PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

BINARY CAT SWARM OPTIMIZATION TO SOLVE THE SET COVERING PROBLEM

NATALIA ANGELINA BERRÍOS PEÑA

Profesor Guía : Broderick Crawford Labrín
Co-referente : Ricardo Soto de Giorgis

INFORME FINAL PROYECTO DE TÍTULO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERÍA CIVIL EN INFORMÁTICA

NOVEMBER, 2016

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

BINARY CAT SWARM OPTIMIZATION TO SOLVE THE SET COVERING PROBLEM

NATALIA ANGELINA BERRÍOS PEÑA

Profesor Guía : **Broderick Crawford Labrín**
Co-referente : **Ricardo Soto de Giorgis**

Carrera: **Ingeniería Civil en Informática**

NOVEMBER, 2016

Acknowledgement

I want to thank my parents for trust and support me throughout this process. I thank my brothers and sister for being always united, who gave me the strength to continue at each stage of the career. I thank the rest of my family Who always stood and giving me a little bit of breath in this road. I thank my colleagues and friends with whom we lived this whole process and shared difficult and happy moments. I also thank my thesis adviser for his help and trust in my abilities. Finally, I thank all the teachers who were part of my professional and personal improvement, thank you very much to everyone.

Natalia Berríos Peña.

Abstract

In this work, we present a binary cat swarm optimization for solving the set covering problem. The Cat Swarm Optimization is a recent swarm metaheuristic technique based on the behavior of cats. Domestic cats show the ability to hunt and are curious about objects in motion. Based on this, the cats have two modes of behavior: seeking mode and tracing mode. Moreover, eight different transfer functions and five discretization techniques are considered for solving the set covering problem. Finally, we illustrate this approach with 65 instances of the problem, we make a comparison between the different binarization techniques and we choose the best of them through Relative Percentage Deviation and Wilcoxon-Mann-Whitney's.

Keywords: the set covering problem, metaheuristic, binary cat swarm optimization.

Resumen

En este trabajo se presenta el Binary Cat Swarm Optimization para resolver el Set Covering Problem. Cat Swarm Optimization es una metaheuristica reciente, técnica basada en el comportamiento de los gatos. Los gatos domésticos muestran su habilidad para cazar y su curiosidad por los objetos en movimiento. Basado en esto, los gatos tienen dos comportamientos: modo de búsqueda y modo de rastreo. Además, ocho funciones de transferencia y cinco técnicas de discretización son utilizadas para resolver el problema binario. Finalmente, se ilustra este enfoque con 65 instancias del problema, se hace una comparación entre las diferentes técnicas de binarización y se escoge la mejor de ellas a través de la Desviación Porcentual Relativa y Wilcoxon-Mann-Whitney's.

Palabras Clave: problema de cobertura de conjuntos, metaheurística, optimización por colonia de gatos binaria.

Contents

List of Figures

List of Tables

1 Glossary

- C : Population size, that is, number of cats
- CDC : Counts of Dimensions to Change, it indicates how many of the dimensions varied
- PMO : Probability of Mutation Operation.
- SMP : Seeking Memory Pool, it is used to define the size of seeking memory for each cat
- MR : Mixture ratio, percentage of cats that undertake the tracing mode.
- c_1 : Constant that is defined by the user
- r_1 : Random value in the interval of [0,1]
- t_{kd} : Probability of mutation in each dimension of the cat_k
- V_{kd}^1 : Probability of d-dimension of cat_k changes to one
- V_{kd}^0 : Probability of d-dimension of cat_k changes to zero
- w : Inertia weight
- $X_{best,d}$: d-dimension of best cat

2 Introduction

The Set Covering Problem (SCP) is one of the classical 21 problems shown to be NP-complete [58] and whose optimization version is NP-hard [52]. Although the SCP is a traditional optimization problem, it is widely considered in the current literature for designing expert systems, which emulate the decision-making ability of human experts in a given field [67]. For example, we find works considering the SCP crew scheduling [4] [8] [46] [70], location of emergency facilities [85] [81], production planning in industry [82] [83] [84], vehicle routing [7] [50], ship scheduling [49] [13], network attack or defense [14], assembly line balancing [51] [72], traffic assignment in satellite communication systems [69] [17], simplifying boolean expressions [15], the calculation of bounds in integer programs [18], information retrieval [45], political districting [53], stock cutting, crew scheduling problems in airlines [56] and other important real life situations.

The SCP consists in finding a set of solutions which allow to cover a set of needs at the lowest cost possible. Chvatal in [21] defined the SCP as follows: given a set M of m objects and a collection S of n sets of these objects, each set with a non-negative cost associated. The goals is to find a minimum cost family of subsets $C \subset S$, such that each element $i \in M$ belongs to at least one subset of the family C.

Some authors solved the SCP by applying exact techniques, such as branch-and-bound and branch-and-cut algorithms. However, such methods are not recommended for solving this type of complex problems, because computational times rise exponentially with the problem dimension. In the field of optimization, approximate techniques should be considered, such as metaheuristics. This type of techniques is successfully considered in the literature for solving NP-hard problems from different fields, including the SCP. However, many algorithms have been developed for solving the SCP. Examples of these optimization algorithms include: Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). In this proposal, we consider swarm intelligence based on the behavior of cats: Binary Cat Swarm Optimization (BCSO).

BCSO refers to a series of heuristic optimization methods and algorithms based on cat behavior in nature. Cats behave in two ways: seeking mode and tracing mode. BCSO is based in CSO algorithm, proposed by Chu and Tsai recently [19]. The difference is that in BCSO the vector position consists of ones and zeros, instead the real numbers of CSO.

Usually, the algorithms are adapted by following the two-step binarization method [59] in their approach of Binary Particle Swarm Optimization

(BPSO) for transforming real numbers into binary ones. In this case, the authors explained how to get a new binary solution according to the particle velocity, which is a real number. The method followed by the authors is as follows. Firstly, we map the real value to a number in the interval [0,1] through a transfer function. Secondly, we transform the number in the interval [0,1] into a binary value through a discretization function. In this line, there are eight major transfer functions and five major discretization functions in the current literature, denoted as $S_1, S_2, \ldots, S_4, V_1, V_2, \ldots, V_4$ and $D_1, D_2 \ldots, D_8$, respectively.

The authors of BCSO considered a transfer and discretization function, without performing any formal study to this task. We change the original formulation of BCSO by combining the eight transfer functions and the five discretization functions introduced before, i.e., we get forty BCSO approaches. We apply the forty BCSO approaches for solving a freely available SCP benchmark proposed by Beasley. We study the results obtained through an accepted statistical methodology to analyze if selecting a binarization technique influences the behavior of the metaheuristic.

The remainder of this work is structured as follows. In Section 3, we include the definition of objectives. In Section 4, we discuss the state of the art, including the major motivations for performing this work. In Section 5, we give a formal SCP definition, including a problem example. In Section 6, we explain the BCSO metaheuristic. In Section 7, we describe how the problem is solved and we explain the transfer and discretization functions. In Section 8, we discuss the experimental method followed. In Section 9, we give some implementation details and the results obtained. Finally, conclusions are left for Section 10.

3 Definition of Objectives

3.1 Main Objective

The resolution of the Set Covering Problem using a Binary Cat Swarm Optimization.

3.2 Specific Objective

- Understand the Set Covering Problem.
- Understand the metaheuristic Binary Cat Swarm Optimization.
- Implement the Binary Cat Swarm Optimization to solve the Set Covering Problem.
- Change the binarization technique usually proposed for this algorithm in order to discover if a different one could help to improve results.
- Use eight transfer functions.
- Use five discretization techniques.
- Test the algorithm using the OR-Library problems.
- Analyze the results with a statistical methodology.
- Improve the results if necessary.
- Publish results.

4 State of the Art

4.1 The Set Covering Problem

Different solving methods have been proposed in the literature for the Set Covering Problem. Exact algorithms are mostly based on branch-and-bound and branch-and-cut techniques [6] [48] [12], linear programing, and heuristic methods [16]. However, these algorithms are rather time consuming and can only solve instances of very limited size. For this reason, many research efforts have been focused on the development of heuristics to find a good result or near-optimal solutions within a reasonable period of time. Metaheuristics were also applied to the SCP as top-level general search strategies.

First we described some of the techniques developed by Crawford and Soto *et al.* Of the techniques we can mention the latest on Shuffled Frog Leaping Algorithm (SFLA) [37] designed in 2006 by Eusuff *et al.* [47]. The SFLA is a novel metaheuristic inspired by natural memetics, in which the authors, Crawford and Soto *et al.*, assumed all the transfer and discretization function.

Another metaheuristic used by Crawford and Soto *et al.* is Artificial Bee Colony (ABC) Algorithm [43] [29] [28], published in 2014. The ABC algorithm is a recent metaheuristic technique based on the intelligent foraging behavior of honey bee swarm. ABC was proposed by Karaboga & Basturk in 2007 [57].

Firefly Optimization is another proposal to solve the SCP by Crawford and Soto *et al.* [35] [33] [34]. The Firefly Algorithm (FA) designed by Yang in 2010 [89]. This proposal is inspired by the flashing behaviour of fireflies. In their works, they changed the original proposal by incorporing transfer and discretization functions. Thus, in citecrawford2015modified, Crawford *et al.* assumed all the transfer and discretization functions; and in [32] they applied the eight transfer functions and the discretization functions D_2, D_3, and D_4.

In 2015, Crawford *et al.* [38] assumed the Teaching-Learning-Based Optimization (TLBO) algorithm proposed by Rao *et al.* in 2011 [66]. Crawford *et al.* applied all the transfer and discretization functions in this paper.

Frui Fly Optimization Algorithm (FFOA) [41] was applied by Crawford *et al.* in 2015, algorithm designed by Pan in 2012 [64]. The authors assumed all the transfer and discretization functions.

Furthermore, the authors propose a novel idea called a 2-level metaheuristic approach where an Artificial Bee Colony Algorithm acts as a low-level metaheuristic and its paremeters are set by a higher level Genetic Algorithm.

[80] [36].

Other important proposals by the same authors are: Cultural Algorithms [22] [30], a Evolutionary Approach [23], Hybrid Ant Algorithm [31] and Two Swarm Intelligence [27].

Finally, other authors have also made proposals to solve the SCP. We found Beasley [10] [9] [11] [12]. In one of his most important works, "A genetic algorithm for the set covering problem", he presented a genetic algorithm-based heuristic for non-unicost SCP, where proposed several modifications to the basic genetic procedures including a new fitness-based crossover operator (fusion), a variable mutation rate and a heuristic feasibility operator tailored specifically for the set covering problem. Also, we find the following proposal list that have solved the SCP: Genetic Algorithm [3], Evolutionary Search Technique [54], Ant Colony Optimization[60][68]. Standing out the latter for proposing "New ideas for applying ant colony optimization to the set covering problem".

4.2 Cat Swarm Optimization

In the case of the metaheuristic used in this work, BCSO. This has been applied to a lot of problems [20] [76] [19].

Shojaee *et al.* presents a new swarm intelligence technique based on CSO algorithm to find near optimal solution. In this paper "A new cat swarm optimization based algorithm for reliability-oriented task allocation in distributed systems [77]" the nodes and links of a Distributed Systems (DS) typically have different hazard rates; therefore, proper task allocation can significantly improve system reliability. On the other hand, optimal task allocation in DSs is an NP-hard problem, thus finding exact solutions are limited to small-scale problems.

Saha *et al.* have a paper "Optimizing least-significant-bit substitution using cat swarm optimization strategy" [86] to adopt the cat swarm optimization (CSO) strategy to obtain the optimal or near optimal solution of the stego-image quality problem. Embedding secret data into a cover image using simple least-significant-bit substitution can degrade the image quality dramatically. The exhaustive least-significant-bit substitution method is proposed to solve this problem.

Saha *et al.* in "Cat Swarm Optimization algorithm for optimal linear phase FIR filter design" [71]. CSO algorithm is applied to determine the best optimal impulse response coefficients of FIR low pass, high pass, band pass and band stop filters, trying to meet the respective ideal frequency response characteristics.

5 The Set Covering Problem

The SCP is formally defined by assuming a binary matrix A of m-rows and n-columns, where $a_{i,j} \in$ 0,1 denotes the value of the cell (i, j) of A, with $i \in$ 1,2,...,m and $j \in$ 1,2,...,n. A is formally defined as

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}, \tag{1}$$

We say that a column j covers a row i if $a_{i,j}$ equals 1 and 0 otherwise. Each column j is associated with a non-negative real cost $c_j \in C$, where $C = \{c_1, c_2, \dots, c_n\}$. Let $I = \{1, 2, \dots, m\}$ and $J = \{1, 2, \dots, n\}$ be the row and columns sets, respectively. The SCP calls for a minimum cost subset $S \subseteq J$, such that each row $i \in I$ is covered by at least one column $j \in S$. Thus, the optimization problem is expressed as

$$min \quad \sum_{j \in J} c_j x_j \ , \tag{2}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq 1, \quad \forall i \in I \ , \tag{3}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in J \ , \tag{4}$$

where x_j equals 1 if column j is in the solution S and 0 otherwise. From this formulation, we reach that the goal is to minimize the sum of the costs of the selected columnns. Note that the constraint in Equation (3) ensures that each row i is covered by at least one column.

