **Chao & Lee**

Chao and Lee (1992) proposed two new estimators based on the estimators proposed by Esty (1985). The methods are also nonparametric, used for infinite population, and based on multinomial sampling.

They are the first methods that included the idea of sampling coverage. These estimators are the sum of the number of local optima observed many times, and quantities dependent on the number of local optima found few times in the sample.

Given a sample S, the sample coverage C is defined as the sum of the probabilities of the observed local optima. That is, the sum of the relative sizes of the attraction basins of the local optima found. Obviously, C is a random variable and an estimator $\hat{C}$ of C (Good, 1953) is $\hat{C} = 1 - \beta_1/M$. Notice that if all the local optima have the same probability of being chosen, that is $p_1 = p_2 = ... = p_v = \frac{1}{v}$, then $C = \frac{r}{v}$. So, an initial estimator of v is $\hat{v}_1 = \frac{r}{\hat{C}}$. Based on (Esty, 1985) and using $\hat{v}_1$, Chao and Lee proposed an estimator $\hat{v}$ of v of the following form: $\hat{v} = \hat{v}_1 + \frac{M(1-\hat{C})}{\hat{C}}\gamma^2$, where $\gamma = \frac{1}{\bar{p}}\left[\sum_{i=1}^{v}(p_i - \bar{p})^2/v\right]^{1/2}$ is the coefficient of variation (with $\bar{p} = \frac{1}{v}\sum_{i=1}^{v} p_i$).

Notice that $\gamma^2 = v\sum_{i=1}^{v} p_i^2 - 1$. Therefore, they suggested: $\gamma^2 = \frac{v\sum_{i=1}^{M} i(i-1)\beta_i}{M(M-1)} - 1$.

Chao and Lee distinguished between what they call abundant species and rare species. Transferring these concepts to our problem, we will distinguish between easy-to-find local optima and hard-to-find local optima. We define a local optimum x_k^* as hard-to-find if $\alpha_k \leq \delta$ for some cut-off value δ. So, the easy-to-find local optima are the x_k^* such that $\alpha_k > \delta$. One can select this cut-off value in advance, but there are studies (Chao and Yang, 1993), that are based on empirical experience set $\delta = 10$. We call r_h the number of hard-to-find local optima, and r_e the number of easy-to-find local optima.

So, taking this distinction into account, the estimators are the following:

$$\hat{v}_{ChaoLee1} = r_e + \frac{r_h}{\hat{C}_h} + \frac{\beta_1}{\hat{C}_h}\hat{\gamma_h}_1^2 \quad , \quad \hat{\gamma_h}_1^2 = max\left\{\frac{r_h}{\hat{C}_h}\frac{\sum_{i=1}^{\delta} i(i-1)\beta_i}{M_h(M_h-1)} - 1, 0\right\}$$

$$\hat{v}_{ChaoLee2} = r_e + \frac{r_h}{\hat{C}_h} + \frac{\beta_1}{\hat{C}_h}\hat{\gamma_h}_2^2 \quad , \quad \hat{\gamma_h}_2^2 = max\left\{\hat{\gamma_h}_1^2\left[1 + (1-\hat{C}_h)\frac{\sum_{i=1}^{\delta} i(i-1)\beta_i}{(M_h-1)\hat{C}_h}\right], 0\right\}$$

where $M_h = \sum_{j=1}^{\delta} j\beta_j$ and $\hat{C}_h = 1 - \beta_1/M_h$.

It is important to take into account that if $\hat{C}_h = 0$, the method does not work. This occurs when $\beta_1 \neq 0$ and $\beta_i = 0, \forall i \in \{2, ..., \delta\}$. Neither does the method work when $M_h = 0$ or $M_h = 1$, that is, when $\beta_i = 0, \forall i \in \{1, ..., \delta\}$, or when $\exists i \in \{1, ..., \delta\}$ such that $\beta_i = 1$ and $\beta_j = 0, \forall j \neq i$. In these cases we redefine the estimators as the number of the different local optima that appear in the sample r. Only under this consideration, we ensure having an unbiased estimator and therefore, obtaining the real number of local optima when M tends to infinity.

3. **Chao & Bunge**

A new estimate technique was proposed by Chao and Bunge (2002). This estimator is also nonparametric in form, but it has some optimality properties under a particular parametric model. This method is based on a mixed Poisson sampling model. It is closely related to the estimators in Chao and Lee (1992).

This method bases the estimate of unobserved local optima on $(\beta_1, \beta_2, ..., \beta_\delta)$ for some cut-off value δ, and then they complete the estimate by adding the number of local optima found more that δ times in the sample.

Firstly, they showed that, for the Gamma-Poisson case, the expected proportion of duplicates in the sample, denoted by θ, can be estimated by $\hat{\theta} = 1 - \frac{\beta_1 \sum_{k=1}^{M} k^2\beta_k}{\left(\sum_{k=1}^{M} k\beta_k\right)^2}$. Secondly, and based on this estimator, they proposed the following estimator for the expected number of unseen optima: $\hat{\beta}_0 = (\hat{\theta}^{-1} - 1)\sum_{k=2}^{M} \beta_k - \beta_1$. Thirdly, with this estimator of the unseen optima, the estimator of the number of local optima was defined as $\hat{v}^* = (\hat{\theta}^{-1} - 1)\sum_{k=2}^{M} \beta_k - \beta_1 + r = \sum_{k=2}^{M} \frac{\beta_k}{\hat{\theta}}$.

Finally, based on the data $(\beta_1, \beta_2, ..., \beta_\delta)$ and the estimator $\hat{v}^*$, and taking into account the distinction between hard-to-find local optima (r_h) and easy-to-find local optima (r_e), their final proposed estimator of v is:

$$\hat{v}_{\mathit{ChaoBunge}} = r_e + \sum_{k=2}^{\delta} \frac{\beta_k}{\hat{\theta}} \quad , \quad \hat{\theta} = 1 - \frac{\beta_1 \sum_{k=1}^{\delta} k^2\beta_k}{\left(\sum_{k=1}^{\delta} k\beta_k\right)^2} \tag{3}$$

where $\hat{\theta}$ takes only into account the hard-to-find local optima, and not all the local optima found in the sample.

Notice that, if $\delta > max\{j \mid \beta_j \neq 0\}$, then all the local optima are considered as hard-to-find, and therefore $\hat{v}_{ChaoBunge} = \sum_{k=2}^{M} \frac{\beta_k}{\hat{\theta}}$, $\hat{\theta} = 1 - \frac{\beta_1}{M^2} \sum_{k=1}^{M} k^2 \beta_k$.

It is important to take into account that if we have a sample in which $\beta_1 \neq 0$ and $\beta_i = 0, \forall i \in \{2, ..., \delta\}$, then $\hat{\theta} = 0$ and therefore the method does not work. In these cases, we consider that $\hat{\theta} = 1$ and so, the estimate is just r. Considering this point we have that, as M tends to infinity, we obtain the real number of local optima.

4 EXPERIMENTS

The accuracy of the different estimators presented in the previous section has been tested on three different datasets: simulated instances of COPs, instances of the Traveling Salesman Problem (TSP) taking random distances, and instances of the TSP with real distances between different cities, as well as instances of the Flow Shop Scheduling Problem (FSSP) obtained from the well-known Taillard's benchmark. Using these datasets we can first test the methods over a wide set of instances with different characteristics (the dataset with simulated instances) and then check whether these conclusions can be generalized for artificial and real instances (second and third datasets). In these three scenarios, we work with problems for which we already know the number of local optima, which allows us to evaluate the accuracy of the different estimates. We report a comparison of the different methods, giving recommendations of the methods that provide the best estimates.

