# The Natural Bias of Artificial Instances

**Imanol Unanue · María Merino · Jose A. Lozano**

**Abstract** Many exact and metaheuristic algorithms presented in the literature are tested by comparing their performance in different sets of instances. However, it is known that when these sets of instances are generated randomly, they neither have nor fulfill the features the authors believe they do, which implies that wrong conclusions were made. In this paper, we reinforce the importance of analyzing randomly generated instances by sampling the problem coefficients uniformly at random. We generate instances of the Unconstrained Binary Quadratic Problem and the Number Partitioning Problem. In both cases, we verify that the generated set of instances do not represent a uniform set of instances of the problem. We have conducted several experiments to quantify the number of different rankings of solutions that the problems can generate. We have classified those rankings according to how often each ranking is sampled, how many local optimal solutions each ranking has, and how similar they are.

Keywords:

Unconstrained Binary Quadratic Problem, Number Partitioning Problem, Rankings of Solutions, Sampling, Instances.

Imanol Unanue
Department of Computer Science and Artificial Intelligence
University of the Basque Country. Donostia-San Sebastian, Spain
E-mail: imanol.unanue@ehu.eus

María Merino
Basque Center for Applied Mathematics. Bilbao, Spain
Department of Mathematics. University of the Basque Country, UPV/EHU. Leioa, Spain.

Jose A. Lozano
Basque Center for Applied Mathematics. Bilbao, Spain
Department of Computer Science and Artificial Intelligence. University of the Basque Country, UPV/EHU. Donostia-San Sebastian, Spain

## 1 Introduction

In the field of Combinatorial Optimization, most researchers have designed and developed exact and metaheuristic algorithms to solve Combinatorial Optimization Problems (COPs) efficiently. Mainly, to evaluate the performance of the proposed algorithms, researchers compare their proposal with several state-of-the-art algorithms in the literature to solve a particular problem. To do so, they run all the methods over a set of instances and they use a metric to evaluate and compare their performance.

One of the main critical steps of this process is to choose an appropriate set of instances (of the considered problem) to make a fair comparison. In other words, the set of instances must be representative of all the possible scenarios that the problem can generate and the algorithms cannot use any information about the instances in advance. There are two types of instances: real-world instances (instances that represent real scenarios) and artificial instances. In this work, we will focus on the latter group.

In order to obtain random instances of a problem, researchers determine the specific values of a set of parameters to define a case of the problem. In general, those values can be conducted in two ways: selecting values to define problem instances that satisfy some properties (to study "easy" and "hard" scenarios, for example), or generating random values from uniform distributions.

The main motivation of this paper is to theoretically analyze artificial instances generated by researchers in the literature. Particularly, one of our idyllic goals is to present a method to associate an "instance-algorithm" according to their properties in order to minimize the cost of solving them. In the literature, this idea is also presented as the Algorithm Selection Problem [1, 2]. Nevertheless, generating artificial instances knowing very little or nothing about them and evaluating the performance of the algorithms using them can induce some wrong assumptions, ideas and/or results. In this work, we will focus on the following statement: sampling the coefficients to describe an instance of a problem uniformly at random is not equivalent to sampling instances of the problem uniformly at random.

The main objective of this paper is not only to show that sampling coefficients (parameters) to describe instances of a particular problem uniformly at random generates "biased fitness functions" (in terms of frequency), but also to extract features and characteristics of the rankings of solutions (orderings of the solutions with respect to a fitness function) generated by this process. The study of the features of the generated rankings will allow us to understand why some algorithms perform better in most of the instances of the studied problem.

To the best of our knowledge, there is one "initial" reference which is closely related to our objectives: [3]. In the mentioned article, the authors prove that, when the algorithm only considers the ranking of the solutions to compare them, sampling in the space of coefficients uniformly at random is not equivalent to sampling instances in the space of functions uniformly at random. To do so, the authors consider the Linear Ordering Problem (and, briefly, the

Quadratic Assignment Problem and the Permutation Flowshop Scheduling Problem) and analyze the instances generated by sampling coefficients uniformly at random. They observe that, from all the possible rankings that can be generated by the definition of the problem, there are some rankings which are sampled more frequently. Furthermore, the authors define a grouping of the rankings according to their frequency in the sample and they analyze the inequalities that each group of rankings induces. Based on that work, in [4] the authors count exactly how many rankings of solutions the Linear Ordering Problem and the Traveling Salesman Problem can exactly generate and which rankings of solutions can be obtained by both problems.

The authors of [3] and [4] illustrate their conclusions considering permutation-based COPs. In this paper, the considered COPs to carry out a similar analysis are the Unconstrained Binary Quadratic Problem (UBQP) and the Number Partitioning Problem (NPP). The UBQP is a binary-based COP that has been extensively studied in the literature because of its relevance and applications; many other problems can be reformulated as particular cases of the UBQP, such as the Maximum Independent Set, the Maximum Cut Problem and (the second problem studied in this work) the Number Partitioning Problem, among others [5–9]. The study of the UBQP is analogous to the study of 2-degree pseudo-Boolean functions and, even if the definition of the UBQP is not complex, it is the NP-hard problem with the lowest possible degree polynomial function. Many metaheuristic algorithms have been proposed in the literature not only to solve the UBQP, but to solve its particular cases and generalizations, such as the Maximum Independent Set [10] and the multi-objective UBQP [11] and to develop theoretical analyses of them [12–14]. In this paper, we study the artificial instances of the UBQP and the NPP generated by sampling coefficients uniformly at random.