5.1 Problem Example

We propose a small location problem as an example of the SCP formulated before. Suppose that we need to offer fire services at the lowest possible cost in the city composed of six zones shown in Figure 1. In this case, we have the following constraints:

- A fire station can only attend the zone in which it is located and inmediately adjacent zones, e.g., a fire station in zone 1 can attend zones 1, 2, 3, and 5.
- The fire services should attend all the zones.
- The greatest number of fire stations per zone is 1.

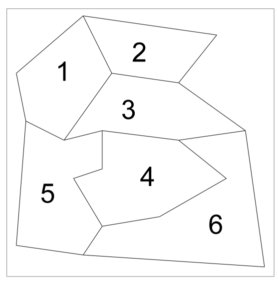

Figure 1: Zones for the location problem example.

Let A be the binary matrix denoting which zones are covered by a hypothetical fire station according to the zone in which it is placed, that is

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{5}$$

where $a_{i,j}$ is the value at the cell (i, j) of A, equaling 1 if the fire station located at the j-th zone covers the i-th zone, with i and $j \in 1, 2, \ldots, 6$. Let x_j be the indicator function equaling 1 if a fire station is built in the j-th zone and 0 otherwise. Let C be the set denoting the cost of building a fire station according to the zone, that is

$$C = \left(\begin{array}{cccccc} 3, & 5, & 6, & 4, & 2, & 1 \end{array}\right), \tag{6}$$

According to this notation, the optimization problem is defined as

$$min \quad 3\ x_1 + 5\ x_2 + 6\ x_3 + 4\ x_4 + 2\ x_5 + 1\ x_6\ \ , \tag{7}$$

subject to

$$\begin{array}{l} a_{1,1}\ x_1 + a_{1,2}\ x_2 + a_{1,3}\ x_3 + a_{1,4}\ x_4 + a_{1,5}\ x_5 + a_{1,6}\ x_6 \geq 1 \\ a_{2,1}\ x_1 + a_{2,2}\ x_2 + a_{2,3}\ x_3 + a_{2,4}\ x_4 + a_{2,5}\ x_5 + a_{2,6}\ x_6 \geq 1 \\ a_{3,1}\ x_1 + a_{3,2}\ x_2 + a_{3,3}\ x_3 + a_{3,4}\ x_4 + a_{3,5}\ x_5 + a_{3,6}\ x_6 \geq 1 \\ a_{4,1}\ x_1 + a_{4,2}\ x_2 + a_{4,3}\ x_3 + a_{4,4}\ x_4 + a_{4,5}\ x_5 + a_{4,6}\ x_6 \geq 1 \\ a_{5,1}\ x_1 + a_{5,2}\ x_2 + a_{5,3}\ x_3 + a_{5,4}\ x_4 + a_{5,5}\ x_5 + a_{5,6}\ x_6 \geq 1 \\ a_{6,1}\ x_1 + a_{6,2}\ x_2 + a_{6,3}\ x_3 + a_{6,4}\ x_4 + a_{6,5}\ x_5 + a_{6,6}\ x_6 \geq 1 \end{array} \quad , \tag{8}$$

$$x_j \in \{0,1\}, \quad \forall j \in 1,\ldots,6\ \ . \tag{9}$$

Equation (8) is simplifiable by replacing the values of $a_{i,j}$ in Equation (5) based on the notation in Equation (1), that is

$$\begin{array}{l} x_1 + x_2 + x_3 + x_5 \geq 1 \\ x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \\ x_3 + x_4 + x_5 + x_6 \geq 1 \\ x_1 + x_3 + x_4 + x_5 + x_6 \geq 1 \\ x_3 + x_4 + x_5 + x_6 \geq 1 \end{array} \quad , \tag{10}$$

The optimal solution to solve this problem is to build a fire station in zones 1 and 6, having a total cost of 4.

6 Binary Cat Swarm Optimization

There are about thirty different species of known felines, e.g., lion, tiger, leopard, and cat, among others [5]. Although they have different living environments, cats share similar behavior patterns [63].

For wild cats, the hunting skill ensures food supply and survival of the species [44]. Feral cats are groups with a mission to hunt their food, are very wild feline colonies, ranging from 2-15 individuals [88].

Domestic cats also show the same ability to hunt, and are curious about moving objects [42] [78] [87]. Analyzing the behavior of cats, you could think that most of the time they are resting, even when awake [1] [2]. In this state of alertness they do not never leave, they may be listening or with wide eyes to look around [73]. Based on all these behaviors we formulate BCSO.

BCSO [76] is an optimization algorithm that imitates the natural behavior of cats [19] [79] [65]. The authors identified two main modes of behavior for simulating cats:

- Seeking mode: cats are attracted by objects in motion and have a great hunting ability. It might be thought that cats spend most of the time resting, but in fact they are constantly alert and moving slowly.
- Tracing mode: when cats detect a prey, they spend lots of energy because of their fast movements.

In BCSO these two behaviors are modeled mathematically to solve complex optimization problems.

In BCSO, the first decision is the number of cats needed for each iteration. Each cat, represented by cat_k, where $k \in [1, C]$, has its own position consisting of M dimensions, which are composed by ones and zeros. Besides, they have speed for each dimension d, a flag for indicating if the cat is on seeking mode or tracing mode and finally a fitness value that is calculated based on the SCP. The BCSO keeps searching the best solution until the end of iterations.

In BCSO the bits of the cat positions are $x_j = 1$ if column j is in the solution and 0 otherwise (Equation 2). Cat position represents the solution of the SCP and the constraint matrix ensure that each row i is covered by at least one column.

Next we describe the BCSO general pseudocode (Algorithm 1) in which MR is a percentage which determines the number of cats that undertake the seeking mode.

Algorithm 1 $BCSO()$

1: Create C cats;
2: Initialize the cat positions randomly with values between 1 and 0;
3: Initialize velocities and flag of every cat;
4: Set the cats into seeking mode according to MR, and the others set into tracing mode;
5: Evaluate the cats according to the fitness function;
6: Keep the best cat which has the best fitness value into memory;
7: Move the cats according to their flags, if cat_k is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process. The process steps are presented above;
8: Re-pick number of cats and set them into tracing mode according to MR, then set the other cats into seeking mode;
9: Check the termination condition, if satisfied, terminate the program, and otherwise repeat since step 5;

Next we describe the two sub-models of the BCSO: seeking mode and tracing mode. Each sub-models contains a brief explanation of their behavior and includes a detailed pseudocode and a figure with the flowchart of behavior.

6.1 Seeking mode

This sub-model is considered for modeling the state of the cat, which is resting, looking around and seeking the next position to move to. Seeking mode has essential factors:

- PMO: Probability of Mutation Operation
- CDC: Counts of Dimensions to Change, it indicates how the dimensions vary
- SMP: Seeking Memory Pool, it is used to define the size of seeking memory for each cat. SMP indicates the points explored by the cat, this parameter can be different for different cats.

The following pseudocode describes cat behavior seeking mode. In which FS_i is the fitness of i-th cat and $FS_b = FS_{max}$ for finding the minimum solution and $FS_b = FS_{min}$ for finding the maximum solution. To solve the SCP we use $FS_b = FS_{max}$.

Step1: Create SMP copies of cat_k

Step2: Based on CDC update the position of each copy by randomly according to PMO

Step3: Evaluate the fitness of all copies

Step4: Calculate the selecting probability of each copy according to

$$P_i = \left| \frac{FS_i - FS_b}{FS_{max} - FS_{min}} \right| \ , \tag{11}$$

Step5: Apply roulette wheel to the candidate points and select one of them

Step6: replace the current position with the selected candidate

6.2 Tracing mode

Tracing mode is the sub-model for modeling the case of the cat in tracing targets. In the tracing mode, cats are moving towards the best target. Once a cat goes into tracing mode, it moves according to its own velocities for each dimension. Every cat has two velocity vectors defined as V^1_{kd} and V^0_{kd}. Where V^0_{kd} is the probability that bits of the cat change to zero and V^1_{kd} is the probability the bits of cat change to one. The velocity vector changes its meaning to the probability of mutation in each dimension of a cat. The tracing mode action is described in the next pseudocode and diagram.

Step1: Calculate d^1_{kd} and d^0_{kd} according to the expression, where $X_{best,d}$ is the d-th dimension of the best cat, r_1 has a random values in the interval of [0,1] and c_1 is a constant which is defined by the user

$$\begin{array}{l} \text{if } X_{best,d} = 1 \text{ then } d^1_{kd} = r_1 c_1 \text{ and } d^0_{kd} = -r_1 c_1 \\ \text{if } X_{best,d} = 0 \text{ then } d^1_{kd} = -r_1 c_1 \text{ and } d^0_{kd} = r_1 c_1 \end{array} \tag{12}$$

Step2: Update process of V^1_{kd} and V^0_{kd} are as follows, where w is the inertia weight and M is the column numbers.

$$\begin{array}{l} V^1_{kd} = wV^1_{kd} + d^1_{kd} \\ V^0_{kd} = wV^0_{kd} + d^0_{kd} \end{array} \qquad \text{d = 1,...,M} \tag{13}$$

Step3: Calculate the velocity of cat_k, V'_{kd}, according to

$$V'_{kd} = \begin{cases} V^1_{kd} \text{ if } X_{kd} = 0 \\ V^0_{kd} \text{ if } X_{kd} = 1 \end{cases} \tag{14}$$

Step4: Calculate the probability of mutation in each dimension, this is defined by parameter t_{kd}, t_{kd} takes a value in the inverval of [0,1]

$$t_{kd} = \frac{1}{1 + e^{-V'_{kd}}} \tag{15}$$

Step5: Based on the value of t_{kd} the new value of each dimension of cat is update as follows

$$X_{kd} = \begin{cases} X_{best,d} \text{ if } rand < t_{kd} \\ X_{kd} \text{ if } t_{kd} < rand \end{cases} \quad \text{d = 1,...,M} \tag{16}$$

The maximum velocity vector of V'_{kd} should be bounded to a value V_{max}. If the value of V'_{kd} becomes larger than V_{max}. V_{max} should be selected for velocity in the corresponding dimension.

7 Solving the Set Covering Problem

For solving the SCP we use the transfer and discretization functions, explained later, to replace those used in Tracing Mode and choose the one that delivers better results. We also use a repair method for solutions that are not feasible. Next is described the pseudocode, Algorithm 2, that we use for solving the SCP:

Algorithm 2 *Solving SCP*()

1: Initialize parameters in cats;
2: Initialization of cat positions, randomly initialize cat positions with values between 0 and 1;
3: Initialization of all parameter of BCSO;
4: Evaluation of the fitness of the population. In this case the fitness function is equal to the objective function of the SCP;
5: Change of the position of the cat. A cat produces a modification in the position based in one of the behaviors. i.e. seeking mode or tracing mode;
6: If solution is not feasible then repaired;
7: Memorizes the best found solution. Increases the number of iterations;
8: Stop the process and show the result if the completion criteria are met. Completion criteria used in this work are the number specified maximum of iterations. Otherwise, go to step 3;

7.1 Improving Operator

Based on the SCP definition discussed in Section 5, it is possible that a solution does not satisfy the constraints, resulting in an infeasible solution. In this section, we describe an improving operator for transforming infeasible solutions into feasible ones and removing redundant columns to reduce the solution cost. Note that a column is redundant if after removing it, the solution remains feasible.

Algorithm 3 shows a repair method where all rows not covered are identified and the columns required are added. So in this way all the constraints will be covered. The search of these columns are based in the relationship showed in the Equation 17.

$$\frac{cost\ of\ one\ column}{amount\ of\ columns\ not\ covered} \tag{17}$$

- I is the set of all rows

- J is the set of all columns
- J_i is the set of columns that cover the row i, $i \in I$
- I_j is the set of rows covered by the column j, $j \in J$
- S is the set of columns of the solution
- U, is the set of columns not covered
- w_i is the number of columns that cover the row $i, \forall i \in I$ in S

Algorithm 3 *Repair Operator*()

$w_i \leftarrow |S \cap J_i| \ \forall i \in I$;
$U \leftarrow \{i | w_i = 0\}, \forall i \in I$;
for $i \in U$ **do**
    find the first column j in J_i that minimize $\frac{c_j}{|U \cap I_j|} S \leftarrow S \cap j$;
    $w_i \leftarrow w_i + 1, \forall i \in I_j$;
    $U \leftarrow U - I_j$;
end for
for $j \in S$ **do**
    if $w_i \geq 2, \forall i \in I_j$ **then**
        $S \leftarrow S - j$;
        $w_i \leftarrow w_i - 1, \forall i \in I_j$
    end if
end for

In the next section we discuss the transfer and discretization functions considered for addressing the step 4 and 5 of the tracing mode described in Section 6.2.