4.1 Synthetic data

4.1.1 Experimental design

The aim of this section is to study the performance of the methods when they are applied to instances with different number of local optima and different distributions of the sizes of the attraction basins. Therefore, we are interested in a set of data that includes instances with attraction basins very similar in size, as well as instances with very different sizes of attraction basins. However, it is not easy to obtain many real or random instances with the desired characteristics. On the one hand, looking for the real number of local optima and the sizes of their attraction basins of a given instance of a COP would require high computation time. On the other hand, it is not easy to find in the literature instances with a high number of local optima that are realistic enough. These are the reasons why we decide to simulate instances of COP, instead of working with real ones.

As far as the methods for estimating the number of local optima are concerned, an instance of a COP is determined by the number of local optima and the size of their attraction basins. Therefore, we can summarize an instance as the pair $(v, \mathbf{p})$, where v denotes the number of local optima and $\mathbf{p} = (p_1, p_2, ..., p_v)$, with $0 < p_i < 1$, $\forall i \in \{1, 2, ..., v\}$, $\sum_{i=1}^{v} p_i = 1$, is the vector that gives the relative sizes of the attraction basins of the local optima. So, we create instances just assuming a certain number of local optima and assigning to each local optimum a probability of being chosen (or a certain relative size of its attraction basin).

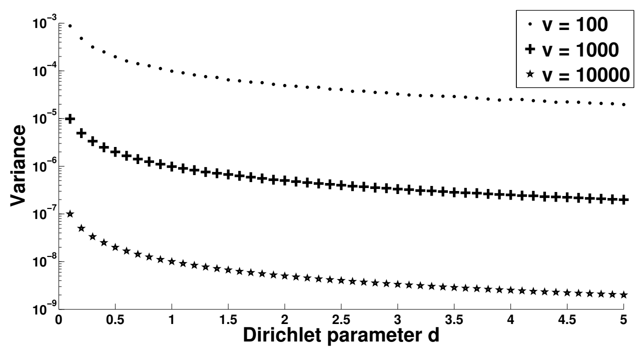


Figure 1: Average variance of the values of the samples obtained from $D(v, d)$, for the different values of d, and for $v = 100, 1000$ and 10000. The Y axis is at logarithmic scale.

One way to do that is by sampling a Dirichlet distribution. When a Dirichlet distribution with v parameters $D(d_1, d_2, ..., d_v)$ is sampled, a vector with v values $(r_1, r_2, ..., r_v)$ is obtained that fulfills the following two relations:

$$0 < r_i < 1 \ , \ \forall i \in \{1, 2, ..., v\} \quad \text{and} \quad \sum_{i=1}^{v} r_i = 1.$$

So, we take advantage of this property of the Dirichlet distribution to simulate the relative sizes of the attraction basins of v local optima. We simplify the sampling by assuming $d_1 = d_2 = ... = d_v = d$, so that we sample Dirichlet distributions with only two parameters: $D(v, d)$.

For the parameter v we decide to work with the following values: 100, 1000 and 10000. Regarding d, working with many values of this parameter would be unfeasible. We decide to make an initial experiment to choose values of d that could simulate very different distributions of the sizes of the attraction basins. In order to choose these values, we sample different $D(v, d)$, for $v = 100, 1000$ and 10000, and $d = 0.1, 0.2, 0.3, ..., 4.8, 4.9, 5$. For each v and d we take 100 samples and calculate the variance of the v data obtained in each sample. Then, according to the average of the 100 variances of each case, we will choose the values of d that we will use to simulate the instances. Figure 1 shows the average values of the 100 variances obtained for each combination of d and v.

Observing the plot, we choose the following values for d: 0.1, 0.2, 0.5 (high-medium variance) and 2, 4 (low variance). Once the values of d are decided, we obtain 10000 samples of a Dirichlet distribution for each combination of d and v, that pretend to be 10000 different instances of COPs for each case. So, we have a set of 150000 instances divided in 15 equally sized groups according to v and d. Each method is run 100 times for each of the simulated instances using two different sample sizes: $M = 1000$ and $M = 10000$. The results provided are the average of the 100 values. Note that as *FRT* does not depend on the sample size, it is only applied 100 times for each instance.

4.1.2 Results

In this section we analyze the performance of the methods and compare them taking into account the different parameters (v, d and M). Firstly, we check if the methods provide useful estimates. Secondly, we use nonparametric tests to rank the methods and study the significant differences among the observed results. Finally, a more qualitative study is developed, emphasizing an important characteristic which is the stability of the methods.

The closeness of the estimates provided by the methods to the value we want to estimate is the most important factor. Obviously, there are methods that estimate better than others, but it does not mean that the estimates provided by the best methods are close enough to the real value. In order to check if we are able to obtain good estimates with these methods, we choose for each combination of v, d, and M, the method that provides the best average estimation over all the 1000000 results (10000 instances x 100 repetitions). In Figure 2 we represent the average estimate obtained with this best method and the real number of local optima (with a dashed line) for each v and d. As expected, the quality of the estimate depends on the parameters of the instances (v and d) and the sample size. For small values of d (0.1, 0.2 and 0.5, i.e. high variance of the sizes of the attraction basins) the estimates are really far from the real number. Furthermore, the higher the number of local optima, the worse the estimate (see Figure 2 (a)-(c)-(e) and (b)-(d)-(f)). For scenarios where the sizes of the attraction basins are quite similar ($d = 2, 4$), the methods provide precise estimates. As regards the sample size, it can be observed that the larger the sample size, the better the estimates provided are (see Figure 2 (a)-(b), (c)-(d) and (e)-(f)). However, this improvement is not enough to reach accurate results in cases of low values of d.

Continuing with the study of the methods, we carry out a statistical analysis to compare the estimates obtained for the different methods. We consider three different scenarios for comparison according to the parameters of the study (M, v, d). In the first scenario considered, the estimates are grouped in two sets according to $M = 1000$ and 10000. The second scenario considers three different sets that contain the estimates of the instances with $v = 100$, 1000 and 10000 local optima. In the last scenario the estimates are grouped in five sets, according to the parameter $d = 0.1, 0.2, 0.5, 2, 4$. A nonparametric Friedman's test with level of significance $\alpha = 0.05$ is used to test if there are statistical significant differences between the estimates provided by the 9 methods in the different scenarios. It provides a ranking of the methods while also giving an