This paper is organized as follows. In Section 2 the definitions of the rankings of solutions, the UBQP and the NPP are presented. In Section 3 the experimental results over the UBQP are shown and an analysis of the obtained results is discussed. In Section 4 the experiments for the NPP are detailed and analyzed. Finally, Section 5 presents conclusions and future work.

## 2 Preliminary concepts

In this section, we present the main definitions of this work: the ranking of solutions, the UBQP and the NPP. Throughout this article, let us consider $\Omega = \{0,1\}^n$ where $n$ is the size of the search space, $x = x_1 \dots x_n \in \Omega$ and let us assume that all the studied fitness functions are injective functions, in order to avoid several notation issues.

**Definition 1** Let $f : \Omega \longrightarrow \mathbb{R}$ be a fitness function, where $\Omega$ is a finite search space. A **ranking of solutions (defined by $f$)** is an ordered list of all the solutions of $\Omega$ according to their fitness function values: the first solution is the solution with the highest fitness function value, the second solution is the second highest fitness function value, and so on. Let us denote by $r_f$ the ranking defined by $f$.

*Example 1* Let $n = 2$ and $f(x) = x_1 + 2x_2 - 4x_1x_2$. Then, the fitness function values are the following:

| $x$ | 00 | 10 | 01 | 11 |
|---|---|---|---|---|
| $f(x)$ | 0 | 1 | 2 | $-1$ |

and the ranking generated by $f$ is:

$$r_f = \begin{bmatrix} 01 \\ 10 \\ 00 \\ 11 \end{bmatrix}.$$

By the definition of the ranking of solutions, fitness functions can be studied as ranking generators. Therefore, a COP can be interpreted as the set of rankings generated by its definition. The most important feature of the rankings is the relative order of the solutions according to the fitness function, not their exact fitness function values. Most local search based algorithms and any evolutionary algorithm that uses tournament selection or ranking selection consider fitness functions as ranking generators.

In addition, two different fitness functions can have the same ranking of solutions.

*Example 2* Let $g(x) = 3x_1 + 4x_2 - 10x_1x_2$. Then, $g$ and the previous $f$ functions generate the same ranking of solutions.

| $x$ | 00 | 10 | 01 | 11 |
|---|---|---|---|---|
| $f(x)$ | 0 | 1 | 2 | $-1$ |
| $g(x)$ | 0 | 3 | 4 | $-3$ |

$$\begin{cases} f(01) > f(10) > f(00) > f(11) \\ g(01) > g(10) > g(00) > g(11) \end{cases} \Longrightarrow r_f = r_g = \begin{bmatrix} 01 \\ 10 \\ 00 \\ 11 \end{bmatrix}.$$

Moreover, for any fitness function $f$ and a real constant $c$, the rankings generated by $f$, $f + c$ and $c \cdot f$ are the same: $r_f$. This property is considered in Sections 3 and 4.

Therefore, even if the number of fitness functions is infinite, when $\Omega = \{0, 1\}^n$, the exact number of different rankings of solutions is $2^n!$. In Section 3, we present the exact procedure to measure all the rankings of solutions generated by the UBQP. Unfortunately, because the domain of the coefficients of the NPP is discrete, the same procedure cannot be used for the NPP.

Now, let us introduce the Unconstrained Binary Quadratic Problem.

**Definition 2 Unconstrained Binary Quadratic Problem (UBQP).** Let $n$ be the size of the problem and $M = [\tilde{a}_{ij}]_{i,j=1}^{n}$ a matrix of real values of size $n \times n$. The fitness function is described in the following way:

$$f(x) = xMx^t = \sum_{i,j=1}^{n} \tilde{a}_{ij} x_i x_j.$$

The objective of the UBQP is to find

$$x^* = \arg\max_{x \in \Omega} f(x).$$

The analysis of the UBQP is equivalent to the analysis of 2-degree pseudo-Boolean functions. It is common to assume, without loss of generality, that the matrix $M$ is upper triangular or symmetric. When the matrix $M$ is upper triangular, we can rewrite the fitness function in the following way:

$$f(x) = \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{ij} x_i x_j$$

where $a_i = \tilde{a}_{ii}$ and $a_{ij} = \tilde{a}_{ij} + \tilde{a}_{ji}$.

Finally, let us introduce the definition of the Number Partitioning Problem.

**Definition 3 Number Partitioning Problem (NPP).** Let $Z = \{z_1, \ldots, z_n\}$ be a set of non-negative integer numbers. The objective of the problem is to find a subset $P$ of $Z$ such that the difference between the sum of the values of $P$ and $Z \backslash P$ is minimized:

$$\left| \sum_{z_i \in P} z_i - \sum_{z_i \in Z \backslash P} z_i \right|.$$

That is to say, for any binary solution $x_1 \ldots x_n$, if we denote $x_i = 1$ if $z_i \in P$ and $x_i = 0$ if $z_i \in Z \backslash P$, we want to minimize the following difference:

$$f(x_1 \ldots x_n) = \left| \sum_{x_i=1} z_i - \sum_{x_i=0} z_i \right| = \left| \sum_{i=1}^{n} z_i - 2 \sum_{i=1}^{n} z_i x_i \right|.$$

If there exists a solution $x'$ such that $f(x') = 0$, then $x'$ is the optimal solution and $Z$ has a perfect partition. If there exists a solution $x'$ such that $f(x') = 1$, then $x'$ is the optimal solution.