7.2 Transfer Functions

Transfer functions define a probability to mutate an element, t_{kd} in Tracing Mode as given by Equation 15, that is the probability to change an element of the solution from 1 to 0, or vice versa. As stated before, we propose to consider several transfer functions, specifically the eight functions defined by Mirjalili *et al.* in [62], which are in Table 1. These functions are divided in two groups called S-shape and V-shape.

S-Shape	V-Shape
S1 $t_{kd} = \frac{1}{1+e^{-2V'_{kd}}}$	**V1** $t_{kd} = \left\lvert \mathrm{erf}\left(\frac{\sqrt{\pi}}{2}V'_{kd}\right)\right\rvert$
S2 $t_{kd} = \frac{1}{1+e^{-V'_{kd}}}$	**V2** $t_{kd} = \left\lvert tanh(V'_{kd})\right\rvert$
S3 $t_{kd} = \frac{1}{1+e^{\frac{-V'_{kd}}{2}}}$	**V3** $t_{kd} = \left\lvert \frac{V'_{kd}}{\sqrt{1+(V'_{kd})^2}}\right\rvert$
S4 $t_{kd} = \frac{1}{1+e^{\frac{-V'_{kd}}{3}}}$	**V4** $t_{kd} = \left\lvert \frac{2}{\pi}arctan\left(\frac{\pi}{2}V'_{kd}\right)\right\rvert$

Table 1: Transfer Functions [62].

Analyzing, functions having less smoothness have a smaller range of input values V'_{kd} providing non-extreme values of the output interval [0,1] than functions having more smoothness. For example, S_1 is less smoothness than S_2. Based on S_1 formulation, if we consider V'_{kd} equaling 2.3 and -2.3, we get a t_{kd} value of 0.99 and 0.01, respectively. On the other hand and based on S_2 formulation, the same t_{kd} values are obtained by assuming V'_{kd} equaling 4.6 and -4.6, respectively. Thus, the range of input values of S_2 providing non-extreme values is greater than the range of S_1. The motivation of including both groups of transfer functions is as follows. A high velocity implies that the cat is away from the optimal solution and a low velocity implies that the cat is close to the optimal solution. Based on this velocity, the strategy of both groups of transfer function is different. S-shape functions cause that cats with a low velocity have associated a low mutation probability, while cats with a high velocity have associated a high mutation probability. On the contrary, v-shape functions cause that the mutation probability be high for cats with low and high velocities.

7.3 Discretization Techniques

Discretization functions transform real values into binary ones. Such functions are needed because SCP is represented by assuming a binary scope, while the BCSO algorithm considers movements in the set of real numbers.

As stated before, we consider five discretization functions, which were introduced by Crawford *et al.* [40]. Such functions are in Table 2, where $\overline{(x_i^k)}$ is the logical complement of a proposition, α is a random number in the interval [0,1], and x_{best}^k is the value of the j-th cell of the best solution.

Finally, the motivation of including the five discretization functions is because all they offer different capabilities to the search strategy. As a way

$\mathbf{D_1}$: (Roulette)

$$X_{kd}(t+1) = \begin{cases} p[X_{kd}] = \frac{f_i}{\sum_{j=1}^{k} f_j} & \text{if } \alpha \leq t_{kd}(t+1), \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{D_2}$: (Complement)

$$X_{kd}(t+1) = \begin{cases} \overline{(X_{kd})} & \text{if } \alpha \leq t_{kd}(t+1), \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{D_3}$: (Set the best)

$$X_{kd}(t+1) = \begin{cases} X_{best,d} & \text{if } \alpha \leq t_{kd}(t+1), \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{D_4}$: (Standard)

$$X_{kd}(t+1) = \begin{cases} 1 & \text{if } \alpha \leq t_{kd}(t+1), \\ 0 & \text{otherwise} \end{cases}$$

$\mathbf{D_5}$: (Static probability)

$$X_{kd}(t+1) = \begin{cases} X_{kd} & \text{if } t_{kd}(t+1) \leq \alpha, \\ X_{best,d} & \text{if } \alpha \leq t_{kd}(t+1) \leq \frac{1}{2}(1+\alpha), \\ 1 & \text{if } \frac{1}{2}(1+\alpha) \leq t_{kd}(t+1). \end{cases}$$

Table 2: Discretization functions.

of identifying such abilities and based on the concepts of explanation and exploitation, we could define the following indicative orders from best to worst: $D_3, D_1, D_5, D, 2, D4$ according to exploitation ability and $D_4, D_2, D_5, D_1, D3$ according to exploration ability. Note that both rankings are reversed. Crawford *et al.* [40] checked that the behavior of such discretization functions was different. However, they did not perform any formal study of the difference observed.

8 Experimental methodology

We apply the CSO algorithm for solving 65 instances from the known benchmark proposed by Beasley in [10] and shown in Table 3. As stated before, the CSO algorithm is a Swarm Intelligence Algorithm from continuous optimization, which was adapted to discrete optimization. The original authors considered a binarization technique, which assumed the discretization function S_2 and the transference function D_3, both described in Section 7. Now, we solve the instances by assuming 40 different binarization techniques, i.e. five discretization functions and eight transference functions.

Instance Set	Number of instances	m	n	Cost range	Density (%)	Optimal solution
4	10	200	1000	[1,100]	2	Known
5	10	200	2000	[1,100]	2	Known
6	5	200	1000	[1,100]	5	Known
A	5	300	3000	[1,100]	2	Known
B	5	300	3000	[1,100]	5	Known
C	5	400	4000	[1,100]	2	Known
D	5	400	4000	[1,100]	5	Known
NRE	5	500	5000	[1,100]	10	Unknown
NRF	5	500	5000	[1,100]	20	Unknown
NRG	5	1000	10000	[1,100]	2	Unknown
NRH	5	1000	10000	[1,100]	5	Unknown

Table 3: Description of the dataset.

We perform 30 independent runs for each instance and binarization technique, being 30 a widely accepted values for getting statistical conclusions [55]. The stop condition considered is the same for all instances, being based on the number of evaluations. This type of criterion is fairer than others, such as elapsed time, which depends on the machine considered. The maximum number of evaluations assumed is 40000 for the instances having known optimal solutions and 5000 for all other. These values were experimentally obtained and are enough for analyzing the behavior of the algorithms.

As a quality metric, we consider the Relative Percentage Deviation (RPD), quantifying how close a solution from the optimal one is. That is calculated as

$$RPD = \left(\frac{Z_{avg} - Z_{opt}}{Z_{opt}}\right) * 100 \tag{18}$$

where Z_{avg} is the average value from the distribution of 30 samples and Z_{opt} is the optimal solution provided by the benchmark. The distribution of

30 solutions for each instance and binarization technique is analyzed through the widely accepted statistical methodology shown in Figure 2 [55] to conclude if the differences observed are significant.

Before running the experiments, the CSO algorithm was configured. To this end and for each parameter of the algorithm described in Section 6 we define a range of values to study and a default configuration. Then, 10 independent runs are performed for each configuration of the parameter, instance, and binarization technique, resulting 1200 run for each value of parameter. Then, the configuration providing the best performance on average is selected. Next, another parameter is selected so long as all of them are fixed. Table 4 shows for each parameter the range of values considered and the configuration selected.

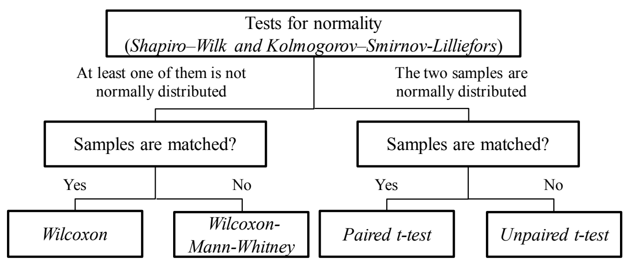

Figure 2: Statistical methodology.

Note that we configure the algorithm splitting the benchmark in five groups according to the complexity of the instances. The idea is to consider the best possible configuration for each case. As shown in Table 4 the configurations are significantly different for each group.

Name	Parameter	Instance Set	Selected	Range
Number of Cats	C	4, 5 and 6	100	[10,20,...,1000]
		A and B	50	
		C and D	30	
		NRE and NRF	25	
		NRG and NRH	20	
Mixture Ratio	MR	4 and 5	0.7	[0.1,0.2,..,0.9]
		A and B	0.65	
		C and D	0.5	
		NRE and NRF	0.5	
		NRG and NRH	0.5	
Seeking Memory Pool	SMP	4 and 5	5	[5,10,...,100]
		A and B	5	
		C and D	10	
		NRE and NRF	15	
		NRG and NRH	20	
Probabily of Mutation Operation	PMO	4 and 5	0,97	[0.10,0.97,...,1.00]
		A and B	0,93	
		C and D	0,9	
		NRE and NRF	1	
		NRG and NRH	1	
Counts of Dimension to Change	CDC	4 and 5	0,001	[0.001,0.01,..,0,9]
		A and B	0,001	
		C and D	0,002	
		NRE and NRF	0,002	
		NRG and NRH	0,01	
Weight	w	All	1	[0.1,0.25,..,5]
Factor c_1	c_1	All	1	[0.1,0.25,..,5]

Table 4: Parametric swap.

9 Results

Tables 6 and 7 show the average RPD for each instance and binarization technique, where lower (better) RPD values are shaded. Note that Table 6 includes s-shape transfer functions and Table 7 includes v-shape ones. In these tables, some binarization techniques seem to offer better performance than others for a given instance. However, we do not know if the differences observed are significant.

To this end, we first study if the data follow a normal distribution through Kolmogorov-Smirnov-Lilliefor's [61] and Shapiro-Wilk's [75] tests, assuming the hypothesis H_0: data follow a normal distribution and H_1: otherwise. We obtained p-values lower than 0.05 for all the cases. Hence, we cannot assume that the data follow a normal distribution. Thus, we should consider the median as the average value. Note that Tables 6 and 7 where generated after performing this study and then the median was assumed.

Next, we study if there are significant differences among the binarization techniques. As samples are independent and data do not follow a normal distribution, we assume the Wilcoxon-Mann-Whitney's [74] test with the hypothesis. $H_0 : RPD_{a,b} \leq RPD_{c,d}\ \forall_{a,c} \in D_1, D_2 \ldots, D_5$ (discretization functions) and $\forall_{b,d} \in S_1, S_2, \ldots, V_4$ (transfer functions), where $RPD_{a,b}$ denotes the median RPD for a given combination of a and b. The $RPD_{c,d}$ definition is similar.

We analyze the p-values obtained by considering a significance level of 0.05. Based on this analysis and for each instance set, Table 8 shows the percentage of cases in which a binarization technique offers the best significant performance compared to all others. In this table, better values are shaded from a darker to a lighter tone, i.e., from better to worse behavior. Analyzing this table and starting with the instances with known optimal solutions, we reach that for the instance.

Analyzing this table and starting with the instances with known optimal solutions, we reach that for the instance set 4 the best combination is (D_1, V_3). For the set 5 are (D_3, S_4), (D_3, V_3), and (D_2, V_4). For the set 6 is (D_3, V_4). For the set A is (D_2, V_3). For the set B is (D_3, S_2). For the set C is (D_4, V_4). For the set D is (D_1, S_2). On average term, the best combination is (D_5, V_3), followed by (D_2, V_3) and (D_5, V_2).

Regarding instances with unknown optimal solutions, we reach that for the set NRE the best combination is (D_4, V_4). For the set NRF is also (D_4, V_4). For the set NRG is (D_1, S_3). For the set NRH is also (D_1, S_3). On average term, the best combination is (D_1, S_3), followed by (D_4, V_4) and (D_2, S_2).

As a result, we note that v-shape transfer functions give a better behavior for solving small and medium problems, i.e., the instance sets with known optimal solutions and the two smaller sets with unknown optimal solutions. On the other hand, s-shape transfer functions give a better behavior for the two remaining sets with unknown optimal solutions, the largest ones.

Regarding discretization functions, we do not observe any relevant trend. However, we note that it is crucial to select the adequate combination of both transfer and discretization functions to ensure that the solving algorithm reaches its full potential.

As a proof of this, we reach that v-shape transfer functions provide a good behavior for solving small and medium instances, e.g., for the set NRE. However, if we consider the combination (D_5, V_4) instead of the before mentioned (D_4, V_4), the percentage of cases providing the best performance is 0.00% instead of 30.77%, meaning a bad behavior.