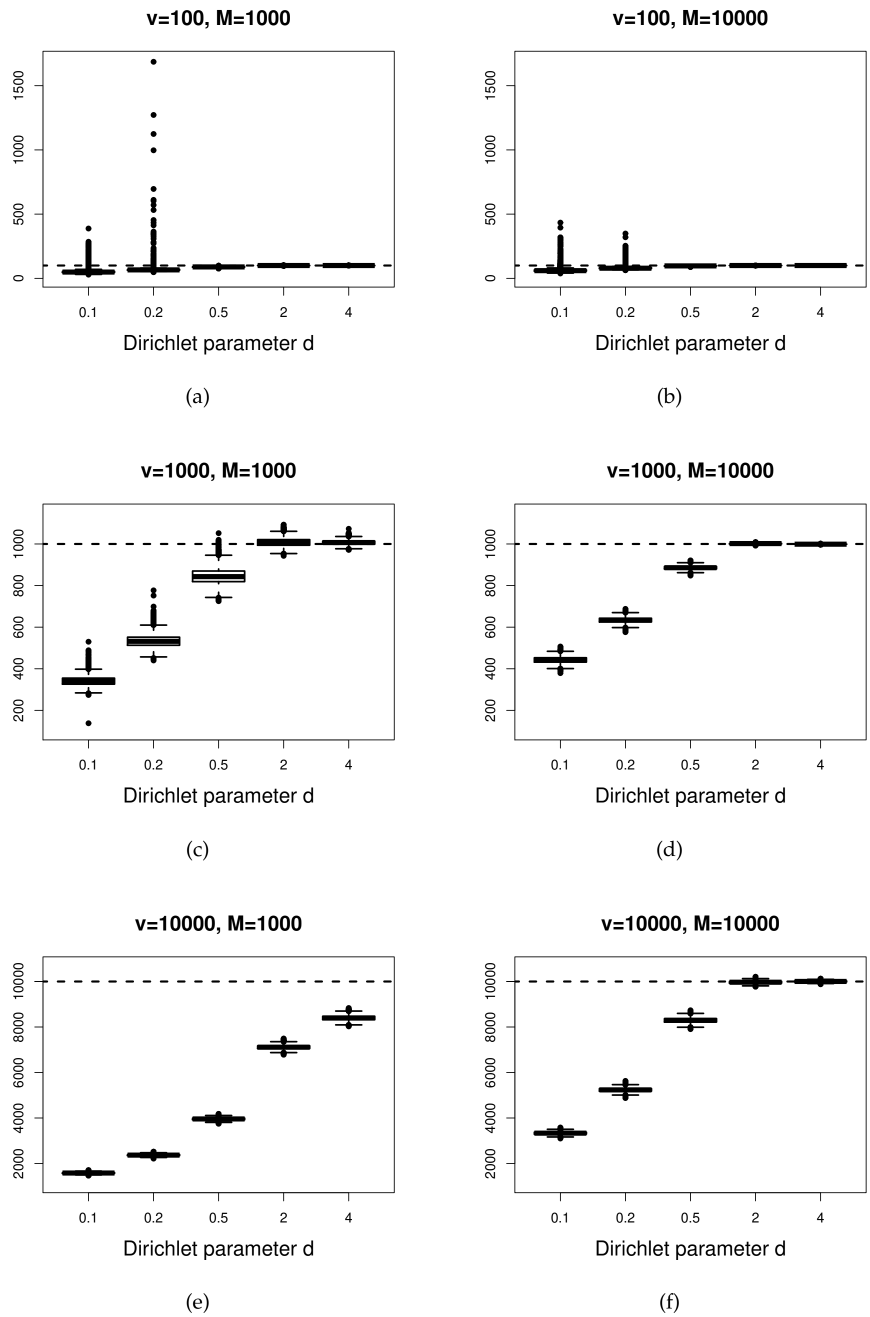


Figure 2: Boxplot that represents the estimations provided by the best method (on average over the 10000 instances x 100 repetitions) for the different values of d. The datasets are created assuming 100 ((a), (b)), 1000 ((c), (d)), and 10000 ((e), (f)) local optima, which are indicated in each figure with a dashed line. In (a), (c) and (e) the methods are applied with sample size 1000, while in (b), (d) and (f) the sample size is 10000.

average rank value for each method. As we always find statistical differences in all the cases, we proceed with a post-hoc test which carries out all pairwise comparisons. Particularly, we use the Holm's procedure fixing the level of significance to $\alpha = 0.05$. Pairwise significant differences are found between all of the methods in the three scenarios.

Table 2 shows the ranking obtained for the methods with the Friedman's test in the first scenario, when using a sample of size $M = 1000$ (first pair of columns) and $M = 10000$ (last pair of columns). The lower the rank, the worse the performance of the method is. So, the methods are ordered from best to worst. Therefore, the best methods when separating the estimates according to the sample size, are *ChaoLee2* and *ChaoBunge*.

In Table 3 the ranking for the methods is shown, but this time for the second scenario, that is, when grouping the estimates for the instances created with $v = 100$ (the first pair of columns), $v = 1000$ (the pair of columns in the middle), and $v = 10000$ (the last pair of columns). In these three cases the Holm's procedure states that significant differences exist among each pair of methods. From this table we can observe that the lower the number of local optima, the better the estimates provided by *Chao1984* are. On the contrary, *ChaoLee2* improves its performance as the number of local optima grows.

Table 2: Average rankings of the methods according to the sample size M

M=1000		M=10000	
Method	Ranking	Method	Ranking
ChaoLee2	7.79	*ChaoBunge*	7.78
ChaoBunge	7.03	*ChaoLee2*	6.88
Chao1984	6.87	*Chao1984*	6.68
ChaoLee1	6.50	*Jckk*	6.31
Jckk	5.67	*ChaoLee1*	5.71
Boots	4.59	*Boots*	4.71
Sch-Cen	3.27	*Sch-Cen*	3.63
MFRT	1.94	*MFRT*	1.88
FRT	1.35	*FRT*	1.41

Finally, Table 4 shows the ranking obtained for the methods when the estimates are separated according to the instances created with the same Dirichlet parameter d. This ranking gives an idea of the method that is better to use according to the variance of the sizes of the attraction basins of the local optima. For high variance (small values of d) the recommended method is *ChaoBunge*, but *ChaoLee2* also provides very good estimates. For instances with quite similar sizes of attraction basins the best method is *ChaoLee2*.

From the statistical analysis, we can conclude that the worst methods in all scenarios are *FRT*, *MFRT* and *Sch-Cen*. On the other hand, we can not conclude that there is a best overall method for all scenarios. However, if we consider the first three best methods, *ChaoBunge* and *ChaoLee2* are always among them. So,

Table 3: Average rankings of the methods according to the number of local optima v

v=100		v=1000		v=10000	
Method	Ranking	Method	Ranking	Method	Ranking
Chao1984	7.25	*ChaoBunge*	8.25	*ChaoLee2*	8.33
ChaoBunge	7.05	*ChaoLee2*	7.30	*ChaoLee1*	7.06
ChaoLee2	6.37	*Chao1984*	6.85	*ChaoBunge*	6.91
Jckk	6.35	*Jckk*	6.03	*Chao1984*	6.24
ChaoLee1	5.40	*ChaoLee1*	5.87	*Jckk*	5.58
Boots	5.21	*Boots*	4.51	*Boots*	4.23
Sch-Cen	3.93	*Sch-Cen*	3.20	*Sch-Cen*	3.22
MFRT	2.22	*MFRT*	2.00	*FRT*	1.92
FRT	1.22	*FRT*	1.00	*MFRT*	1.52

Table 4: Average rankings of the methods according to the Dirichlet parameter d

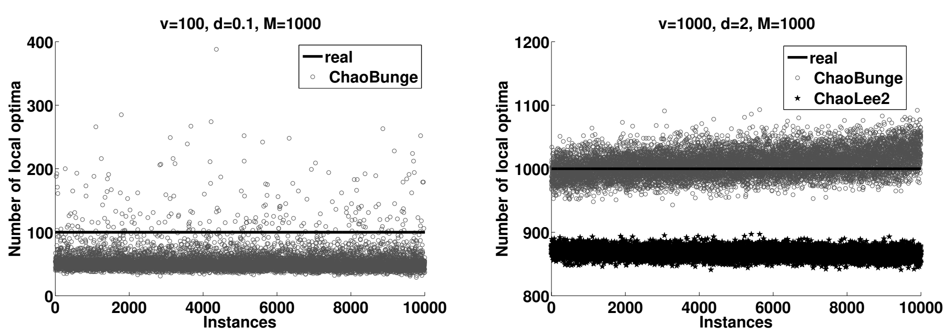
d=0.1		d=0.2		d=0.5	
Method	Ranking	Method	Ranking	Method	Ranking
ChaoBunge	8.37	*ChaoBunge*	8.02	*ChaoBunge*	7.39
ChaoLee2	7.52	*ChaoLee2*	7.31	*ChaoLee2*	7.13
Chao1984	7.07	*Chao1984*	7.01	*Chao1984*	7.08
Jckk	6.30	*Jckk*	6.58	*Jckk*	7.01
ChaoLee1	5.51	*ChaoLee1*	5.82	*ChaoLee1*	6.05
Boots	4.06	*Boots*	4.07	*Boots*	4.13
Sch-Cen	3.06	*Sch-Cen*	3.07	*Sch-Cen*	3.07
MFRT	2.05	*MFRT*	1.90	*MFRT*	1.73
FRT	1.05	*FRT*	1.23	*FRT*	1.40

d=2		d=4	
Method	Ranking	Method	Ranking
ChaoLee2	7.17	*ChaoLee2*	7.53
Chao1984	6.75	*ChaoLee1*	6.84
ChaoBunge	6.59	*ChaoBunge*	6.63
ChaoLee1	6.31	*Chao1984*	5.99
Boots	6.03	*Boots*	4.96
Jckk	5.16	*Jckk*	4.90
Sch-Cen	3.72	*Sch-Cen*	4.33
MFRT	1.80	*MFRT*	2.07
FRT	1.47	*FRT*	1.74

in case of lack of information about the number of local optima of the instance, or the sizes of their attraction basins, using both of them we will probably be obtaining more accurate estimates than by using any other method.