In order to avoid several trivial situations, let us assume that $z_i \neq 0$, for any $i$ value. NPP can be modeled as an instance of an UBQP. To do so, the $f^2$ fitness function is calculated, instead of $f$. This variation does not affect the relative comparisons among the solutions: for any two solutions $x$ and $y$, $f(x) > f(y) \iff f^2(x) > f^2(y)$ due to the non-negativity of the numbers. Hence, they produce the same order of the solutions according to their fitness function value or, to simplify, they produce the same ranking of solutions.

For that reason, any algorithm based on the ranking of solutions will behave similarly for the $f$ and $f^2$ fitness functions. In order to simplify the notation, let us denote $c = \sum_{i=1}^{n} z_i$. So,

$$
\begin{aligned}
f^2(x_1 \dots x_n) = & \left( c - 2 \sum_{i=1}^{n} z_i x_i \right)^2 \\
& = c^2 - 4c \left( \sum_{i=1}^{n} z_i x_i \right) + 4 \left( \sum_{i=1}^{n} z_i x_i \right)^2 \\
& = c^2 + 4 \sum_{i=1}^{n} z_i (z_i - c) x_i + 8 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} z_i z_j x_i x_j .
\end{aligned}
$$

Consequently, we can model this problem as an UBQP. Dropping the additive constant $c^2$ and defining

$$a_{ii} = 4z_i(z_i - c) \text{ and } a_{ij} = 8z_i z_j \ (i < j)$$

an equivalent UBQP is obtained.

## 3 Experimental analysis of the rankings of the UBQP

In this section, we have conducted some experiments on the rankings of solutions generated by the UBQP (similar to the experiments carried out in the work [3]). The objectives of our experiments are to study those rankings in terms of their frequency when the coefficients of the UBQP matrix are sampled uniformly at random and to extract characteristics of them.

The experiments conducted in this section are done for the case $n = 3$. The main drawback of our experiments is that for $n \geq 4$ it is not computationally tractable. Note that when $n = 4$, the cardinality of the space of possible rankings of solutions is $2^4!$.

First of all, we have exhaustively checked how many rankings of solutions can be generated by instances of the UBQP. By definition of the rankings of solutions and the definition of the UBQP, each ranking generates a system of inequalities, some of which are inconsistent.

*Example 3* The following ranking $r_f$ cannot be generated by any instance of the UBQP.

$$r_f = [111\ 100\ 010\ 001\ 110\ 101\ 011\ 000]^T$$

Let us prove by reduction ad absurdum. On the one hand, the solution 111 being the optimal solution means that

$$a_1 + a_2 + a_3 + a_{12} + a_{13} + a_{23} > 0.$$

On the other hand,

**Table 1** Number of different rankings of the UBQP ($n = 3$) ordered by their frequencies in the sample and the percentage they represent.

| Size of the sample | 10% | 20% | 30% | 40% | 50% | 60% | 70% | 80% | 90% | 100% |
|---|---|---|---|---|---|---|---|---|---|---|
| Number of different rankings | 89 | 220 | 418 | 672 | 1006 | 1469 | 2221 | 3609 | 6621 | 24192 |

$$\left.\begin{array}{l} f(100) > f(110) \Longrightarrow a_2 + a_{12} < 0 \\ f(010) > f(011) \Longrightarrow a_3 + a_{23} < 0 \\ f(001) > f(101) \Longrightarrow a_1 + a_{13} < 0 \end{array}\right\} \Longrightarrow$$
$$\Longrightarrow a_1 + a_2 + a_3 + a_{12} + a_{13} + a_{23} < 0,$$

which contradicts the first inequality QED.

So, before starting with the experimental analysis, we have considered all the possible rankings of solutions ($2^3! = 40320$ rankings) one by one and we have analyzed which systems of inequalities are inconsistent and which are solvable. After checking all the rankings of solutions, we have observed that 40% of the rankings cannot be generated by the UBQP (exactly, 16128 rankings). Therefore, the space of rankings of solutions generated by the UBQP is formed by 24192 rankings.

The next step is to generate a representative sample of instances of the UBQP by sampling the coefficients of the UBQP matrix uniformly at random. For $n = 3$, the UBQP requires 6 coefficients to describe an instance: $a_1, a_2, a_3, a_{12}, a_{13}, a_{23}$. The considered space to generate the coefficients of the UBQP matrix to generate the representative sample is the hypercube of dimension 6 centered at the origin[1]: $[-0.5, 0.5]^6$ (to avoid the unbounded space $\mathbb{R}^6$). In order to have a representative sample of the rankings generated by the UBQP, initially we have generated 5 million rankings and then the sample size has been increasing by 1 million until all the possible 24192 rankings have been generated at least once. After generating a sample of 27 million rankings, all the rankings have been generated at least once. In Figure 1, we have ordered the 24192 rankings according to the number of times that each ranking has been generated.

In Table 1, a summary of the number of different rankings of solutions that have been sampled according to their frequencies in the sample are shown. In this table, we take into account the sample, order the rankings by their frequency, and observe the deciles: that is to say, how many of them represent $(10d)$% of the sample, for $d = 1, \dots, 10$. For example, the 89 most frequent rankings of the generated sample represent approximately 2.7 million of the generated rankings (10%); the 220 most frequent rankings of the generated sample represent approximately 5.4 million of the generated rankings (20%); and so on. It is clear that a few rankings represent most of the sample, which clearly means that there are rankings which are more intriguing to analyze.