In terms of RPD, we study how affects using an adequate binarization technique. Table 5 compares the results obtained through the original BCSO to the binarization techniques analyzed in this work. In this table, $diff_{rpd}$ is the difference between the RPD value obtained from the best binarization technique in this work, $\overline{rpd}$ field, and the original BCSO, $\overline{rpd}$ (original) field. Analyzing this table, we note that the algorithm provides a clear better behavior when an adequate binarization technique is assumed. This way, the RPD value decreases up to 26.19% for the instance set 4, 16.18% for the instance set 5, 10.23% for the instance set 6, 8.32% for the instance set A, 10.25% for the instance set B, 6.56% for the instance set C, 6.37% for the instance set D, 12.43% for the instance set NRE, 5.90% for the instance set NRF, 5.74% for the set instance NRG, and 4.55% for the instance set NRH.

Inst.	Trans.	Discr.	z_{opt}	z_{best}	z_{avg}	$\overline{rpd}$	$\overline{rpd}$(default)	$diff_{\overline{rpd}}$
4.1	S_2	D_1	429	432	440.07	2.58	6.44	3.86
4.2	V_1	D_4	512	517	529.87	3.49	6.22	2.73
4.3	V_4	D_5	516	531	552.77	7.13	7.93	0.80
4.4	S_2	D_1	494	496	510.23	3.29	3.90	0.61
4.5	S_2	D_5	512	514	523.23	2.19	2.75	0.56
4.6	V_2	D_3	560	560	566.10	1.09	1.24	0.15
4.7	V_3	D_1	430	434	437.53	1.75	2.18	0.43
4.8	V_4	D_5	492	494	511.07	3.88	4.98	1.10
4.9	V_3	D_5	641	660	674.37	5.21	6.04	0.83
4.10	V_3	D_2	514	518	524.93	2.13	2.63	0.50
Avg.	—	—	—	—	—	3.27	4.43	1.16
5.1	V_1	D_1	253	258	261.54	3.37	3.77	0.40
5.2	V_3	D_3	302	306	313.30	3.74	5.11	1.37
5.3	S_2	D_4	226	229	232.73	2.98	3.58	0.60
5.4	V_3	D_3	242	242	245.13	1.29	1.49	0.20
5.5	S_1	D_3	211	216	219.43	4.00	4.31	0.31
5.6	V_1	D_3	213	217	223.41	4.89	6.12	1.23
5.7	V_2	D_1	293	294	303.40	3.55	4.60	1.05
5.8	V_4	D_4	288	294	305.70	6.15	6.42	0.27
5.9	S_2	D_5	279	280	280.42	0.51	1.49	0.98
5.10	S_4	D_3	265	271	274.80	3.7	3.92	0.22
Avg.	—	—	—	—	—	3.42	4.08	0.66
6.1	V_2	D_2	138	143	146.20	5.94	6.57	0.63
6.2	V_3	D_1	146	146	149.13	2.15	2.74	0.59
6.3	V_3	D_3	145	148	151.77	4.67	5.15	0.48
6.4	V_4	D_3	131	133	134.40	2.6	2.65	0.05
6.5	V_1	D_5	161	165	168.07	4.39	4.87	0.48
Avg.	—	—	—	—	—	3.95	4.40	0.45
A.1	V_1	D_1	253	271	274.67	8.56	9.16	0.60
A.2	S_3	D_1	252	259	264.27	4.87	5.16	0.29
A.3	V_3	D_2	232	238	242.53	4.54	5.19	0.65
A.4	S_2	D_4	234	241	244.90	4.66	5.07	0.41
A.5	V_2	D_3	236	237	238.47	1.05	1.27	0.22
Avg.	—	—	—	—	—	4.74	5.17	0.43
B.1	S_1	D_5	69	70	73.70	6.81	8.79	1.98
B.2	S_2	D_3	76	80	83.80	10.26	10.26	0.00
B.3	S_3	D_5	80	80	82.27	2.83	3.50	0.67
B.4	V_3	D_3	79	81	83.63	5.86	6.33	0.47
B.5	S_1	D_1	72	73	73.00	1.39	1.39	0.00
Avg.	—	—	—	—	—	5.43	6.05	0.62

Inst.	Trans.	Discr.	z_{opt}	z_{best}	z_{avg}	$\overline{rpd}$	$\overline{rpd}$(original)	$diff_{\overline{rpd}}$
C.1	V_1	D_5	227	232	234.30	3.22	3.39	0.17
C.2	V_2	D_1	219	225	229.07	4.6	5.43	0.83
C.3	S_1	D_2	243	251	264.07	8.67	9.42	0.75
C.4	S_1	D_3	219	231	237.70	8.54	8.86	0.32
C.5	V_4	D_4	215	222	228.60	6.33	6.47	0.14
Avg.	—	—	—	—	—	6.27	6.71	0.44
D.1	S_4	D_4	60	60	64.03	6.72	7.33	0.61
D.2	S_4	D_5	66	69	69.70	5.61	6.06	0.45
D.3	S_1	D_2	72	76	78.50	9.03	9.44	0.41
D.4	S_3	D_3	62	63	65.37	5.43	5.91	0.48
D.5	S_2	D_4	61	64	64.83	6.28	6.56	0.28
Avg.	—	—	—	—	—	6.61	7.06	0.45
NRE.1	S_1	D_1	29	30	30.00	3.45	3.45	0.00
NRE.2	V_2	D_1	30	34	34.00	13.33	15.56	2.23
NRE.3	V_4	D_4	27	29	31.87	18.02	23.21	5.19
NRE.4	V_4	D_4	28	32	32.73	16.9	17.86	0.96
NRE.5	S_1	D_1	28	30	30.00	7.14	7.14	0.00
Avg.	—	—	—	—	—	11.77	13.44	1.68
NRF.1	S_1	D_1	14	17	17.00	21.43	21.43	0.00
NRF.2	S_1	D_3	15	16	17.70	18.00	20.00	2.00
NRF.3	S_1	D_1	14	17	17.00	21.43	21.43	0.00
NRF.4	V_2	D_3	14	15	16.87	20.48	25.00	4.52
NRF.5	S_1	D_1	13	16	16.00	23.08	23.08	0.00
Avg.	—	—	—	—	—	20.88	22.19	1.30
NRG.1	S_1	D_1	176	191	193.10	9.72	10.3	0.58
NRG.2	S_3	D_1	154	165	166.43	8.07	8.79	0.72
NRG.3	S_2	D_2	166	182	182.00	9.64	9.92	0.28
NRG.4	V_1	D_4	168	180	182.87	8.85	9.15	0.30
NRG.5	S_3	D_1	168	183	183.00	8.93	9.80	0.87
Avg.	—	—	—	—	—	9.04	9.59	0.55
NRH.1	S_3	D_3	63	69	71.00	12.7	15.19	2.49
NRH.2	S_1	D_1	63	67	67.00	6.35	6.35	0.00
NRH.3	S_1	D_1	59	69	69.00	16.95	16.95	0.00
NRH.4	S_2	D_5	58	64	66.73	15.06	15.52	0.46
NRH.5	S_1	D_1	55	61	61.00	10.91	10.91	0.00
Avg.	—	—	—	—	—	12.39	12.98	0.59

Table 5: Comparing results.

Discr.	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5
Trans.	S_1					S_2					S_3					S_4				
Instance																				
4.1	3.58	2.80	2.84	3.80	3.24	2.58	6.11	6.44	2.74	3.16	2.83	6.34	6.85	3.01	3.11	3.08	5.93	3.92	3.70	3.13
4.2	5.11	4.11	4.77	3.89	4.71	4.40	8.44	6.22	4.62	3.81	4.40	8.83	6.00	5.16	5.05	4.72	8.14	5.03	3.71	4.05
4.3	7.95	7.31	7.18	7.54	8.09	7.98	11.92	7.93	7.71	7.96	7.90	11.52	7.95	7.41	8.28	7.49	10.72	7.86	8.51	7.89
4.4	4.15	4.45	5.26	5.16	3.62	3.29	5.92	3.90	4.28	3.85	4.44	7.32	3.73	3.61	4.22	4.68	6.71	3.77	3.91	3.54
4.5	2.77	2.78	2.44	2.46	2.55	2.51	4.28	2.75	2.55	2.19	2.81	4.36	2.53	2.60	2.40	2.29	4.19	2.57	2.42	2.34
4.6	1.73	2.33	1.24	1.55	1.61	1.74	4.37	1.24	1.32	1.42	1.50	3.92	1.66	1.42	1.68	1.57	4.92	1.43	1.30	1.13
4.7	2.09	2.05	2.36	2.29	2.19	2.57	3.97	2.18	2.24	2.26	2.09	4.00	2.13	2.15	2.52	2.33	4.29	2.07	2.22	1.98
4.8	5.65	4.07	5.09	4.95	4.63	4.75	10.00	4.98	5.22	4.30	4.78	8.64	3.92	4.97	5.09	4.80	9.48	5.63	5.73	5.37
4.9	5.82	5.82	5.54	5.40	5.83	5.90	8.32	6.04	5.60	5.92	5.28	8.13	5.96	5.68	5.46	5.63	6.97	5.37	5.43	5.55
4.10	2.52	2.72	2.79	2.15	2.61	2.85	4.99	2.63	2.57	3.17	2.51	4.51	2.27	2.24	2.83	3.05	5.40	2.72	2.53	2.47
5.1	3.62	4.41	3.97	3.79	4.16	4.44	6.09	3.77	3.60	3.78	4.22	6.23	4.02	3.54	4.49	4.08	5.72	3.53	3.65	3.75
5.2	4.48	4.25	4.62	4.58	4.12	4.45	6.92	5.11	4.23	4.02	4.18	6.56	4.07	4.30	3.77	4.42	6.58	4.03	4.24	4.67
5.3	3.72	4.31	3.88	3.83	3.39	3.94	6.28	3.58	2.98	3.01	3.42	6.14	3.33	3.08	3.39	3.67	6.08	3.61	3.94	3.86
5.4	1.58	1.60	1.46	1.50	1.47	1.45	1.96	1.49	1.54	1.46	1.47	1.97	1.46	1.49	1.56	1.52	1.87	1.38	1.51	1.53
5.5	4.49	4.09	4.00	4.19	4.33	4.25	5.61	4.31	4.27	4.44	4.04	5.37	4.27	4.23	4.28	4.66	5.61	4.28	4.31	4.27
5.6	6.24	5.99	5.70	6.65	5.99	5.46	10.53	6.12	6.12	5.20	6.29	9.61	6.32	6.04	6.08	5.54	9.56	6.40	5.07	5.18
5.7	4.43	4.82	4.45	4.23	4.37	4.55	9.31	4.60	4.45	4.46	3.61	9.62	3.79	4.53	4.23	4.82	9.40	4.78	3.98	5.01
5.8	6.72	6.30	6.69	6.62	6.82	7.06	10.31	6.42	6.82	6.85	7.03	11.04	6.63	6.55	6.69	6.22	10.57	6.40	7.30	6.82
5.9	0.61	1.18	1.46	0.56	0.62	0.58	1.58	1.49	0.62	0.51	0.72	1.95	0.72	1.18	1.18	0.58	1.82	0.61	0.62	1.15
5.10	4.05	4.11	3.92	4.00	4.36	3.90	5.60	3.92	4.23	4.30	4.05	5.58	4.29	4.14	3.95	3.87	5.26	3.70	4.23	4.09
6.1	7.10	7.08	7.00	7.15	7.00	6.67	12.44	6.57	6.84	7.29	6.45	11.74	7.78	7.51	6.62	7.08	12.63	7.46	7.32	6.57
6.2	3.07	2.95	2.94	2.93	2.97	2.74	5.21	2.74	2.74	2.74	2.74	5.27	2.74	2.99	3.02	2.74	5.32	2.74	2.74	2.95
6.3	4.94	5.31	5.21	5.10	5.06	5.22	6.28	5.15	5.10	4.90	5.26	6.90	5.13	5.54	5.13	4.74	6.25	5.82	4.87	5.86
6.4	2.65	2.93	2.80	3.00	2.93	2.90	3.64	2.65	3.03	2.93	2.88	3.92	2.95	2.98	3.23	3.08	3.94	3.00	2.93	3.18
6.5	4.84	5.24	5.01	4.64	5.07	5.09	6.21	4.87	5.16	5.65	5.13	6.31	5.57	5.18	4.93	4.80	6.25	5.20	4.89	4.70
A.1	8.87	8.85	8.85	9.00	9.14	9.16	10.84	9.16	9.18	9.45	8.74	10.67	8.75	9.33	9.16	9.05	10.75	9.10	8.88	9.28
A.2	5.03	5.16	5.20	5.20	5.09	5.38	6.20	5.16	5.17	5.21	4.87	6.19	5.34	5.20	5.05	5.05	6.18	5.13	5.12	5.24
A.3	5.13	5.47	4.86	4.68	5.47	5.17	8.72	5.19	5.01	5.01	5.46	7.82	4.86	4.84	5.26	5.53	7.57	4.70	4.77	5.29
A.4	4.91	4.83	5.26	4.96	5.37	4.99	5.90	5.07	4.66	4.91	5.16	6.04	5.24	5.24	5.09	4.93	6.41	5.41	4.91	5.07
A.5	1.27	1.26	1.12	1.13	1.17	1.26	1.57	1.27	1.27	1.12	1.21	1.36	1.30	1.27	1.23	1.36	1.41	1.16	1.19	1.26
B.1	8.41	8.84	7.58	8.21	6.81	7.78	10.34	8.79	7.83	8.70	8.45	11.64	7.83	7.90	8.45	7.63	11.26	8.36	8.02	8.16
B.2	11.49	11.49	12.02	11.71	11.93	12.46	14.74	10.26	12.11	11.58	11.67	15.04	12.37	11.71	11.40	11.40	14.21	12.50	11.45	11.45
B.3	3.54	3.29	3.46	4.00	3.79	3.75	5.29	3.50	3.54	3.83	3.25	6.46	3.54	3.42	2.83	4.04	6.00	3.87	3.67	2.92
B.4	6.33	6.33	6.33	6.33	6.33	6.33	7.17	6.33	6.29	6.33	6.33	7.26	6.33	6.33	6.33	6.33	7.05	6.33	6.33	6.54
B.5	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39
C.1	3.38	3.25	3.32	3.33	3.30	3.38	4.41	3.39	3.26	3.36	3.32	4.82	3.33	3.36	3.42	3.27	4.76	3.32	3.45	3.38
C.2	5.13	5.10	5.05	5.04	5.33	5.18	6.93	5.43	5.31	4.84	5.08	6.83	5.07	5.16	5.25	4.63	7.05	5.08	5.07	4.87
C.3	9.16	8.67	9.20	8.90	9.11	9.27	12.47	9.42	9.40	9.81	9.01	12.17	9.47	9.00	8.94	8.79	12.72	9.57	9.49	9.55
C.4	8.87	8.81	8.54	9.01	9.19	9.24	11.17	8.86	9.39	8.86	11.71	11.05	9.33	9.27	8.71	9.21	11.29	9.06	9.68	8.93
C.5	6.96	7.18	7.18	7.12	7.12	6.79	9.22	6.47	6.88	7.97	9.57	9.49	7.49	7.16	6.88	7.09	8.56	6.81	7.07	7.32
D.1	8.33	8.64	7.94	7.67	8.64	9.01	7.72	7.33	8.97	7.67	9.14	7.78	8.33	8.77	8.33	7.56	7.67	8.33	6.72	8.83
D.2	6.06	6.06	6.06	6.06	6.06	5.71	6.06	6.06	6.06	6.06	6.06	5.71	6.06	5.71	6.06	6.06	6.06	6.06	6.06	5.61
D.3	9.26	9.03	9.31	9.21	9.07	9.12	9.31	9.44	9.12	9.72	10.09	9.72	9.72	9.72	9.72	9.40	9.21	9.35	9.26	9.12
D.4	6.13	6.24	6.34	6.51	6.13	5.81	6.29	5.91	5.97	5.97	7.37	6.40	5.43	6.13	5.86	6.08	6.18	5.59	6.02	6.13
D.5	6.99	6.67	6.99	6.61	6.83	6.50	6.56	6.56	6.28	6.50	7.70	7.05	7.16	6.45	6.72	6.72	7.16	6.89	6.39	6.61
NRE.1	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45
NRE.2	14.56	15.00	14.67	15.11	15.00	14.89	15.00	15.56	15.33	15.00	15.89	15.00	14.56	15.22	14.78	15.11	14.78	15.00	15.11	15.56
NRE.3	23.21	23.58	21.36	22.72	22.96	23.95	24.07	23.21	24.20	23.58	23.95	21.85	22.84	22.10	23.70	23.46	23.09	23.21	22.72	22.72
NRE.4	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86
NRE.5	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14
NRF.1	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43
NRF.2	20.00	20.00	18.00	20.00	20.00	20.00	20.00	20.00	18.44	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
NRF.3	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43
NRF.4	23.81	23.33	21.43	24.76	24.76	25.00	24.52	25.00	21.43	24.29	23.81	22.86	22.86	24.05	24.52	24.52	24.05	22.62	24.05	23.10
NRF.5	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08
NRG.1	9.72	10.36	10.28	10.27	10.42	9.85	10.02	10.30	10.32	10.52	9.79	10.00	10.30	10.45	10.27	10.32	10.51	10.35	10.34	10.31
NRG.2	8.81	8.61	8.66	8.81	8.66	8.77	8.14	8.79	8.61	8.74	8.07	8.16	8.81	8.79	8.61	8.79	8.48	8.64	8.87	8.64
NRG.3	9.94	10.04	9.98	9.84	9.86	9.96	9.64	9.92	9.90	9.90	9.64	9.64	9.94	10.06	10.00	10.00	9.94	9.92	9.90	9.94
NRG.4	9.15	9.25	9.13	9.13	9.09	9.01	8.97	9.15	9.27	9.11	8.89	9.15	9.52	9.01	9.07	9.13	9.21	9.17	9.52	9.11
NRG.5	10.08	9.88	9.92	10.02	9.84	9.94	9.92	9.80	9.76	10.06	8.93	10.00	9.70	9.88	9.96	9.92	10.00	10.06	9.66	9.76
NRH.1	15.19	14.87	14.92	15.19	15.19	14.97	14.50	15.19	14.97	15.87	12.70	14.81	15.08	14.81	14.71	15.34	15.19	14.71	14.81	15.34
NRH.2	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35
NRH.3	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95
NRH.4	15.29	15.75	15.80	15.40	15.69	15.29	15.52	15.52	15.52	15.06	15.52	15.75	15.98	15.52	15.46	15.81	15.52	15.63	15.69	15.11
NRH.5	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91