Although the statistical analysis gives a global picture of the performance of the methods, it is also relevant to consider some aspects that are not reflected in the hypothesis tests. One of these aspects is that of stability. Imagine that we have some instances with similar properties (the same number of local optima and similar distribution of sizes of the attraction basins). We are interested in knowing if the method will provide comparable estimates for the different instances, or if they will be extremely different. In the first case we say that the method is stable, while in the second it is unstable. *ChaoBunge* is a very unstable method in certain situations, while the rest of the methods are very stable. Under some circumstances, *ChaoBunge* provides a very good estimate for the number of local optima for most of the instances, but there are instances where the given estimate is very far from v. For example, for $v = 100$, $M = 1000$ and $d = 0.1$ (Figure 3 (a)). In other situations, for example, for $v = 1000$, $M = 1000$ and $d = 2$ (Figure 3 (b)) it is the method that provides the best estimates of the number of local optima for all instances. In order to compare the performance of this method with *ChaoLee2* in these particular scenarios, the estimates provided by *ChaoLee2* are also reflected in Figure 3 (b). We provide all the figures that represent the estimates given by each method for each v and each d, in the following website [1].

We conclude from the tables that *ChaoBunge* is one of the best methods, but we find specific situations where the estimate provided is very far from the real value we want to estimate. In order to know if the estimate provided by *ChaoBunge* is valid, we could also apply other methods, such us *ChaoLee2*, and compare both results. If these estimates are close enough, the one provided by *ChaoBunge* could be accepted. Otherwise, if these estimates are very far one from each other, we are almost sure that *ChaoBunge* is giving a useless estimate. So, we consider that a suitable way for estimating the number of local optima of an instance is not by using a single method, but a comparison of different methods.



(a) (b)

Figure 3: Estimates of the number of local optima provided by *ChaoBunge* (a) and *ChaoBunge* and *ChaoLee2* (b) for 10000 synthetic instances created by sampling $D(100, 0.1)$ and $D(1000, 2)$, respectively. The sample size used in both cases is 1000. In (a) we see the instability of the *ChaoBunge* method, but in (b) it is stable.

[1] http://www.sc.ehu.es/ccwbayes/members/leticia/EstimationNumOpt/EstNumOptFig.html

4.2 Random instances of TSP

4.2.1 Experimental design

In order to contrast our initial conclusions, in this section we work with random instances of the Traveling Salesman Problem. Given a list of cities and their pairwise distances, the aim of this problem is to find the shortest tour that visits each city exactly once, returning to the initial city. Particularly, we work with random instances of the Symmetric Traveling Salesman Problem with 14 and 15 cities. In the symmetric version, the distance from the city A to the city B is considered the same as from B to A. The instances were created by placing 14 and 15 points respectively, uniformly at random on a square of area 100 in an Euclidean space (Gent and Walsh, 1996). Afterwards, we calculated the matrix that gives the Euclidean distance between every pair of cities. We randomly created 500 instances with 14 cities, and 110 instances with 15 cities.

For the purpose of measuring the accuracy of the estimation methods we first calculated the exact number of local optima of the instances when using a 2-exchange neighborhood (N_S). The 2-exchange neighborhood considers that two solutions are neighbors if one is generated by swapping two elements of the other:

$$N_S(\pi_1\pi_2...\pi_n) = \left\{(\pi'_1\pi'_2...\pi'_n) \mid \pi'_k = \pi_k, \forall k \neq i, j, \pi'_i = \pi_j, \pi'_j = \pi_i, i \neq j\right\}.$$

So, applying to each solution of the search space a deterministic local search algorithm (see Algorithm 1 in Section 2) with a 2-exchange neighborhood, the exact number of local optima of the instance and their corresponding sizes of the attraction basins are obtained. Notice that in the symmetric TSP there are $2n$ permutations encoding the same solution. Therefore, we only take into account one of all these different representations, and thus we search in a space of size $(n-1)!/2$.

The different methods for estimating the number of local optima were applied to all the instances considering two sample sizes: $M = 1000$ and 10000. For each instance the methods are repeated 100 times and we evaluate and compare the average estimates of each method.

4.2.2 Results

A first step in the analysis of the methods when applying them to random instances of the TSP is the study of the accuracy of the estimates provided by the methods. Secondly, a parameter d is associated to each instance and, as in the previous section, the performance of the methods is studied again according to d, v and M.

In order to check if the methods provide useful estimates, the average errors of the estimates with respect to the real number of local optima are calculated. Table 5 shows the average relative errors and the standard deviations (in brackets) grouped by the number of cities and the sample size.

A general conclusion deduced from Table 5 is that for $n = 14$ the methods provide better estimates than for $n = 15$. Table 5 also confirms the improvement of the estimates as the sample size grows. It is remarkable that for $n = 15$ cities (higher number of local optima than for $n = 14$) and sample size $M = 1000$, the estimates are very far from the real value, and the standard deviations for these estimates are considerably high. Particularly, *ChaoBunge* has a very high standard deviation when the sample size is 1000. This fact confirms the unstable behavior observed in the previous experiments. The instability of this method is a consequence of the variability on the estimation of the parameter $\hat{\theta}$ (see equation (3) of Section 3.2). If $\beta_1 >> \beta_i$ $(2 \leq i \leq \delta)$, then $\hat{\theta} \approx 0$ and the estimation $\hat{v}_{ChaoBunge}$ in this case is very large and very far from the real value. This occurs when we have a sample where a lot of local optima are seen only once, but there is a small number of local optima seen twice, three times, etc. These particularities are commonly found when the sample size is small with respect to the number of local optima, or even when the variance of the sizes of the attraction basins of the local optima is high.

Table 5: Average relative errors and standard deviations (in brackets) of the estimates provided by the different methods, according to the number of cities n and the sample size M. The range for the real number of local optima of the instances appears in brackets under the number of cities n.

		FRT	*MFRT*	*Sch-Cen*	*Jckk*	*Boots*	*Chao1984*	*ChaoBunge*	*ChaoLee1*	*ChaoLee2*
n = 14 ($34 \leq v \leq 648$)	**M=1000**	96.15 (3.48)	89.20 (17.76)	43.61 ? (76.89)	27.95 ? (71.81)	36.75? (74.53)	26.08 (57.02)	27.15 ? (5949.52)	28.42 (49.49)	**22.70 (45.48)**
	M=10000		87.15 (23.55)	14.29 (29.03)	**5.65 (14.61)**	10.28 (21.50)	6.71 (14.77)	7.34 (15.49)	8.93 (15.99)	7.86 (13.83)
n = 15 ($97 \leq v \leq 1087$)	**M=1000**	97.46 (2.11)	93.04 (11.48)	59.38 (66.19)	44.42 (83.04)	52.87 (73.92)	41.42 (62.66)	59.46 (19247.69)	41.72 (61.08)	**33.78 (61.47)**
	M=10000		91.68 (15.19)	26.47 (47.48)	14.64 (29.51)	20.91 (39.67)	16.33 (24.74)	**14.64 (14.20)**	18.16 (24.32)	16.20 (18.40)

To visualize the performance of the methods, in Figure 4 we represent the estimates provided. We first arrange the instances according to the number of local optima and take 10 groups of 11 instances. For each group we calculate the average estimate of each method. We represent the five methods that provide the best results: *Jckk*, *Boots*, *Chao1984*, *ChaoLee1* and *ChaoLee2*. *ChaoBunge* is removed from the plots because of its instability. Figure 4 shows the average estimates obtained for the instances of the TSP with 15 cities, for sample size 1000 (up) and 10000 (bottom). We observe from the graphs that, when the number of local optima is lower than $400 - 500$, the estimates are close to the real values. However, as the number of local optima grows, the estimates provided by all the methods tend to distance themselves from the real number of local optima. For sample size 1000, it can be clearly seen that in all groups the best method is *ChaoLee2*, while *ChaoLee1*, *Chao1984* and *Jckk* provide similar estimates. For sample size 10000, we observe that *Jckk* is the best method. Additional information about the estimates provided by each method for each instance, can be found in the website [2].