To see the main features of the most frequent rankings, we have focused on three characteristics: number of local optimal solutions with respect to

[1] Throughout this work, the function considered to generate random numbers is *RandomReal*, defined in the software system *Wolfram Mathematica*.

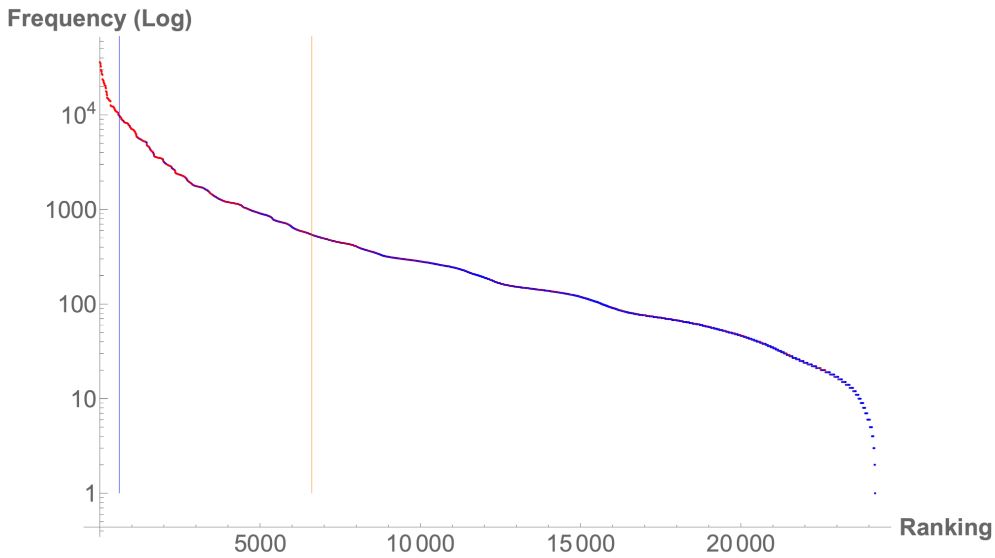


**Fig. 1** Frequency of the 24192 rankings of the UBQP ($n = 3$) generated in a 27M size sample, in descending order of frequency (Y axis in logarithmic scale). Considering the Hamming distance 1, the red points represent the rankings with one local optimal solution (the global optimum). The blue points represent the rankings with two local optimal solutions. Finally, the orange points represent the rankings with three local optimal solutions. The colored vertical lines indicate the most frequent ranking with two and three optimal solutions.

the Hamming distance, the similarities of the rankings and the probability of occurrence.

First, we have calculated the number of local optimal solutions of each ranking according to the Hamming distance. For the Hamming distance, two solutions are neighbors if the distance between them is 1. In Figure 1, the colors represent the number of local optimal solutions of each ranking.

We observe that the most frequent rankings have only one optimal solution, with significant difference with the rest of rankings. In the ordered list of rankings with respect to their occurrence in the sample, the most frequent ranking with two local optimal solutions is placed in the 607th position (its frequency in the sample is 9839). Furthermore, the most frequent ranking with three local optimal solutions is placed in the 6617th position (its frequency in the sample is 540). This is very intriguing knowing that, from all the possible rankings generated by the UBQP, most of the rankings have two optimal solutions. In Table 2, the number of possible rankings and sampled rankings are shown. It is clear that the distribution of the number of rankings with one, two or three local optimal solutions and the distribution of the sampled rankings with one, two or three local optimal solutions are completely different. For the UBQP, 6912 rankings have one local optimal solution (28.6% of the rankings), 15840 rankings have two optimal solutions (65.5% of the rankings) and the rest of rankings have three optimal solutions (5.9% of the rankings). But in

**Table 2** Number of the rankings generated by the UBQP for $n = 3$. Each row groups the rankings according to the number of local optimal solutions. In the second and third columns, the number and percentage of different rankings generated by the UBQP are shown. In the fourth and fifth columns, the number of sampled rankings are shown (from the 27M size sample). In the last column, the 95% confidence interval (CI) of the number of sampled rankings is shown.

| | Rankings (24192) | | Sampled rankings (27M) | | |
|---|---|---|---|---|---|
| L. Opt. Sol. | # | % | # | % | 95% CI |
| 1 | 6912 | 28.57 | 21535236 | 79.760 | (79.745, 79.775) |
| 2 | 15840 | 65.48 | 5318316 | 19.697 | (19.682, 19.712) |
| 3 | 1440 | 5.95 | 146448 | 0.542 | (0.540, 0.545) |

the generated sample for the UBQP, the majority of the rankings generated by sampling coefficients uniformly at random have one local optimal solution (79.76%), 19.69% of the rankings have two local optimal solutions and very few have three local optimal solutions (0.542%).

Next, based on the generated sample of rankings, we have calculated the exact frequency/probability of generating a specific ranking of solutions by sampling the coefficients of the UBQP matrix uniformly at random. This measures exactly the "regions" in which each ranking is generated by the UBQP. To do so, we have calculated the hypervolume of each ranking in the defined hypercube $[-0.5, 0.5]^6$; that is to say, we calculate the regions of the hypercube in which all the points of a specific region generate a ranking $r_f$. So, we divide the hypercube in 24192 regions. Based on the previous results (Table 2), we expected in advance that the measures of each region of the hypercube would not be the same.