Table 6: Average RPD for each instance and binarization technique: S-shape transference functions.

Discr.	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5	D_1	D_2	D_3	D_4	D_5
Trans.			V_1					V_2					V_3					V_4		
Instance																				
4.1	2.76	3.50	3.14	3.30	2.94	3.01	3.08	3.28	3.64	3.05	2.69	3.64	3.19	3.48	3.61	3.23	6.90	2.95	3.57	3.38
4.2	4.45	4.20	4.55	3.49	4.18	5.18	4.52	4.88	3.72	4.14	3.95	4.83	4.46	4.15	4.30	3.74	5.62	5.02	4.11	4.34
4.3	8.31	7.77	7.73	7.69	7.67	7.40	7.51	7.20	8.54	7.96	7.55	8.64	7.84	8.12	7.59	7.50	7.70	7.64	7.81	7.13
4.4	4.95	4.20	3.73	4.89	3.79	3.87	4.22	4.14	3.60	4.15	4.14	3.92	3.91	4.64	4.57	3.92	4.33	4.10	3.70	4.23
4.5	2.29	2.86	2.89	2.66	2.32	2.49	3.07	2.42	2.49	2.74	2.94	2.58	2.43	2.81	2.37	2.36	2.79	3.25	2.73	2.54
4.6	1.33	1.47	1.83	1.53	1.23	1.57	1.50	1.09	1.42	1.26	1.55	1.40	1.55	1.39	1.15	1.68	1.57	1.70	1.32	1.51
4.7	2.49	2.18	2.24	2.22	2.44	2.42	2.35	2.27	2.41	1.91	1.75	2.18	2.60	2.47	2.54	2.17	2.51	2.50	2.11	2.75
4.8	5.06	5.43	5.14	5.29	5.15	5.62	5.12	4.56	4.80	5.49	5.28	4.19	4.74	5.35	5.38	4.83	5.87	5.64	4.71	3.88
4.9	5.66	5.72	5.73	5.45	5.61	5.62	5.46	5.69	5.94	6.00	5.84	5.65	5.46	5.93	5.21	5.91	5.56	5.40	5.52	5.54
4.10	2.69	2.74	2.50	2.13	2.82	3.05	2.52	2.85	2.44	2.96	2.44	2.13	2.68	2.92	2.36	2.26	2.85	2.84	2.38	3.15
5.1	3.37	3.76	3.63	3.80	4.57	3.99	4.68	3.72	3.97	4.01	3.63	3.87	3.71	3.47	3.82	4.58	3.38	3.86	3.83	3.76
5.2	4.03	4.46	4.69	4.15	4.19	4.27	4.32	4.48	3.86	3.82	4.38	4.14	3.74	4.07	4.37	4.18	4.17	4.37	3.93	3.98
5.3	3.13	3.27	4.07	3.35	3.45	3.73	3.69	3.29	3.44	3.38	4.12	3.32	3.39	3.55	3.92	3.50	4.00	3.53	3.22	3.61
5.4	1.54	1.58	1.45	1.52	1.57	1.45	1.63	1.46	1.54	1.34	1.54	1.53	1.29	1.57	1.53	1.36	1.58	1.47	1.52	1.49
5.5	4.09	4.15	4.33	4.60	4.30	4.27	4.47	4.25	4.41	4.22	4.50	4.45	4.15	4.27	4.17	4.41	4.27	4.47	4.15	4.38
5.6	6.49	5.83	4.89	6.43	5.93	5.60	6.06	6.79	5.57	6.12	5.23	5.92	5.76	6.10	5.76	6.10	5.76	5.82	5.70	5.20
5.7	4.11	4.66	4.82	4.32	4.82	3.55	3.92	4.33	4.49	4.97	4.77	4.62	4.23	4.62	4.08	4.23	4.02	3.88	4.27	4.97
5.8	7.25	7.12	7.09	6.28	6.52	6.57	6.33	6.98	6.68	6.89	6.61	6.47	6.38	6.70	6.62	6.71	7.23	6.70	v[5pt]6.15	6.68
5.9	0.72	1.46	1.34	0.58	0.55	0.58	0.72	1.23	0.63	1.22	0.72	0.66	0.61	1.31	0.55	1.28	0.57	0.58	1.08	0.58
5.10	4.24	4.25	4.06	4.20	4.42	4.26	3.94	4.20	4.04	4.13	4.00	3.94	3.89	4.16	3.92	4.10	4.25	4.04	4.10	3.96
6.1	7.10	7.05	6.84	6.52	6.76	7.03	5.94	6.79	7.25	7.44	6.47	6.52	7.37	7.39	6.11	6.64	7.87	7.05	6.98	7.49
6.2	2.74	2.74	2.99	2.95	2.95	2.74	3.03	3.10	3.01	2.74	2.15	2.74	2.97	2.94	2.74	3.03	2.92	2.74	3.06	2.74
6.3	5.47	4.99	5.66	4.83	5.26	5.22	4.76	4.99	4.85	5.06	5.17	4.87	4.67	5.20	5.36	5.06	4.97	4.74	5.08	5.43
6.4	2.98	3.00	2.82	3.13	3.05	3.05	3.21	3.10	2.95	2.65	3.05	2.75	2.70	3.05	2.95	2.93	3.00	2.60	2.72	2.90
6.5	4.97	4.76	4.91	5.05	4.39	4.70	5.07	5.11	5.22	5.09	4.93	5.09	5.49	4.99	4.97	4.82	4.72	4.45	5.01	5.01
A.1	8.56	8.59	9.14	9.09	9.01	9.20	9.20	9.03	9.03	8.91	8.84	9.06	9.01	8.92	8.87	8.99	8.79	8.83	8.88	8.88
A.2	5.25	5.13	5.37	5.13	5.21	5.11	5.20	5.17	5.46	5.30	5.28	5.16	5.34	5.08	4.99	5.25	5.20	5.20	5.13	5.26
A.3	5.79	4.83	5.01	5.57	5.24	5.13	5.10	5.16	5.20	4.77	4.94	4.54	5.07	5.70	5.07	5.73	5.50	5.99	5.16	5.65
A.4	4.90	5.40	4.94	5.11	4.96	4.87	5.28	5.20	4.87	5.11	5.28	4.99	5.23	4.81	5.09	4.97	5.16	5.10	4.86	5.11
A.5	1.10	1.19	1.17	1.21	1.33	1.19	1.17	1.05	1.10	1.14	1.34	1.05	1.29	1.24	1.34	1.20	1.27	1.09	1.27	1.27
B.1	8.21	8.12	8.12	8.12	7.87	7.87	8.26	8.16	8.89	7.58	8.55	8.07	8.36	8.55	7.97	8.36	7.39	8.41	8.21	8.12
B.2	11.75	11.97	11.23	11.18	12.15	12.11	10.83	11.40	11.75	10.75	12.50	11.71	10.83	11.93	11.18	10.75	11.54	11.40	11.62	10.79
B.3	3.21	4.04	3.58	3.92	4.33	3.50	2.96	3.71	3.46	4.08	3.96	3.38	3.50	3.75	4.08	3.08	3.75	3.08	2.83	3.46
B.4	6.33	6.33	6.33	6.33	6.33	6.33	6.62	6.33	6.33	6.33	6.84	6.33	5.86	6.33	6.33	6.33	6.33	6.58	6.33	6.33
B.5	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39	1.39
C.1	3.35	3.52	3.35	3.29	3.22	3.25	3.39	3.30	3.35	3.25	3.45	3.52	3.38	3.30	3.52	3.39	3.37	3.31	3.30	3.41
C.2	4.84	5.08	5.39	4.75	5.14	4.60	5.48	4.73	5.56	5.33	5.14	4.81	5.24	5.21	5.62	5.81	4.89	5.19	5.13	5.07
C.3	9.57	9.44	9.26	9.40	9.37	9.68	9.95	9.38	9.33	8.82	9.31	9.36	9.56	9.73	9.44	9.55	8.86	9.57	8.81	9.48
C.4	8.89	9.27	9.41	9.33	9.42	8.80	9.24	9.63	9.30	10.12	9.36	9.24	9.12	9.50	9.09	8.75	8.92	9.06	9.32	8.78
C.5	6.42	6.84	7.01	6.54	7.01	6.81	7.15	7.30	6.93	7.69	6.43	6.82	6.99	6.64	6.93	7.09	7.09	7.13	6.33	6.74
D.1	8.64	7.67	7.39	6.94	7.89	7.44	6.78	7.28	7.44	7.67	7.72	7.59	7.83	7.78	8.87	7.83	7.33	7.00	7.78	7.94
D.2	6.06	6.06	6.06	6.06	6.06	6.06	6.06	6.06	6.06	5.66	6.06	6.06	6.06	5.71	5.71	6.06	6.06	6.06	6.06	6.06
D.3	9.44	9.44	9.31	9.07	9.21	9.40	9.07	9.21	9.21	9.07	9.72	9.21	9.17	9.21	9.21	9.72	9.40	9.31	9.72	9.72
D.4	5.97	6.40	5.91	5.48	6.08	6.40	6.02	6.45	6.83	5.81	5.97	6.02	6.18	6.18	6.08	6.18	6.29	6.02	6.61	6.08
D.5	6.61	7.05	6.56	6.89	6.72	6.83	6.78	6.94	6.50	7.21	6.94	6.45	6.56	6.72	6.72	6.89	6.39	6.67	6.83	6.67
NRE.1	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45	3.45
NRE.2	14.89	15.00	15.00	15.44	15.33	13.33	15.00	15.11	14.56	14.67	15.11	14.89	15.67	14.67	15.44	15.11	15.11	15.33	14.89	15.44
NRE.3	22.72	22.35	22.47	23.46	22.59	23.95	22.35	22.72	23.58	22.59	21.60	22.59	22.47	22.72	22.96	22.22	23.21	22.35	18.02	23.09
NRE.4	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	17.86	16.90	17.86
NRE.5	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14	7.14
NRF.1	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43
NRF.2	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	18.22	20.00
NRF.3	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43	21.43
NRF.4	23.33	24.05	23.10	24.05	23.81	22.62	24.05	20.48	22.86	23.81	23.33	23.57	24.29	22.86	23.33	24.05	21.43	23.10	21.43	23.81
NRF.5	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08	23.08
NRG.1	10.19	10.32	10.17	10.40	10.21	10.36	10.28	10.23	10.44	10.32	10.30	10.51	10.45	10.19	10.27	10.53	10.28	10.42	10.36	10.34
NRG.2	8.66	8.77	8.72	8.70	8.44	8.87	8.79	8.83	8.96	8.94	8.92	8.87	9.18	8.70	9.05	8.61	8.85	8.96	9.05	8.90
NRG.3	9.98	10.02	9.88	9.84	10.02	9.94	10.02	9.94	9.94	9.98	9.94	10.00	9.98	9.96	9.96	9.94	9.94	10.06	10.00	9.96
NRG.4	9.09	9.52	9.29	8.85	9.21	9.21	9.13	9.21	9.09	8.95	9.07	9.09	9.19	8.99	9.03	8.95	9.15	9.27	9.13	9.29
NRG.5	10.04	9.66	9.78	9.90	9.82	9.90	10.00	10.14	9.90	9.52	9.92	9.96	9.82	9.98	9.82	9.98	10.26	10.04	10.14	9.94
NRH.1	14.39	15.08	15.40	14.97	15.03	15.29	15.08	14.87	14.87	14.87	15.24	14.97	14.76	15.45	14.92	14.92	15.13	14.55	15.13	14.92
NRH.2	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35	6.35
NRH.3	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95	16.95
NRH.4	15.52	15.29	15.86	15.69	15.52	15.52	15.52	15.86	15.92	15.52	15.52	15.52	16.03	15.80	15.52	15.52	15.52	15.23	15.34	15.80
NRH.5	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91	10.91