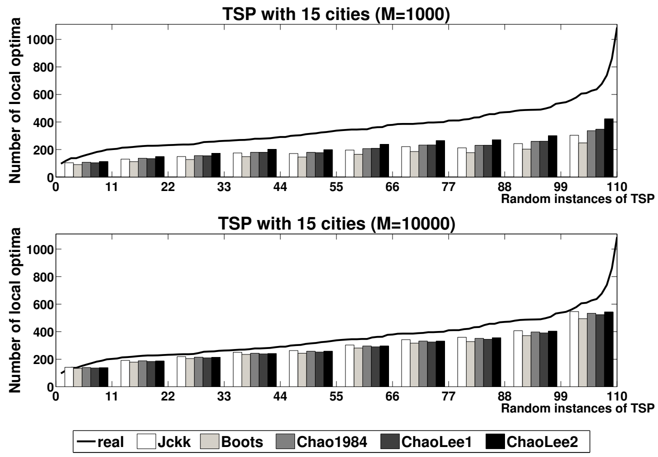


Figure 4: Estimates of the number of local optima provided by the different methods for 110 random instances of the TSP with 15 cities. The 110 instances are arranged according to the number of local optima, and are put in 10 groups of 11 instances each. The average of the 11 estimates is shown by the histogram for each method. The methods consider a sample of size 1000 (top graph) and 10000 (bottom graph). The solid line indicates the number of local optima of the instances.

We compare the methods according to different parameters, as proceeded in the previous section. Firstly, a parameter d is associated to each instance, supposing that the sizes of the attraction basins were created sampling a Dirichlet distribution with that particular d. Due to the fact that we know the number of local optima v of the 610 random instances of the TSP, for each value of v we sample Dirichlet distributions $D(v, d)$, for each $d = 0.1, 0.2, 0.5, 2, 4$. We take 100 samples for each v and d and the variance of the v data is calculated in each sample. On the other hand, the variance of the relative sizes of the attraction basins of the local optima of each of the random instances is calculated. For each instance, we compare the variance of its relative sizes of the attraction basins with the variances obtained when sampling $D(v, d)$, being v the corresponding number of local optima of that instance. We associate to each instance the value of d for which the variance of the instance is closer to the average variance of $D(v, d)$.

A classification of the instances according to d and v is carried out and we realize that most of the instances have low values of d, that is, they have high variances of the sizes of the attraction basins of the local optima. Table 6 shows the number of instances that we associate with the different values of d depending on the different number of local optima. Next, we study the performance of the methods taking into account d, v and M, and we compare it with the results of the previous section. As we have only found one instance with

[2]http://www.sc.ehu.es/ccwbayes/members/leticia/EstimationNumOpt/EstNumOptFig.html

parameter $d = 4$, we analyze it separately. We saw in the previous section that the three methods that provided the best estimates for $d = 0.1, 0.2, 0.5, 2$ as well as for $M = 1000, 10000$, were *Chao1984*, *ChaoLee2* and *ChaoBunge*. Table 7 shows the percentage of instances for which the three best estimates are provided by these three methods, according to d and M. We observe that for $M = 1000$ the percentages are lower than for $M = 10000$ and this is because when $M = 1000$ *ChaoBunge* is more unstable than for $M = 10000$. For small values of d ($d = 0.1, 0.2, 0.5$), and when the sample size used by the methods is 1000, the best method is *ChaoBunge* in more than 77% of the estimates. In 379 of the 610 instances the second best method is *ChaoLee2*, and the third best method is *Chao1984* in 367 of the 610 instances. These results corroborate the conclusions obtained from the analysis of the methods for the synthetic instances (Table 4). When sample size is 10000, for small values of d, *Chao1984* provides very good estimates, and in most of the instances it is the best method. The reason is that almost all of the instances that have been associated a low value of d have also a low number of local optima and, as was seen in Table 3, the best method in these scenarios is *Chao1984*.

We only find one instance with a high value of d, and furthermore, it is the instance that has the highest number of local optima ($v = 1087$). When the methods are applied to this instance with sample size 1000, the best estimates are provided by *ChaoLee2* and *ChaoLee1*. As we saw in Table 4, these are the best methods for d=4. On the other hand, when sample size is 10000, the best methods are *ChaoBunge* and *ChaoLee2*. This matches the results shown in Table 2 (last pair of columns) and Table 3 for $v = 1000$.

Table 6: Number of instances that are assigned different values of d according to v.

	$30 < v < 500$	$500 < v < 1100$
d=0.1	346	0
d=0.2	155	0
d=0.5	81	3
d=2	13	11
d=4	0	1

Table 7: Percentages of the number of instances for which the best three estimates obtained are provided by *Chao1984*, *ChaoBunge* and *ChaoLee2*.

	$M = 1000$	$M = 10000$
d=0.1	80.64%	97.40%
d=0.2	78.71%	99.35%
d=0.5	55.95%	100.00%
d=2	41.67%	95.83%

4.3 Instances of TSP and FSSP

4.3.1 Experimental design

This section is devoted to experiments with real instances of COPs, as well as instances taken from the Taillard's benchmark. We work with 10 instances of the Traveling Salesman Problem (with real distances between cities) and other 10 instances of the Flow Shop Scheduling Problem. The FSSP can be stated as follows: there are n jobs to be scheduled in m machines. A job consists of m operations and the j-th operation of each job must be processed on machine j for a specific processing time without interruption. We consider that the jobs are processed in the same order on different machines, what is known as the Permutation Flow Shop Scheduling Problem (PFSP). The objective of the PFSP is to find a permutation that represents the order in which the jobs have to be scheduled on the machines, minimizing the total flow time.

For the TSP we take the real distances between 14 cities of the continents Africa, America, Asia and Europe, and 14 cities of The United States, Spain and Australia and Pacific cities [3]. For the PFSP we consider instances with 13 jobs and 5 machines, obtained from the well-known benchmark proposed by Taillard [4] that has been commonly used by numerous authors, such as Taillard (1990), Bierwirth and Mattfeld (1999) or Ceberio et al. (2012). The instances of both TSP and PFSP used in this section of the experiments are available in the website [5].

We apply Algorithm 1 to each instance starting from each solution of the search space. Notice that the size of the search space of the TSP instances is $13!/2$, while the instances of the PFSP have a search space of size $13!$. In this section we consider two neighborhoods: 2-exchange and Insert. Two solutions are neighbors under the Insert neighborhood (N_I) if one is obtained by moving an element of the other one to a different place:

$$N_I(\pi_1\pi_2...\pi_n) = \left\{(\pi'_1\pi'_2...\pi'_n) \mid \pi'_k = \pi_k, \forall k < i \text{ and } \forall k > j, \pi'_k = \pi_{k+1}, \forall i \le k < j, \pi'_j = \pi_i\right\}$$
$$\cup \left\{(\pi'_1\pi'_2...\pi'_n) \mid \pi'_k = \pi_k, \forall k < i \text{ and } \forall k > j, \pi'_i = \pi_j, \pi'_k = \pi_{k-1}, \forall i < k \le j\right\}.$$

The different methods for estimating the number of local optima are applied to all the instances using both neighborhoods. The reason why we have considered these two neighborhoods in this section is that they provoke different situations for the estimates obtained with the different methods. As the Insert neighborhood explores at each step more solutions than the 2-exchange neighborhood, the number of local optima obtained when considering the first neighborhood is probabilistically lower than when assuming the second one.