To calculate the hypervolume of each region, we have considered the system of inequalities that each ranking defines and calculate the implicit region defined by the system of inequalities and the hypercube $[-0.5, 0.5]^6$. Consequently, the hypervolume is obtained integrating 1 in the calculated region, and because the hypervolume of the hypercube is 1, the obtained result is also the probability of generating a specific ranking of solutions by sampling the coefficients of the UBQP matrix uniformly at random. However, even if this process is exact, it is worth mentioning that there have been some computational issues in the calculation of the exact hypervolumes of some rankings. We believe that the issues are due to the dimension of the space (6) and some particularly small regions. Therefore, sampling would provide an estimation of the hypervolume of each region generated by the UBQP.

A main observation of these hypervolumes is that there exists a symmetry of the rankings. Two rankings of solutions are symmetric and have the same hypervolume value if the difference between both rankings is a permutation of the bits (in other words, for all the solutions, permute the bits according to a rule) and/or a reversion of the ranking of solutions (in other words, the optimal solution in the first ranking is the worst solution in the second ranking, the second best solution in the first ranking is the second worst solution in the second ranking, and so on).

**Table 3** First 4 non-symmetric rankings with the largest hypervolume values generated by the UBQP for $n = 3$.

| Ranking | Hypervolume |
|---|---|
| $f(111) > f(101) > f(110) > f(100) > f(001) > f(011) > f(000) > f(010)$ | $0.0013237847 \sim 32a$ |
| $f(111) > f(011) > f(110) > f(101) > f(010) > f(001) > f(100) > f(000)$ | $0.0012966579 \sim 31a$ |
| $f(011) > f(010) > f(000) > f(001) > f(100) > f(110) > f(111) > f(101)$ | $0.0012152778 \sim 29a$ |
| $f(111) > f(011) > f(101) > f(110) > f(010) > f(001) > f(100) > f(000)$ | $0.0011013455 \sim 27a$ |

*Example 4* The rankings

$$r_f = [111\ 101\ 110\ 100\ 001\ 011\ 000\ 010]^T$$

and

$$r_f = [100\ 000\ 110\ 010\ 001\ 101\ 011\ 111]^T$$

are symmetric rankings because $r'_f$ is $r_f$ after a reversion and the permutation of the bits (2 3 1) (explicitly, $x_1x_2x_3 \rightarrow x_2x_3x_1$):

$$\begin{bmatrix} 111 \\ 101 \\ 110 \\ 100 \\ 001 \\ 011 \\ 000 \\ 010 \end{bmatrix} \overset{\text{Reversion}}{\Longrightarrow} \begin{bmatrix} 010 \\ 000 \\ 011 \\ 001 \\ 100 \\ 110 \\ 101 \\ 111 \end{bmatrix} \underset{(2\ 3\ 1)}{\overset{\text{Permutation}}{\Longrightarrow}} \begin{bmatrix} 100 \\ 000 \\ 110 \\ 010 \\ 001 \\ 101 \\ 011 \\ 111 \end{bmatrix}.$$

So, when $n = 3$, each ranking has $2 \times n! = 12$ rankings (including itself) which are symmetric and have the same hypervolume value. Therefore, the regions of the hypercube can be grouped in sets of 12 regions and there might be $24192/12 = 2016$ different hypervolume values. In Table 3, the first 4 non-symmetric rankings with the largest and different hypervolume values are shown.

Even if the obtained largest hypervolume values are very small, let us take into account that if all the rankings had the same probability to be sampled (or equivalently, the assumption in question in [3] was true), the values would be $a = 1/24192 = 0.0000413770$. Consequently, there is a significant difference of the values. The 4 groups of rankings of Table 3 (the 48 symmetric rankings) represent almost 6% of the hypercube (in the case that the regions were uniform, the 4 groups of rankings would represent almost 0.2% of the hypercube).

## 4 Experimental analysis of the rankings of the NPP

In this section, we have conducted several experiments on the rankings generated by the NPP. First, we observe how many rankings are generated by the problem sampling integer numbers uniformly at random. Then, we study those rankings and analyze their features.

Because the NPP can be reformulated as a particular case of the UBQP, it is already known that the problem cannot generate all the rankings of solutions by sampling coefficients of the NPP uniformly at random (for any integer value $n$). Additionally, before starting with the experiments, it is necessary to elaborate "injective NPP instances" and local optimal solutions. By definition of the NPP, two opposite solutions have the same fitness function value: $f(x_1 \dots x_n) = f((1-x_1)\dots(1-x_n))$, for any solution $x_1 \dots x_n$. Hence, two assumptions have been considered: (i) we only consider the solutions such that $x_1 = 1$ for the rankings generated by the NPP, and (ii) we study instances of the NPP which are injective (that is to say, for any $Z' \subset Z$, $Z'$ does not have a perfect partition). On the other hand, when we consider local optimal solutions (regarding the Hamming distance), we must consider the opposite solutions (the ones such that $x_1 = 0$) to make realistic conclusions. For example, the solutions 1100 and 1011 are neighbors because the Hamming distance between 1100 and 0100 (which is the opposite solution of 1011) is 1.