Table 7: Average RPD for each instance and binarization technique: V-shape transference functions.

Discr.	Trans.	4	5	6	A	B	C	D	Average	NRE	NRF	NRG	NRH	Average
		Instance Set												
D_1		1.87%	2.62%	2.46%	3.03%	1.97%	2.47%	2.08%	2.33%	3.85%	0.00%	4.55%	2.36%	3.35%
D_2		3.79%	2.06%	1.36%	2.90%	2.37%	2.68%	2.08%	2.52%	0.38%	0.69%	1.22%	0.39%	0.88%
D_3	S_1	2.72%	2.06%	1.86%	2.64%	2.50%	2.88%	1.77%	2.33%	7.69%	20.49%	0.78%	0.39%	5.11%
D_4		2.99%	2.62%	1.61%	3.03%	1.97%	2.88%	2.49%	2.56%	0.00%	0.00%	0.44%	1.97%	0.53%
D_5		2.08%	2.47%	1.61%	2.11%	4.34%	2.57%	2.28%	2.39%	0.38%	0.00%	1.11%	0.39%	0.70%
D_1		3.15%	2.77%	3.22%	1.71%	2.37%	2.47%	5.50%	3.06%	0.77%	0.00%	4.88%	3.94%	3.29%
D_2		0.11%	0.00%	0.08%	0.00%	0.26%	0.10%	2.28%	0.33%	0.38%	0.00%	12.32%	3.15%	7.05%
D_3	S_2	1.60%	1.54%	4.15%	1.84%	5.39%	3.30%	2.39%	2.59%	0.00%	0.00%	1.55%	0.79%	0.94%
D_4		2.40%	2.93%	3.05%	3.95%	2.11%	2.57%	3.01%	2.82%	0.00%	22.57%	2.33%	1.97%	5.34%
D_5		3.47%	4.01%	3.22%	3.29%	1.97%	2.47%	1.35%	3.05%	0.38%	0.00%	0.44%	2.36%	0.65%
D_1		3.15%	2.77%	3.14%	3.82%	2.24%	1.44%	0.00%	2.48%	0.00%	0.00%	17.54%	16.93%	11.80%
D_2		0.11%	0.00%	0.00%	0.13%	0.00%	0.00%	4.15%	0.51%	5.00%	1.39%	12.10%	1.18%	7.57%
D_3	S_3	3.21%	2.52%	3.05%	2.24%	2.37%	2.27%	1.77%	2.59%	3.85%	0.35%	1.55%	0.39%	1.53%
D_4		2.88%	2.62%	1.27%	1.98%	2.37%	2.37%	4.36%	2.58%	2.69%	0.00%	1.00%	1.97%	1.23%
D_5		1.98%	2.06%	1.69%	1.98%	4.21%	2.68%	1.04%	2.13%	1.54%	0.00%	1.22%	2.76%	1.29%
D_1		2.30%	2.57%	3.64%	2.50%	2.76%	3.81%	2.18%	2.77%	0.38%	0.00%	1.00%	0.39%	0.65%
D_2		0.27%	0.00%	0.08%	0.26%	0.00%	0.21%	1.66%	0.31%	1.54%	0.00%	2.00%	1.97%	1.59%
D_3	S_4	2.19%	4.16%	2.71%	1.98%	1.97%	2.57%	2.60%	2.77%	0.38%	2.43%	1.66%	3.15%	1.82%
D_4		2.94%	2.88%	3.31%	3.43%	2.37%	2.06%	3.12%	2.89%	0.38%	0.00%	2.00%	0.39%	1.17%
D_5		3.15%	2.26%	1.69%	1.71%	2.89%	1.96%	5.09%	2.67%	0.00%	0.00%	2.33%	4.72%	1.94%
D_1		2.51%	3.34%	3.05%	4.22%	2.11%	3.71%	1.56%	2.92%	0.38%	0.69%	2.00%	7.48%	2.35%
D_2		2.08%	1.85%	3.31%	4.08%	1.97%	2.06%	1.97%	2.35%	1.15%	0.00%	1.11%	3.94%	1.35%
D_3	V_1	2.24%	2.72%	1.61%	2.24%	2.37%	2.27%	2.39%	2.30%	0.77%	1.39%	1.78%	0.39%	1.35%
D_4		3.90%	2.93%	1.78%	1.84%	2.63%	3.71%	3.53%	3.02%	0.00%	0.00%	3.44%	1.18%	2.00%
D_5		2.56%	2.57%	2.46%	1.71%	2.24%	3.19%	2.18%	2.47%	0.38%	0.00%	3.11%	1.97%	2.00%
D_1		2.30%	3.55%	3.39%	2.77%	2.11%	3.81%	2.80%	2.99%	15.00%	0.69%	0.44%	1.97%	2.94%
D_2		2.35%	2.11%	3.73%	1.98%	3.82%	2.27%	2.28%	2.57%	1.15%	0.00%	1.11%	1.57%	1.00%
D_3	V_2	2.56%	1.80%	1.69%	3.95%	1.71%	2.78%	1.77%	2.25%	0.38%	11.81%	1.44%	1.18%	2.99%
D_4		3.10%	3.03%	1.69%	3.03%	2.11%	2.27%	2.39%	.62%	3.85%	0.00%	0.44%	0.79%	0.94%
D_5		2.72%	2.47%	3.90%	2.37%	3.68%	2.27%	4.78%	3.06%	3.08%	0.00%	4.33%	1.97%	3.05%
D_1		4.11%	1.95%	3.47%	2.24%	1.18%	3.09%	0.52%	2.57%	5.00%	0.69%	0.55%	1.97%	1.47%
D_2		2.88%	2.57%	3.90%	5.53%	1.97%	2.78%	2.80%	3.09%	0.77%	0.00%	0.44%	1.97%	0.65%
D_3	V_3	2.30%	4.16%	2.63%	1.84%	3.16%	2.47%	3.12%	2.92%	1.92%	0.00%	0.89%	0.39%	0.82%
D_4		2.08%	2.57%	1.53%	3.29%	1.97%	3.09%	5.30%	2.70%	2.69%	1.74%	1.33%	0.39%	1.47%
D_5		2.94%	2.98%	4.07%	2.50%	2.50%	2.06%	4.67%	3.12%	0.00%	0.69%	1.22%	2.36%	1.12%
D_1		3.26%	1.75%	1.86%	1.98%	3.95%	2.06%	0.83%	2.25%	0.77%	0.00%	2.33%	1.57%	1.59%
D_2		1.50%	4.16%	1.78%	2.11%	3.42%	2.78%	2.28%	2.62%	0.38%	11.11%	0.67%	1.97%	2.58%
D_3	V_4	2.35%	3.03%	5.76%	3.29%	2.63%	2.37%	2.08%	3.06%	1.92%	0.00%	0.33%	11.42%	2.17%
D_4		2.67%	2.77%	2.12%	2.64%	4.21%	4.84%	1.04%	2.82%	30.77%	23.26%	0.44%	3.54%	9.40%
D_5		3.21%	2.77%	3.05%	1.84%	3.82%	2.37%	0.52%	2.62%	0.00%	0.00%	0.55%	0.39%	0.35%
		100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 8: Statistical analysis. Percentage of cases in which a binarization technique offers the best significant performance compared to all others.

10 Conclusion

In this work we use a binary version of cat swarm optimization, to solve SCP using its column based representation. In binary optimization problems the position vector is binary. This causes a significant change in BCSO with respect to CSO with real numbers. In fact, in BCSO in the seeking mode the slight change in the position takes place by introducing the mutation operation. The interpretation of velocity vector in tracing mode also changes to probability of change in each dimension of position of the cats.

The proposed BCSO was implemented and tested using 65 SCP test instances from the OR-Library of Beasley. In addition, five discretization functions and eight transfer functions were combined and tested, resulting forty tracing modes, with the purpose of improving the results obtained with the original behaviors. These were analyzed using RPD and a statistical analysis, to demostrate on a solid basis the best combination.

We could see the premature convergence problem, a typical problem in metaheuristics, which occurs when the cats quickly attain to dominate the population, constraining it to converge to a local optimum. For future works the objective will be make them highly immune to be trapped in local optima and thus less vulnerable to premature convergence problem. Thus, we could propose an algorithm that shows improved results in terms of both computational time and quality of solution.

We find significant performance differences according to the binarization approach assumed when an SIA (the BCSO) is adapted to the discrete scope. Hence, it is crucial to select an adequate binarization approach. Otherwise, it is possible that the algorithm does not reach its full potential as occurs with the original BCSO compared to the recommended configurations obtained in this work. As a direct result of this statement, it is possible that other algorithms could be improved by studying other binarization approaches.