[3]http://www.mapcrow.info
http://locuraviajes.com/blog/wp-content/uploads/2011/08/cuadro-distancias-ciudades-espa?a.gif

[4]http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html

[5]http://www.sc.ehu.es/ccwbayes/members/leticia/EstimationNumOpt/EstNumOptInst.html

4.3.2 Results

In this final section our aim is to extend the previous analysis focusing on the accuracy of the methods and their relation to the sample size. For this purpose we first look for the minimum sample size that allows each method to reach estimates closed to the real number of local optima. We consider that a method that needs a smaller sample size to provide good estimates will be more useful. In addition, we also analyze the effect of the sample size on the methods using small as well as large sample sizes, in order to find the methods that provide better estimates in more realistic situations. That is, when the sample size is very small compared with the real number of local optima of the instance.

In order to study the sample sizes needed to obtain good estimates, we choose for each method the minimum sample size for which at least 95 of 100 estimates provided are closer than 95% from the real number of local optima of each instance under each neighborhood. The algorithm used to obtain the minimum sample sizes starts with $M = 100$. It doubles the value of M, until it succeeds or reaches the maximum sample size considered (6553600 = 100x2^{16}). In case of success for a given M, a bisection procedure is applied until the difference between the last accepted sample size and the previously discarded one is 100. So, it converges to the minimum sample size wanted. We repeat this process 10 times and show the average values.

Tables 8 and 9 show the average sample sizes that the methods need when they are applied to the TSP and PFSP instances, respectively. In both tables, each row represents an instance and a neighborhood. The first half of the tables is related to the Insert operator and the second half to the 2-exchange operator. Inside each group, instances are put in an ascending order according to the number of local optima (first column). In most of the instances the *MFRT* method is not able to fulfill the condition stated (a line is drawn for these cases). Notice that *FRT* is not taken into account because this method does not depend on the sample size and, as we saw in the previous sections, this method provides such bad estimates that we decided to take it out from the study in this section.

Looking at the overall results, one could conclude that the best methods are Jackknife and Bootstrap, because in almost all instances they need a smaller sample size to provide very good estimates. This fact seems to be in conflict with almost all the results obtained in the previous sections, where *Chao1984*, *ChaoBunge*, *ChaoLee1* and *ChaoLee2* seemed to be the most promising. However, this result agrees with that observed in the previous section, where *Jckk* provided better estimates than the rest of the methods for sample size 10000. Therefore, and in order to obtain additional information about the performance of the methods, we decided to study the estimates provided by them when the sample sizes are small. This idea arose when we realized that in real life we have to face problems of such high dimensions that they have a huge number of local optima. So, the sample we are able to deal with is usually tiny compared to the number of local optima. Thus, we are interested in finding methods that do not need such a large sample size to provide a good estimate.

Table 8: Average of the minimum sample sizes obtained for which at least 95 of 100 estimates provided by each of the methods are closer than 95% to the real number of local optima for each TSP instance under Insert and 2-exchange neighborhoods. The minimum sample size obtained for each instance is in bold.

	Number of local optima	*MFRT*	*Sch-Cen*	*Jckk*	*Boots*	*Chao1984*	*ChaoBunge*	*ChaoLee1*	*ChaoLee2*
TSP Insert	1	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	2	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	2	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	5	650	**100**	110	**100**	**100**	**100**	**100**	**100**
	? 9	——	1320	2030	**1220**	1330	1910	1870	1840
	9	353940	?**200**	?250	**200**	**200**	230	**200**	**200**
	12	——	1690	2750	**1650**	1750	2520	2600	2590
	22	——	2740	4330	**2720**	3600	4030	3580	4000
	29	——	**2260**	3360	2270	2670	5240	2860	5480
	32	——	2370	3490	**2320**	2960	3110	2750	2950
TSP 2-exchange	67	——	23270	28810	**22010**	37130	34250	24280	28590
	73	——	522490	**483810**	515930	647950	1095310	793700	1009610
	90	——	55220	**44460**	57090	81340	174680	55320	152200
	92	——	24980	30450	**20020**	39110	23770	21740	23250
	103	——	40450	**26450**	32110	48070	35920	36390	36790
	117	——	179670	**132460**	173490	251070	228310	165270	177680
	188	——	88860	**47150**	66520	129460	75440	74640	70690
	201	——	93970	**45850**	68870	107460	69160	70890	65720
	393	——	224390	**91510**	150410	167600	168980	167560	166950
	455	——	275540	**89380**	?161850	189310	189850	199250	193950

Table 9: Average of the minimum sample sizes obtained for which at least 95 of 100 estimates provided by each of the methods are closer than 95% to the real number of local optima for each FSSP instance under Insert and 2-exchange neighborhoods. The minimum sample size obtained for each instance is in bold.

	Number of local optima	*MFRT*	*Sch-Cen*	*Jckk*	*Boots*	*Chao1984*	*ChaoBunge*	*ChaoLee1*	*ChaoLee2*
FSSP Insert	14	——	3320	5540	**3150**	3640	4720	4830	4760
	70	——	17920	19380	16520	25220	17680	**16390**	18130
	134	——	25710	25730	**18880**	45320	21540	19870	20750
	160	——	47300	**28170**	34350	63310	39850	37040	35380
	190	——	19930	**11630**	13770	32110	16050	15910	15960
	285	——	23820	**9840**	15260	19310	16620	18110	17240
	404	——	29830	**10670**	18430	19880	18790	20100	19270
	461	——	51320	**17550**	31200	35040	34770	36670	33560
	506	——	77700	**24780**	45910	52990	54730	58060	56180
	923	——	137950	**40850**	79530	88750	92460	94730	93780
FSSP 2-exchange	192	——	39410	**19540**	29300	55320	31750	32290	33600
	1643	——	445780	**134260**	254870	264950	264640	285060	264130
	1846	——	628440	**199600**	363320	338380	323990	366380	338730
	1997	——	592030	**177390**	341020	337870	343950	370630	354610
	2130	——	912200	**273490**	527160	508690	521420	576430	539570
	2382	——	763420	**227140**	435560	411080	426840	466190	431230
	2386	——	613130	**179250**	353810	346650	357130	384600	363640
	5119	——	2149000	**643230**	1229450	1098740	1093250	1235370	1128300
	6485	——	1671460	**456690**	927350	863640	875150	994270	900410
	8194	——	2052570	**568480**	1148250	1032760	1058350	1204950	1094970

We apply the methods taking small sample sizes (compared to the number of local optima) and analyze them according to the estimate they provide. We have only considered the instances with more than 100 local optima, without making distinctions between the two neighborhoods. We take sample sizes in the range 50-600 with steps of 50. *MFRT* and *Sch-cen* have been removed from the study because for the instances with the highest number of local optima, the estimates provided by these methods are lower bounds very far from the real values. So we analyze *Jckk, Boots, Chao1984, ChaoBunge, ChaoLee1* and *ChaoLee2*.

Table 10: Best methods obtained from the Friedman's test for the TSP and PFSP according to the sample size. The average relative error of the estimates provided by these methods is also shown.