The experiments conducted to study the NPP are done for the cases $n \in \{3, 4, 5\}$. For the cases $n \in \{3, 4\}$, similar to the initial step of Section 3, first we have exhaustively counted how many rankings of solutions can be generated by instances of the NPP. To do so, we have analyzed whether or not a ranking of solutions generates a consistent system of inequalities (regarding the definition of the NPP, in which the number of coefficients is $n$). When $n = 3$, there are 4 solutions such that $x_1 = 1$ (which implies that the number of possible rankings is $4! = 24$) and the number of rankings generated by the NPP is 6. When $n = 4$, the number of rankings generated by the NPP is 168 (out of the $8! = 40320$ possible rankings). For the case $n = 5$, a different avenue has been followed because we have not been able to calculate in advance the exact number of different rankings that can be generated by the NPP.

### 4.1 Cases $n \in \{3, 4\}$

For $n \in \{3, 4\}$, to generate an instance, we sample $n$ integer values from the set $\{1, \dots, 2^k\}$ uniformly at random[2], where $k \geq n$. We have tested the results for several $k$ values and sample sizes, obtaining similar results. Because of that, we will only show the results for a sample whose size is 1M and $k = n + 2$. From all the sampled instances, we have only considered the injective rankings.

When $n = 3$, we have obtained the 6 rankings of solutions, and they follow a symmetry: from one ranking, the rest of rankings are obtained by permuting the bits. The main difference among the different samples generated for $n = 3$ is that when the value of $k$ increases, the number of non-injective instances is reduced. Notwithstanding the value of $k$, the 6 rankings are generated uni-

[2] Throughout this work, the function considered to generate random integers is *RandomInteger*, defined in the software system *Wolfram Mathematica*.

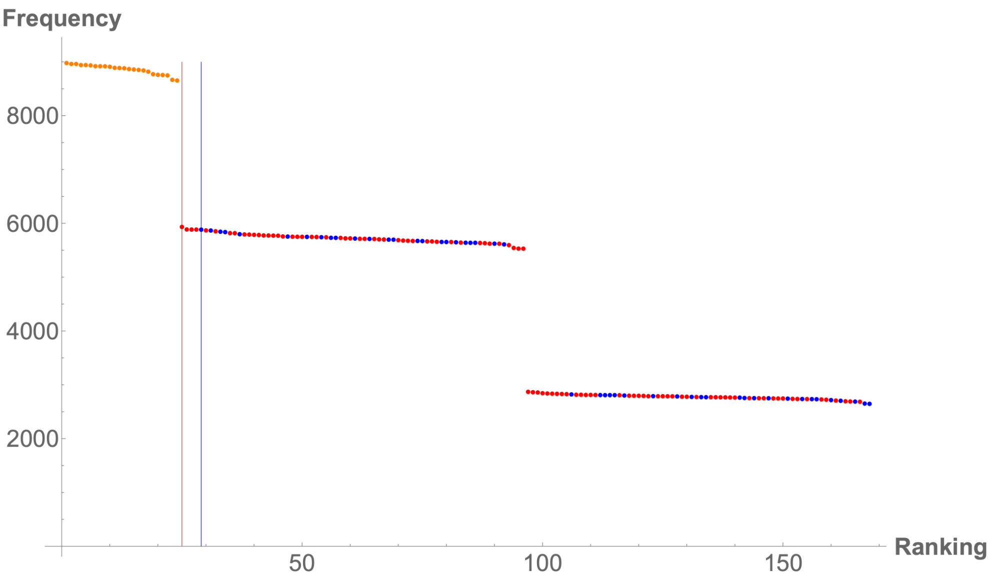


**Fig. 2** Frequency of the 168 rankings of the NPP ($n = 4$) generated in a 1M size sample, in descending order of frequency. Considering the Hamming distance 1, the red points represent the rankings with one local optimal solution (the global optimum). The blue points represent the rankings with two local optimal solutions. Finally, the orange points represent the rankings with three local optimal solutions. The colored vertical lines indicate the most frequent ranking with one and two optimal solutions.

**Table 4** Number of the rankings generated by the NPP for $n = 4$. Each row groups the rankings according to the number of local optimal solutions. In the second and third columns, the number and percentage of different rankings generated by the NPP are shown. In the fourth and fifth columns, the number of sampled injective rankings are shown (from the 1M size sample). In the last column, the 95% confidence interval (CI) of the number of sampled rankings is shown.

| | Rankings (168) | | Sampled rankings (1000000) | | |
|---|---|---|---|---|---|
| L. Opt. Sol. | # | % | # | % | 95% CI |
| 1 | 96 | 57.14 | 408296 | 49.54 | (49.43,49.65) |
| 2 | 48 | 28.57 | 203309 | 24.67 | (24.58,24.76) |
| 3 | 24 | 14.29 | 212613 | 25.8 | (25.90,25.70) |

formly. Consequently, in this particular case, sampling integers uniformly at random is equivalent to sampling NPP instances uniformly at random.

Nevertheless, when $n = 4$, we have obtained the 168 rankings of solutions. If we consider the symmetry of the rankings obtained by the permutation of the bits (from one ranking, we obtain $4! = 24$ symmetric rankings), there are 7 non-symmetric rankings. In Figure 2, we have ordered the 168 rankings according to the number of times that each ranking has been generated in the sample, and the colors represent the number of local optimal solutions of each ranking. In Table 4, the number of possible rankings and sampled rankings are shown.