Comparing with previous work, for most instances the combinations gave better results were the transfer functions S_1 and S_2 with the Roulette Wheel and Complement method. Moreover, it could also better solutions using different parameter setting for each set of instances. As can be seen from the results, metaheuristic performs well in all cases observed according to old RPD works [26].

But, we did not find any relevant trend, which leads us to recommend a specific binarization approach for solving the problem. Instead, we recommend studying different methods and not only one, as did most works in the field when adapting an algorithm to the discrete scope.

We appreciably increase the BCSO performance after selecting an adequate binarization approach for each instance. However, we cannot recommend this algorithm for solving the SCP, due to it is far from other current state of-the-art techniques in terms of performance.

The main and specific goals of this thesis work were successfully achieved. Moreover, it should be mentioned that the following papers based on this thesis work were published: "A Binary Cat Swarm Optimization Algorithm for the Non-Unicost Set Covering Problem" was published in the Mathematical Problems in Engineering Journal [26]; "Binary Cat Swarm Optimization for the Set Covering Problem" was published in the 2015 10th Iberian Conferenceon Information Systemsand Technologies, CISTI 2015 [24]; "Solving the Set Covering Problem with Binary Cat Swarm Optimization" was published in 6th International Conference, ICSI 2015 [25]; and finally was published "Cat Swarm Optimization with Different Transfer Functions for Solving Set Covering Problems" in Computational Science and Its Applications, ICCSA 2016 [39].

For future work, there are different directions that researchers may take, some of them are the follows: Use a parameter tuning technique to find the appropriated values to help improve results, the use of Autonomous Search could help the algorithm to be able to self-tune the performance of the Binary Cat Swarm Optimization. Moreover, find an ideally standard configuration for the set covering type problems. Finally, it would be interesting to investigate the impact of binarization techniques on other binary algorithms solving different problems.

References

[1] R. Adamec. The interaction of hunger and prey in the domestic cat (felis catus): An adaptive hierarchy? *Behavioural Biology*, pages 263–272, 1976.

[2] H. Adler. Some factors of observation learning in cats. *Journal of Genetic Psychology*, pages 159–177, 1995.

[3] U. Aickelin. An indirect genetic algorithm for set covering problems. *Journal of the Operational Research Society*, 31(5):1118–1126, 2002.

[4] A. I. Ali and H. Thiagarajan. A network relaxation based enumeration algorithm for set partitioning. *European Journal of Operational Research*, 38(1):76–85, 1989.

[5] V. Aspinall. *Complete Textbook of Veterinary Nursing*. Butterworth–Heinemann, Oxford, UK: Butterworth-Heinemann, 2006.

[6] E. Balas and M. C. Carrera. A dynamic subgradient-based branch-and-bound procedure for set covering. *Operations Research*, 44(6):875–890, 1996.

[7] M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300–304, 1964.

[8] J. J. Bartholdi. A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research*, 29(3):501–510, 1981.

[9] J. E. Beasley. An algorithm for set covering problem. *European Journal of Operation Research*, 31(1):85–93, 1987.

[10] J. E. Beasley. A lagrangian heuristic for set covering problems. *Naval Research Logistics*, 37:151–164, 1990.

[11] J. E. Beasley and P. C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operation Research*, 94(2):392–404, 1996.

[12] J. E. Beasley and K. Jornsten. Enhancing an algorithm for set covering problems. *European Journal of Operational Research*, 58(2):293–300, 1992.

[13] M. Bellmore, H. J. Geenberg, and J. J. Jarvis. Multi-commodity disconnecting sets. *Management Science*, 16(6):B427–B433, 1970.

[14] M. Bellmore and H. D. Ratliff. Optimal defense of multi-commodity networks. *Management Science*, 18(4-part-i):B174–B185, 1971.

[15] M. A. Breuer. Simplification of the covering problem with application to boolean expressions. *J. ACM*, 17(1):166–181, January 1970.

[16] A. Caprara, M. Fischetti, and P. Toth. Algorithms for the set covering problem. *Annals of Operations Research*, 98:353–371, 2000.

[17] S. Ceria, P. Nobili, and A. Sassano. A lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programming*, 81(2):215–228, 1998.

[18] N. Christofides. Zero-one programming using non-binary tree-search. *Comput. J.*, 14(4):418–421, 1971.

[19] S. Chu and P. Tsai. Computational intelligence based on the behavior of cats. *International Journal of Innovative Computing, Information and Control*, pages 163 – 173, 2007.

[20] S. Chu, P. Tsai, and J. Pan. Cat swarm optimization. In *Trends in Artificial Intelligence*, pages 854–858. Springer-Verlag, Berlin, Heidelberg, 2006.

[21] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[22] B. Crawford, C. Lagos, C. Castro, and F. Paredes. A cultural algorithm for solving the set covering problem. In *Analysis and Design of Intelligent Systems using Soft Computing Techniques, a selection of papers from IFSA 2007*, volume 41 of *Advances in Soft Computing*, pages 408–415. Springer, 2007.

[23] B. Crawford, C. Lagos, C. Castro, and F. Paredes. A evolutionary approach to solve set covering. In *ICEIS 2007 - Proceedings of the Ninth International Conference on Enterprise Information Systems*, volume AIDSS, pages 356–363, Funchal, Madeira, Portugal, 2007.

[24] B. Crawford, R. Soto, N. Berrios, F. Johnson, and F. Paredes. Binary cat swarm optimization for the set covering problem. In *Proceedings of the 10th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–4, 2015.

[25] B. Crawford, R. Soto, N. Berrios, F. Johnson, and F. Paredes. Solving the set covering problem with binary cat swarm optimization. In *Advances in Swarm and Computational Intelligence*, volume 9140, pages 41–48, 2015.

[26] B. Crawford, R. Soto, N. Berrios, F. Johnson, F. Paredes, C. Castro, and E. Norero. A binary cat swarm optimization algorithm for the non-unicost set covering problem. *Mathematical Problems in Engineering*, 2015(Article ID 578541):1–8, 2015.

[27] B. Crawford, R. Soto, R. Cuesta, M. Olivares-Suárez, F. Johnson, and E. Olguín. Two swarm intelligence algorithms for the set covering problem. In *ICSOFT-EA 2014 - Proceedings of the 9th International Conference on Software Engineering and Applications*, pages 60–69, Vienna, Austria, 2014. SciTePress.

[28] B. Crawford, R. Soto, R. Cuesta, and F. Paredes. Application of the artificial bee colony algorithm for solving the set covering problem. *The Scientific World Journal*, 2014(Article ID 189164):1–8, 2014.

[29] B. Crawford, R. Soto, R. Cuesta, and F. Paredes. A new artificial bee colony algorithm for set covering problems. In *1st International Conference on Electrical Engineering and Information Technology (CEEIT)*, volume 63, pages 31–36. WIT press, 2014.

[30] B. Crawford, R. Soto, and E. Monfroy. Cultural algorithms for the set covering problem. In *Advances in Swarm Intelligence, 4th International Conference*, volume 7929 of *Lecture Notes in Computer Science*, pages 27–34. 2013.

[31] B. Crawford, R. Soto, E. Monfroy, F. Paredes, and W. Palma. A hybrid ant algorithm for the set covering problem. *Int. J. Phys. Sci.*, 6(19):4667–4673, 2011.

[32] B. Crawford, R. Soto, M. Olivares-Suárez, W. Palma, F. Paredes, E. Olguin, and E. Norero. A binary coded firefly algorithm that solves the set covering problem. *Romanian Journal of Information Science and Technology*, 17(3):252–264, 2014.

[33] B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes. A binary firefly algorithm for the set covering problem. In *3rd Computer Science On-line Conference 2014, Modern Trends and Techniques in Computer Science*, volume 285, pages 65–73. Springer, 2014.

[34] B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes. A new approach using a binary firefly algorithm for the set covering problem. In *1st International Conference on Electrical Engineering and Information Technology (CEEIT)*, volume 63, pages 51–56. WIT press, 2014.

[35] B. Crawford, R. Soto, M. Olivares-Suárez, and F. Paredes. Using the firefly optimization method to solve the weighted set covering problem. In *HCI International 2014 - Posters' Extended Abstracts - International Conference, HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings, Part I*, volume 434 of *Communications in Computer and Information Science*, pages 509–514. Springer, 2014.

[36] B. Crawford, R. Soto, W. Palma, F. Johnson, F. Paredes, and E. Olguín. A 2-level approach for the set covering problem: Parameter tuning of artificial bee colony algorithm by using genetic algorithm. In Y. Tan, Y. Shi, and C. A. Coello, editors, *Advances in Swarm Intelligence - 5th International Conference, ICSI*, volume 8794 of *LNCS*, pages 189–196, Hefei, China, 2014. Springer.

[37] B. Crawford, R. Soto, C. Peña, W. Palma, F. Johnson, and F. Paredes. Solving the set covering problem with a shuffled frog leaping algorithm. In *Intelligent Information and Database Systems*, volume 9012, pages 41–50, 2015.

[38] Broderick Crawford, Ricardo Soto, Felipe Aballay, Sanjay Misra, Franklin Johnson, and Fernando Paredes. A teaching-learning-based optimization algorithm for solving set covering problems. In *Computational Science and Its Applications–ICCSA 2015*, pages 421–430. 2015.

[39] Broderick Crawford, Ricardo Soto, Natalia Berrios, Eduardo Olguín, and Sanjay Misra. Cat swarm optimization with different transfer functions for solving set covering problems, 2016.

[40] Broderick Crawford, Ricardo Soto, Cristian Peña, Marco Riquelme-Leiva, Claudio Torres-Rojas, Franklin Johnson, and Fernando Paredes. Binarization methods for shuffled frog leaping algorithms that solve set covering problems. In *Software Engineering in Intelligent Systems*, pages 317–326. 2015.

[41] Broderick Crawford, Ricardo Soto, Claudio Torres-Rojas, Cristian Peña, Marco Riquelme-Leiva, Sanjay Misra, Franklin Johnson, and Fernando

Paredes. A binary fruit fly optimization algorithm to solve the set covering problem. In *Computational Science and Its Applications–ICCSA 2015*, pages 411–420. 2015.

[42] S. Crowell-Davis. Cat behaviour: social organization, communication and development. In I. Rochlitz, editor, *The Welfare Of Cats*, pages 1–22. Springer Netherlands, 2005.

[43] R. Cuesta, B. Crawford, R. Soto, and F. Paredes. An artificial bee colony algorithm for the set covering problem. In *3rd Computer Science On-line Conference 2014 (CSOC), Modern Trends and Techniques in Computer Science*, volume 285 of *AISC*, pages 53–63. Springer, 2014.

[44] J. Dards. Feral cat behaviour and ecology. *Bulletin of the Feline Advisory Bureau*, 15, 1976.

[45] R. H. Day. Letter to the editorâĂŤon optimal extracting from a multiple file data storage system: An application of integer programming. *Operations Research*, 13(3):482–494, 1965.

[46] M. Desrochers and F. Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1):1–13, 1989.

[47] Muzaffar Eusuff, Kevin Lansey, and Fayzul Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2):129–154, 2006.

[48] M. L. Fisher and P. Kedia. Optimal solution of set covering/partitioning problems using dual heuristics. *Management Science*, 36(6):674–688, 1990.

[49] M. L. Fisher and M. B. Rosenwein. An interactive optimization system for bulk-cargo ship scheduling. *Naval Research Logistics (NRL)*, 36(1):27–42, 1989.

[50] B. A. Foster and D. M. Ryan. An integer programming approach to the vehicle scheduling problem. *Operations Research*, 27:367–384, 1976.

[51] B. A. Freeman and J. V. Jucker. The line balancing problem. *Journal of Industrial Engineering*, 18:361–364, 1967.

[52] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co, 1979.

[53] R. S. Garfinkel and G. L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B495–B508, 1970.

[54] D. Gouwanda and S. Ponnambalam. Evolutionary search techniques to solve set covering problems. *World Academy of Science, Engineering and Technology*, 39:20–25, 2008.

[55] W. Hays and R. Winkler. *Statistics: probability, inference, and decision.* Holt, Rinehart and Winston., 1970.

[56] E. Housos and T. Elmroth. Automatic optimization of subproblems in scheduling airline crews. *Interfaces*, 27(5):68–77, 1997.

[57] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39(3):459–471, 2007.

[58] Richard M Karp. *Reducibility among combinatorial problems.* Springer, 1972.

[59] J. Kennedy and R. C. Eberhart. In *Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, volume 5, pages 4104–4108, 1997.

[60] L. Lessing, I. Dumitrescu, and T. Stützle. A comparison between aco algorithms for the set covering problem. In *Ant Colony Optimization and Swarm Intelligence*, pages 1–12, 2004.

[61] H. W. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62:399–402, 1967.

[62] S. Mirjalili and A. Lewis. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9(0):1 – 14, 2013.