	Sample size	Best method	Average relative error
	50 ... 350	*ChaoLee2*	22.56
TSP	400 ... 450	*ChaoLee2*, *ChaoBunge*	18.76, 20.05
	500 ... 600	*ChaoBunge*	11.46
PFSP	50 ... 600	*ChaoLee2*	49.95

The methods are applied 100 times for each sample size. We carry out the nonparametric Friedman's test with level of significance $\alpha = 0.05$ to the estimates, grouping them according to the sample size. We observe that there are statistical significant differences between the estimates provided by the six methods for all the sample sizes. We continue with the Holm's procedure which carries out all pairwise comparisons, setting the level of significance to $\alpha = 0.05$. Table 10 shows the best methods obtained from the Friedman's test for the TSP and PFSP according to the sample sizes. For the TSP, and using sample sizes lower than 400, we find that the best method is *ChaoLee2* and significant differences between *ChaoLee2* and the rest of the methods are found. For sample sizes 400 and 450, the best methods are *ChaoLee2* and *ChaoBunge*. There are no significant differences between them, but there are between them and the rest of the methods. For sample sizes larger than 450, the best method in the ranking is *ChaoBunge*, with significant differences between this method and the rest. For the PFSP instances, and for all sample sizes, *ChaoLee2* is the best performing method, and when we study the pairwise significant differences with the Holm's procedure we can see that there are significant differences between *ChaoLee2* and the rest of the methods.

Let's now study in detail the estimates obtained for the two instances with the highest number of local optima. Tables 11 and 12 show the average of 100 estimates provided by each method for small sample sizes (with respect to the number of local optima) for the instances 9 and 10 of PFSP, respectively, when using the 2-exchange neighborhood. These tables show that, when sample size is small, *Boots* and *Jckk* provide worse estimates than the other methods. Obviously, the estimates improve as the sample size grows for all methods, except for *ChaoBunge*. As was seen in previous sections, the *ChaoBunge* method is very unstable. The estimate provided by this method varies significantly depending on the sample size. Notice that although *ChaoLee2*, *ChaoLee1* and *Chao1984* provide the best estimates, these are also far from the real number of local optima.

If we analyze all the results obtained in this section, on the one hand, we find that *Jckk* and *Boots* need a smaller sample size than the rest of the methods to provide very good estimates. On the other hand, *ChaoLee2* and *ChaoBunge*

Table 11: Average of 100 estimates provided by each method for small sample sizes for the instance 9 of PFSP when using 2-exchange neighborhood.

Instance 9 PFSP. Real number of local optima: 6485

Method	M=50	M=100	M=150	M=200	M=250	M=300	M=350	M=400	M=450	M=500	M=550	M=600
Jckk	92	171	241	304	363	418	469	517	563	605	646	687
Boots	64	122	173	220	265	307	344	381	416	449	481	511
Chao1984	508	683	741	756	798	835	879	919	968	1016	1061	1099
ChaoBunge	254	390	508	259	425	607	600	533	1345	53853	519	393
ChaoLee1	576	682	799	847	876	910	950	988	1032	1074	1129	1176
ChaoLee2	789	1009	1221	1293	1290	1292	1341	1378	1441	1493	1584	1664

Table 12: Average of 100 estimates provided by each method for small sample sizes for the instance 10 of PFSP when using 2-exchange neighborhood.

Instance 10 PFSP. Real number of local optima: 8194

Method	M=50	M=100	M=150	M=200	M=250	M=300	M=350	M=400	M=450	M=500	M=550	M=600
Jckk	95	182	262	335	405	470	532	593	648	702	754	803
Boots	66	129	185	241	292	340	386	431	473	512	551	587
Chao1984	1018	1050	1048	1077	1150	1194	1240	1278	1334	1367	1405	1445
ChaoBunge	341	476	753	1634	5392	1144	5634	1211	1165	1279	4140	540
ChaoLee1	701	1035	1055	1061	1140	1200	1274	1323	1402	1443	1492	1543
ChaoLee2	745	1195	1195	1247	1396	1492	1633	1718	1858	1915	1992	2080

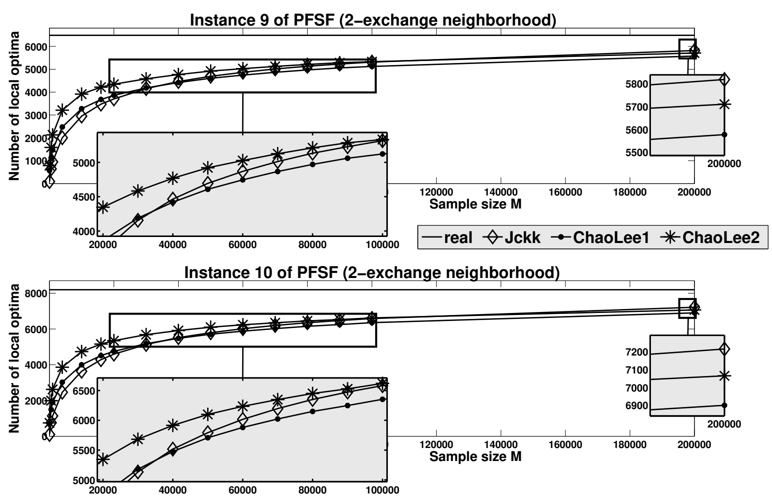


Figure 5: Average of 100 estimates obtained by the different methods for the instance number 9 (up) and instance number 10 (bottom) of PFSP using the 2-exchange neighborhood as the sample size grows.

are considered the best methods for small sample sizes. So, we suspect that there is a threshold for the sample size where the estimates provided by *ChaoLee2* and *ChaoBunge*, or even *ChaoLee1* and *Chao1984*, are worse when compared with the estimates provided by *Jckk* and *Boots*.

These suspicions motivate us to represent the estimates obtained by the different methods for the two instances with the highest number of local optima as the sample size grows. Here, we just plot the estimates corresponding to *Jckk*, *ChaoLee1* and *ChaoLee2* to see the threshold mentioned more clearly. Nevertheless, more detailed graphs are available in the website [6]. In Figure 5 we take into account very small sample sizes as well as high sample sizes (from $M = 50$ to $M = 200000$). We observe that, when the sample size is small, the best methods are *ChaoLee2* and *ChaoLee1*. There is a threshold (for sample size between 20000 and 60000) where *Jckk* improves its estimates compared with those provided by *ChaoLee1*, and for sample size between 80000 and 200000, the estimates given by *Jckk* also improve those provided by *ChaoLee2*. The reason is that with a small growth in the sample size, the estimate provided by *Jckk* improves more than the estimates given by *ChaoLee1* or ChaoLee2. So, our recommendation is to use the *Jckk* and *Boots* methods when we are able to work with large sample sizes. But, if we suspect that our sample is very small compared to the real number of local optima, the best methods to apply are *ChaoLee2* or *ChaoLee1*.

5 Conclusions and future work

In this paper we have reviewed different methods for estimating the number of local optima of instances of combinatorial optimization problems. Our main contribution is the comparison of methods in the optimization field with some methods previously used for estimating the number of species in a population in the field of statistics.

The methods have been applied to three datasets: synthetic instances, instances of the TSP with 14 and 15 cities taken at random, and instances of TSP with real distances between cities and instances of FSSP taken from the well-known Taillard's benchmark. The main conclusions observed for all the methods in the three scenarios are the following:

1. When the attraction basins are similar in size, the methods provide estimates close to the real number of local optima. Of course, the higher the sample, the more precise the estimates.

2. The further the sizes of the attraction basins from the uniformity, the worse the estimates. In fact, in the real instances (where the variance of the sizes of the attraction basins is very high) the predictions are really far from the real number of local optima.

Based on the results observed through the experiments, we provide the following rules of thumb:

- If we are able to take a sample of large size with respect to the number of local optima, we recommend using *Jckk*.

[6]http://www.sc.ehu.es/ccwbayes/members/leticia/EstimationNumOpt/EstNumOptFig.html

- If we suspect that our sample size is small (with respect to the number of local optima), we recommend using *ChaoBunge* and *ChaoLee2*. Due to the instability observed for *ChaoBunge*, both methods should be executed independently. If the results provided are close, *ChaoBunge* is usually the choice. Otherwise, select *ChaoLee2*.