It is obvious that, even if the NPP can be reformulated as a particular case of the UBQP, the obtained results for the NPP (Table 4) differs significantly from the results of the UBQP (Table 2). For the NPP, 96 (out of 168) rankings have one local optimal solution, 48 have two local optimal solutions and 24 have three local optimal solutions. Nevertheless, in the generated sample, nearly half of the sample is composed of rankings with one local optimal solution (49.54%), a quarter of rankings with two local optimal solutions (24.67&) and the remaining quarter of rankings with three local optimal solutions (25.8%).

To conclude the experiments of the case $n = 4$, we study the rankings generated by the NPP. A meaningful characteristic of our sample is that we identify 3 group of rankings according to the number of times that each ranking has been sampled. There are 24 rankings that each ranking has been sampled more than 8600 times; there are 72 rankings that each ranking has been sampled between 5500 and 6000 times; and the last 72 rankings have been sampled less than 2900 each. This is similar to the grouping that appears in the work [3], where the authors group the instances generated by the LOP in four classes labeled as "S rankings", "M rankings", "L rankings" and "XL rankings".

### 4.2 Case $n = 5$

First, to generate each ranking, we sample 5 integer values from the set $\{1, \ldots, 20000\}$ uniformly at random. Due to the symmetry of the rankings obtained by the permutation of bits, the total number of rankings generated by the NPP and the number of rankings generated by the NPP with $l$ local optimal solutions must be divisible by $5! = 120$. Therefore, we have stopped increasing the sample when the number of different rankings and the number of different rankings with $l$ local optimal solutions (generated by the NPP) were divisible by 120. This scenario has been obtained with a sample of 5 million rankings, whose number of different rankings is 32760 (273 non-symmetric rankings). From all the sampled instances, we have only considered the injective rankings.

In Figure 3, we have ordered the 32760 rankings according to the number of times that each ranking has been generated and where the colors represent the number of local optimal solutions of each ranking. We can observe a similarity between the shapes of Figure 1 and Figure 3, but the local optimal distribution is completely different. The most frequent rankings have three local optimal solutions. Moreover, in the ordered list of rankings with respect to their occurrence in the sample, the most frequent ranking with one local optimal solution is placed in the 590th position (its frequency in the sample is 824); the most frequent ranking with two local optimal solutions is placed in the 587th position (its frequency in the sample is 827); and, lastly, the most frequent ranking with four local optimal solutions is placed in the 1207th position (its frequency in the sample is 577). In addition, in Table 5, a summary of the number of rankings generated in the sample are shown. From all the different rankings generated by the NPP, 4800, 8160, 16800 and 3000

**Table 5** Number of the rankings generated by the NPP for $n = 5$. Each row groups the rankings according to the number of local optimal solutions. In the second and third columns, the number and percentage of different rankings generated by the NPP are shown. In the fourth and fifth columns, the number of sampled injective rankings are shown (from the 5M size sample). In the last column, the 95% confidence interval (CI) of the number of sampled rankings is shown.

| | Rankings (32760) | | Sampled rankings (5000000) | | |
|---|---|---|---|---|---|
| L. Opt. Sol. | # | % | # | % | 95% CI |
| 1 | 4800 | 14.65 | 665803 | 13.34 | (13.31,13.37) |
| 2 | 8160 | 24.91 | 914194 | 18.32 | (18.29,18.35) |
| 3 | 16800 | 51.28 | 2965922 | 59.43 | (59.39,59.48) |
| 4 | 3000 | 9.16 | 444431 | 8.91 | (8.88,8.93) |

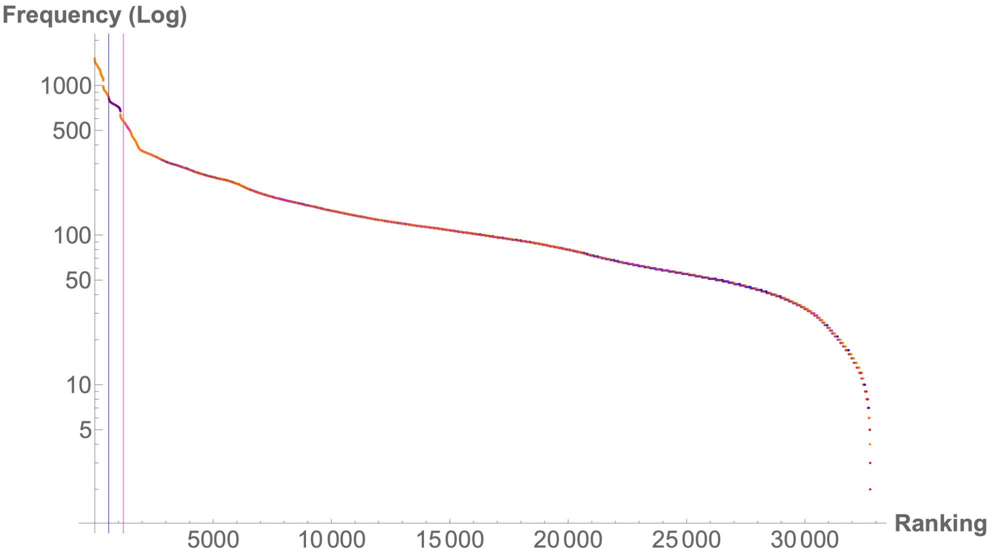


**Fig. 3** Frequency of the 32760 rankings of the NPP ($n = 5$) generated in a 5M size sample, in descending order of frequency (Y axis in logarithmic scale). Considering the Hamming distance 1, the red points represent the rankings with one local optimal solution (the global optimum). The blue points represent the rankings with two local optimal solutions. The orange points represent the rankings with three local optimal solutions. Finally, the light magenta points represent the rankings with four local optimal solutions. The colored vertical lines indicate the most frequent ranking with one, two and four optimal solutions.

rankings have one, two, three and four local optimal solutions, respectively. On the other hand, the number of rankings generated in the sample with one, two, three and four local optimal solutions are 13.34%, 18.32%, 59.43% and 8.91%, respectively. If we compare the obtained results with the case $n = 4$ (Table 4), the most intriguing result is that the proportion of the number of different rankings and the rankings generated in the 5M sample (columns 3 and 5 of Table 5) seem more similar, although the differences are statistically significative (with respect to Pearson's chi-squared test).