- If analyzing the sample we realize that each (or most) of the initial solutions reach different local optima, that is $r = M$ and $\beta_1 = M$, none of the previous methods can be applied. In this case, we can base our estimator on the proportion of local optima over the sample (Caruana and Mullin, 1999; Grundel et al., 2007).

We consider two different lines of future work. In a first step, we plan to improve the quality of the estimates of some of the presented methods. For example, methods such us *ChaoBunge* or *ChaoLee2* depend on a cut-off value that fixes the border between rare and abundant species. We think that this cut-off value could be properly tuned for each instance and sample size instead of being a fixed number. Our second line is related to the design of specific estimation methods for COPs. As we have seen, the evaluated methods provide unacceptable estimates in real instances. We conjecture that this is due to, on the one hand, the fact that the methods have not been explicitly designed to calculate the number of local optima but the number of species, and these are different problems. On the other hand, they do not use all the information that we have at hand when we try to calculate the number of local optima in a COP. For example, the search space is structured and therefore, it could be divided based on a certain criterion, performing estimates for each chunk. Also, the number of steps (solutions) traversed from the initial solution to the local optima can provide valuable information about the relative sizes of the attraction basins.

6 Acknowledgements

This work has been partially supported by the Saiotek, Etortek and Research Groups 2007-2012 (IT- 242-07) programs (Basque Government),TIN2010-14931 and Consolider Ingenio 2010 - CSD2007-00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute). Leticia Hernando holds a grant from the Basque Government.

References

Albrecht, A., Lane, P., and Steinhofel, K. (2008). Combinatorial landscape analysis for k-SAT instances. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2498 –2504.

Albrecht, A., Lane, P., and Steinhofel, K. (2010). Analysis of Local Search Landscapes for k-SAT Instances. *Mathematics in Computer Science*, 3(4):465–488.

Bierwirth, C. and Mattfeld, D. C. (1999). Production Scheduling and Rescheduling with Genetic Algorithms. *Evolutionary Computation*, 7(1):1–17.

Bunge, J. and Fitzpatrick, M. (1993). Estimating the Number of Species: A Review. *Journal of the American Statistical Association*, 88(421):364–373.

Burnham, K. P. and Overton, W. S. (1978). Estimation of the Size of a Closed Population when Capture Probabilities vary Among Animals. *Biometrika*, 65(3):625–633.

Caruana, R. and Mullin, M. (1999). Estimating the Number of Local Minima in Big, Nasty Search Spaces. In *In Proceedings of IJCAI-99 Workshop on Statistical Machine Learning for Large-Scale Optimization*.

Ceberio, J., Irurozki, E., Mendiburu, A., and Lozano, J. A. (2012). A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence*, 1(1):103–117.

Chao, A. (1984). Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics*, 11(4):265–270.

Chao, A. and Bunge, J. (2002). Estimating the Number of Species in a Stochastic Abundance Model. *Biometrics*, 58(3):531–539.

Chao, A. and Lee, S.-M. (1992). Estimating the Number of Classes via Sample Coverage. *Journal of the American Statistical Association*, 87(417):210–217.

Chao, A. and Yang, M. C. K. (1993). Stopping Rules and Estimation for Recapture Debugging with Unequal Failure Rates. *Biometrika*, 80(1):193–201.

Eremeev, A. V. and Reeves, C. R. (2002). Non-parametric Estimation of Properties of Combinatorial Landscapes. In Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., and Raidl, G., editors, *Applications of Evolutionary Computing*, volume 2279 of *Lecture Notes in Computer Science*, pages 31–40. Springer Berlin / Heidelberg.

Eremeev, A. V. and Reeves, C. R. (2003). On Confidence Intervals for the Number of Local Optima. In *In Proceedings of EvoWorkshops 2003*, pages 224–235.

Esty, W. W. (1985). Estimation of the Number of Classes in a Population and the Coverage of a Sample. *Mathematical Scientists*, 10:41–50.

Fonlupt, C., Robilliard, D., Preux, P., and Talbi, E.-G. (1999). Fitness Landscapes And Performance of Meta-Heuristics. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 257–268. Kluwer Academic Publishers.

Garnier, J. and Kallel, L. (2001). How to detect all maxima of a function. In *Theoretical aspects of evolutionary computing*, pages 343–370. Springer-Verlag, London, UK.

Gent, I. P. and Walsh, T. (1996). The TSP Phase Transition. *Artificial Intelligence*, 88(1-2):349–358.

Good, I. J. (1953). The Population Frequencies of Species and the Estimation of Population Parameters. *Biometrika*, 40(3/4):237–264.

Grundel, D., Krokhmal, P., Oliveira, C., and Pardalos, P. (2007). On the Number of Local Minima for the Multidimensional Assignment Problem. *Journal of Combinatorial Optimization*, 13:1–18.

Harris, B. (1959). Determining bounds on integrals with applications to cataloging problems. *Annals of Mathematical*, 30(2):521–548.

Hertz, A., Jaumard, B., and de Aragão, M. P. (1994). Local optima topology for the k-coloring problem. *Discrete Applied Mathematics*, 49(1-3):257–280.

Kirkpatrick, S. and Toulouse, G. (1985). Configuration space analysis of travelling salesman problems. *Journal de Physique*, 46(8):1277–1292.

Link, W. A. (2003). Nonidentifiability of Population Size from Capture-Recapture Data with Heterogeneous Detection Probabilities. *Biometrics*, 59(4):1123–1130.

Mattfeld, D. C. and Bierwirth, C. (1999). A Search Space Analysis of the Job Shop Scheduling Problem. *Annals of Operations Research*, 86:441–453.

Reeves, C. and Aupetit-Bélaidouni, M. (2004). Estimating the Number of Solutions for SAT problems. In Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tino, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 101–110. Springer Berlin / Heidelberg.

Reeves, C. R. (2001). Statistical Properties of Combinatorial Landscapes: An Application to Scheduling Problems. In *MIC'2001: Proceedings of the 4th Metaheuristic International Conference*, pages 691–695.

Reeves, C. R. (2002). The "Crossover Landscape" and the Hamming Landscape for Binary Search Spaces. In *Foundations of Genetic Algorithms 7*, pages 81–98. Morgan Kaufmann.

Reeves, C. R. and Eremeev, A. V. (2004). Statistical analysis of local search landscapes. *Journal of the Operational Research Society*, 55(7):687–693.

Schwarz, C. J. and Seber, G. A. F. (1999). Estimating Animal Abundance: Review III. *Statistical Science*, 14(4):427–456.

Seber, G. A. F. (1986). A Review of Estimating Animal Abundance. *Biometrics*, 42(2):267–292.

Seber, G. A. F. (1992). A Review of Estimating Animal Abundance II. *International Statistical Review*, 60(2):129–166.

Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47(1):65–74.

Tomassini, M., Vérel, S., and Ochoa, G. (2008). Complex-network analysis of combinatorial spaces: The NK landscape case. *Physical Review E*, 78(6):66–114.

Vose, M. D. and Wright, A. H. (1995). Stability of Vertex Fixed Points and Applications. In *Foundations of Genetic Algorithms 3*, pages 103–113. Morgan Kaufmann.

Wang, J.-P. (2010). Estimating the species richness by a Poisson-Compound Gamma model. *Biometrika*, 97(3):727–740.

Wang, J.-P. and Lindsay, B. (2008). An exponential partial prior for improving NPML estimation for mixtures. *Statistical Methodology*, 5:30–45.

Wang, J.-P. Z. and Lindsay, B. G. (2005). A Penalized Nonparametric Maximum Likelihood Approach to Species Richness Estimation. *Journal of the American Statistical Association*, 100(471):942–959.