## 5 Conclusions

In this paper, we present experimental analyses of the rankings generated by the UBQP for $n = 3$ and the NPP for $n \in \{3, 4, 5\}$. It is confirmed that sampling coefficients uniformly at random to generate instances of the problem does not always generate instances of the problem uniformly at random, at least in terms of the rankings generated. For example, whereas in the case of the NPP for $n = 3$ we have generated instances of the problem uniformly, in the cases of the UBQP for $n = 3$ and the NPP for $n \in \{4, 5\}$, the generated samples are biased. Furthermore, we present an analysis of the generated samples of rankings of solutions, studying several properties of them, such as the number of local optimal solutions and the probability of the occurrence.

There are several possible future works. Firstly, it would be interesting to know if there exists a different avenue to observe whether a specific problem can or cannot generate a ranking of solutions, without solving the system of inequalities described by the problem instance. Secondly, knowing that the NPP can be described as a particular case of the UBQP, it remains to explain the differences between the results of both problems (according to the sampling regions and/or number of local optimal solutions). In addition, it would be ideal to extract more features about the rankings of solutions or to find a different grouping of the instances (not only the symmetries presented in this work) in order to solve them efficiently. Finally, the cases $n > 3$ for the UBQP and $n > 5$ for the NPP need to be explored. On the other hand, it would be interesting to present similar analyses for other COPs.

## Acknowledgments

This research has been partially supported by Spanish Ministry of Science and Innovation through the grants PID2019-104966GB-I00, PID2019-104933GB-I00 and PID2019-106453GA-I00 funded by MCIN/AEI/10.13039/501100011033 and BCAM Severo Ochoa accreditation CEX2021-001142-S; and by the Basque Government through the program BERC 2022-2025 and the projects IT1504-22 and IT1494-22; and by UPV/EHU through the project GIU20/054. Imanol holds a grant from the Department of Education of the Basque Government (PRE_2021_2_0224).

## References

1. K. Smith-Miles and L. Lopes, "Measuring instance difficulty for combinatorial optimization problems," *Computers & Operations Research*, vol. 39, no. 5, pp. 875–889, 2012.
2. K. Smith-Miles, D. Baatar, B. Wreford, and R. Lewis, "Towards objective measures of algorithm performance across instance space," *Computers & Operations Research*, vol. 45, pp. 12–24, 2014.
3. J. Ceberio, A. Mendiburu, and J. A. Lozano, "Are we generating instances uniformly at random?" in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 1645–1651.

4. L. Hernando, A. Mendiburu, and J. A. Lozano, “Characterising the rankings produced by combinatorial optimisation problems and finding their intersections,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 266–273.
5. G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, “The unconstrained binary quadratic programming problem: A survey,” *Journal of Combinatorial Optimization*, vol. 28, no. 1, pp. 58–81, 2014.
6. E. Boros and P. L. Hammer, “Pseudo-boolean optimization,” *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 155–225, 2002.
7. S. E. Venegas-Andraca, W. Cruz-Santos, C. McGeoch, and M. Lanzagorta, “A cross-disciplinary introduction to quantum annealing-based algorithms,” *Contemporary Physics*, vol. 59, no. 2, pp. 174–197, 2018.
8. M. Jünger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, and T. Stollenwerk, “Performance of a quantum annealer for ising ground state computations on chimera graphs,” *arXiv preprint arXiv:1904.11965*, 2019.
9. P. Merz and K. Katayama, “Memetic algorithms for the unconstrained binary quadratic programming problem,” *BioSystems*, vol. 78, no. 1-3, pp. 99–118, 2004.
10. F. Mahdavi Pajouh, B. Balasundaram, and O. A. Prokopyev, “On characterization of maximal independent sets via quadratic optimization,” *Journal of Heuristics*, vol. 19, no. 4, pp. 629–644, 2013.
11. A. Liefooghe, S. Verel, and J.-K. Hao, “A hybrid metaheuristic for multiobjective unconstrained binary quadratic programming,” *Applied Soft Computing*, vol. 16, pp. 10–19, 2014.
12. F. Chicano, L. D. Whitley, and E. Alba, “A methodology to find the elementary landscape decomposition of combinatorial optimization problems,” *Evolutionary Computation*, vol. 19, no. 4, pp. 597–637, 2011.
13. P. F. Stadler, W. Hordijk, and J. F. Fontanari, “Phase transition and landscape statistics of the number partitioning problem,” *Physical Review E*, vol. 67, no. 5, p. 056701, 2003.
14. S. Mertens, “Phase transition in the number partitioning problem,” *Physical Review Letters*, vol. 81, no. 20, p. 4281, 1998.