[63] B. Pallaud. Hypotheses on mechanisms underlying observational learning in animals. *Behavioural Processes*, 9:381–394, 1984.

[64] Wen-Tsao Pan. A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, 26:69–74, 2012.

[65] G. Panda, P. Pradhan, and B. Majhi. Iir system identification using cat swarm optimization. *Expert Systems with Applications*, 38:12671–12683, 2011.

[66] Ravipudi V Rao, Vimal J Savsani, and DP Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3):303–315, 2011.

[67] J. A. Reggia, D. S. Nau, and P. Y Wang. Diagnostic expert systems based on a set covering model. *International Journal of Man-Machine Studies*, 19:437–460, 1983.

[68] Z. Ren, Z. Feng, L. Ke, and Z. Zhang. New ideas for applying ant colony optimization to the set covering problem. *Computers & Industrial Engineering*, pages 774 – 784, 2010.

[69] C. C. Ribeiro, M. Minoux, and M. C. Penna. An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment. *European Journal of Operational Research*, 41(2):232–239, 1989.

[70] J. Rubin. A technique for the solution of massive set covering problems, with application to airline crew scheduling. *Transportation Science*, 7(1):34–48, 1973.

[71] S. K. Sahaa, S. P. Ghoshalb, R. Kara, and D. Mandala. Cat swarm optimization algorithm for optimal linear phase fir filter design. *ISA Transactions*, 52(6):781–794, 2013.

[72] M. E. Salveson. The assembly line balancing problem. *Journal of Industrial Engineering*, 6:18–25, 1955.

[73] B. Santosa and M. K. Ningrum. Cat swarm optimization for clustering. *2009 International Conference of Soft Computing and Pattern Recognition*, pages 54–59, 2009.

[74] S. S. Shapiro and M. B. Wilk. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 1:50–60, 1947.

[75] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52:591–611, 1965.

[76] Y. Sharafi, M. Khanesar, and M. Teshnehlab. Discrete binary cat swarm optimization algorithm. *Computer,Control and Communication*, pages 1–6, 2013.

[77] R. Shojaee, H.R. Faragardi, S. Alaee, and N. Yazdani. A new cat swarm optimization based algorithm for reliability-oriented task allocation in distributed systems. In *Sixth International Symposium on Telecommunications (IST)*, volume 192, pages 861–866, 2012.

[78] W. Sung. *Effect of gender on initiation of proximity in free ranging domestic cats (Felis catus)*. 1998.

[79] P. Tsai, J. Pan, S. Chen, and B. Liao. Enhanced parallel cat swarm optimization based on the taguchi method. *Expert Systems with Applications*, 39:6309–6319, 2012.

[80] C. Valenzuela, B. Crawford, R. Soto, E. Monfroy, and F. Paredes. A 2-level metaheuristic for the set covering problem. *International Journal of Computers Communications and Control*, 7(2):377–387, 2012.

[81] F. J. Vasko and G. R. Wilson. Using a facility location algorithm to solve large set covering problems. *Operations Research Letters*, 3(2):85–90, 1984.

[82] F. J. Vasko, F. E. Wolf, and K. L. Stott. Optimal selection of ingot sizes via set covering. *Operations Research*, 35(3):346–353, June 1987.

[83] F. J. Vasko, F. E. Wolf, and K. L. Stott. A set covering approach to metallurgical grade assignment. *European Journal of Operational Research*, 38(1):27–34, 1989.

[84] F. J. Vasko, F. E. Wolf, K. L. Stott, and J. W. Scheirer. Selecting optimal ingot sizes for bethlehem steel. *Interfaces*, 19(1):68–84, 1989.

[85] W. Walker. Using the set-covering problem to assign fire companies to fire houses. *Operations Research*, 22:275–277, 1974.

[86] Z. H. Wanga, C. C. Changb, and M. C. Lia. Optimizing least-significant-bit substitution using cat swarm optimization strategy. *Information Science*, 192:98–108, 2013.

[87] R. C. Wolfe. *The social organization of the free ranging domestic cat (Felis catus)*. 2001.

[88] A. Yamane, T. Doi, and Y. Ono. Mating behaviors, courtship rank and mating success of male feral cat (felis catus). *Journal of Ethology*, 14:35–44, 1996.

[89] Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2):78–84, 2010.

11 Appendix

11.1 Instance Result Tables

In this study we used 65 standard benchmark problems from the OR-Library. These intances were randomly generated non-unicost and have been widely used in literature. Moreover, we used eight transfer functions and five discretization techniques.

The following eight tables contain the resume results of the executed tests. The column Med reports the median statistics of the results, the Max and Min columns report the maximun and the minimum cost of the best solutions obtained in 30 runs respectively.

S1														
D1			D2			D3			D4			D5		
med	max	min	med	max	min	med	max	min	med	max	min	med	max	min
443.0	457.0	432.0	440.0	454.0	434.0	440.0	460.0	434.0	444.0	466.0	434.0	441.0	456.0	433.0
538.0	559.0	517.0	531.5	565.0	515.0	534.0	564.0	515.0	529.0	559.0	516.0	534.0	558.0	517.0
559.0	580.0	527.0	552.5	572.0	536.0	554.0	571.0	526.0	554.0	582.0	528.0	558.0	581.0	523.0
512.0	537.0	499.0	514.0	538.0	497.0	524.0	538.0	499.0	526.0	537.0	497.0	509.0	537.0	496.0
526.0	538.0	514.0	527.0	539.0	514.0	526.5	538.0	514.0	526.0	537.0	514.0	526.0	537.0	514.0
567.5	593.0	560.0	573.5	591.0	560.0	564.0	585.0	560.0	565.0	589.0	560.0	569.0	588.0	560.0
437.0	449.0	430.0	436.0	448.0	434.0	437.5	449.0	434.0	437.0	452.0	431.0	438.5	449.0	432.0
522.0	554.0	494.0	503.0	553.0	503.0	522.0	546.0	503.0	517.5	560.0	494.0	516.5	543.0	494.0
682.0	694.0	659.0	676.5	693.0	661.0	676.0	687.0	662.0	675.5	694.0	658.0	676.5	693.0	661.0
526.0	543.0	518.0	526.5	544.0	517.0	527.5	543.0	518.0	525.0	539.0	518.0	526.5	540.0	518.0
262.0	264.0	261.0	262.0	273.0	258.0	262.0	272.0	256.0	262.0	269.0	258.0	262.0	272.0	257.0
315.0	326.0	308.0	314.0	323.0	306.0	315.0	324.0	306.0	316.0	326.0	307.0	314.5	324.0	307.0
232.0	244.0	229.0	233.5	243.0	230.0	232.5	242.0	229.0	232.0	244.0	229.0	232.0	243.0	229.0
246.0	249.0	245.0	246.0	248.0	245.0	245.0	247.0	245.0	246.0	248.0	242.0	245.5	248.0	242.0
221.0	224.0	216.0	219.0	224.0	216.0	219.0	224.0	216.0	219.0	224.0	216.0	221.0	224.0	216.0
226.0	239.0	217.0	226.0	239.0	213.0	226.0	239.0	217.0	226.5	239.0	217.0	226.0	239.0	217.0
307.0	315.0	297.0	305.5	327.0	295.0	307.0	314.0	297.0	306.0	320.0	294.0	305.0	316.0	297.0
308.0	320.0	291.0	305.0	323.0	288.0	307.5	314.0	289.0	307.0	318.0	289.0	307.5	319.0	299.0
281.0	281.0	280.0	281.0	290.0	280.0	281.0	288.0	280.0	281.0	281.0	280.0	281.0	281.0	280.0
275.5	281.0	271.0	276.0	282.0	271.0	276.0	281.0	270.0	275.0	281.0	271.0	277.0	281.0	271.0
146.0	157.0	144.0	146.0	158.0	144.0	148.0	154.0	143.0	147.0	158.0	143.0	147.0	156.0	144.0
150.0	154.0	150.0	150.0	151.0	150.0	150.0	151.0	150.0	150.0	151.0	150.0	150.0	151.0	150.0
152.0	155.0	149.0	153.0	161.0	148.0	153.0	159.0	149.0	153.0	155.0	148.0	153.0	155.0	149.0
135.0	137.0	133.0	135.0	138.0	133.0	135.0	137.0	133.0	135.0	137.0	133.0	135.0	138.0	133.0
170.5	173.0	164.0	171.0	172.0	166.0	170.0	172.0	165.0	168.5	172.0	165.0	170.5	172.0	165.0
276.0	280.0	271.0	275.0	281.0	271.0	274.0	281.0	271.0	276.0	280.0	271.0	276.0	281.0	271.0
265.0	268.0	261.0	265.0	268.0	261.0	265.5	268.0	259.0	265.0	268.0	262.0	265.0	268.0	261.0
245.0	254.0	234.0	245.0	255.0	235.0	243.5	251.0	233.0	242.0	253.0	235.0	244.0	254.0	234.0
244.0	249.0	244.0	244.0	249.0	244.0	247.0	249.0	244.0	244.0	250.0	237.0	247.0	250.0	244.0
239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	236.0	239.0	240.0	237.0	239.0	240.0	237.0
74.5	78.0	72.0	75.0	78.0	73.0	74.0	79.0	70.0	74.0	79.0	70.0	74.0	77.0	70.0
85.0	88.0	80.0	85.0	88.0	80.0	85.0	88.0	81.0	85.0	88.0	81.0	85.5	88.0	80.0
83.0	85.0	80.0	83.0	85.0	80.0	83.0	86.0	80.0	83.0	85.0	80.0	83.0	86.0	80.0
84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	237.0	232.0	235.0	239.0	231.0	235.0	238.0	231.0	235.0	238.0	233.0	235.0	235.0	231.0
231.0	239.0	224.0	231.0	237.0	224.0	230.5	235.0	224.0	230.0	234.0	226.0	231.0	238.0	223.0
265.0	276.0	257.0	264.0	276.0	251.0	265.0	277.0	256.0	264.0	276.0	257.0	264.5	275.0	258.0
239.5	245.0	229.0	238.0	246.0	229.0	235.0	246.0	231.0	239.0	246.0	231.0	239.0	245.0	231.0
232.0	234.0	222.0	232.0	233.0	224.0	232.0	235.0	222.0	232.0	236.0	224.0	232.0	233.0	224.0
65.0	65.0	65.0	65.0	66.0	64.0	65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	63.0
70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0
79.0	80.0	78.0	79.0	79.0	76.0	79.0	79.0	76.0	79.0	80.0	78.0	79.0	79.0	78.0
66.0	67.0	64.0	66.5	67.0	63.0	66.0	67.0	64.0	66.5	67.0	64.0	66.0	67.0	64.0
65.0	66.0	64.0	65.0	66.0	63.0	65.0	66.0	63.0	65.0	66.0	63.0	65.0	66.0	64.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
34.0	35.0	34.0	34.5	35.0	34.0	34.0	35.0	32.0	35.0	35.0	32.0	34.5	35.0	34.0
34.0	34.0	32.0	34.0	34.0	31.0	32.0	34.0	31.0	34.0	34.0	32.0	34.0	34.0	30.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	16.0	18.0	18.0	18.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	17.0	17.0	18.0	17.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
193.0	194.0	191.0	195.0	195.0	191.0	194.0	195.0	191.0	194.0	195.0	189.0	194.5	195.0	192.0
168.0	169.0	164.0	167.0	169.0	166.0	167.0	169.0	164.0	167.0	169.0	165.0	167.0	169.0	165.0
183.0	183.0	180.0	183.0	183.0	181.0	183.0	183.0	181.0	183.0	183.0	179.0	182.5	183.0	180.0
184.0	184.0	182.0	184.0	184.0	181.0	184.0	184.0	181.0	184.0	185.0	181.0	184.0	184.0	182.0
184.5	186.0	184.0	184.0	186.0	182.0	184.0	186.0	183.0	185.0	186.0	181.0	184.0	186.0	181.0
73.0	73.0	71.0	73.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	71.0	73.0	74.0	70.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	68.0	65.0	67.0	68.0	66.0	67.0	68.0	66.0	67.0	68.0	66.0	67.0	68.0	66.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 9: S_1 with all discretization functions

$\mathbf{S_2}$														
$\mathbf{D_1}$			$\mathbf{D_2}$			$\mathbf{D_3}$			$\mathbf{D_4}$			$\mathbf{D_5}$		
med	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**
439.5	456.0	432.0	456.5	466.0	439.0	457.0	468.0	441.0	440.0	452.0	432.0	441.0	459.0	434.0
535.0	557.0	518.0	559.0	574.0	525.0	539.0	584.0	517.0	535.5	557.0	517.0	530.0	566.0	515.0
557.0	580.0	526.0	579.5	595.0	555.0	557.0	579.0	525.0	552.0	584.0	525.0	557.0	581.0	523.0
505.0	537.0	496.0	526.0	541.0	499.0	511.0	537.0	496.0	513.0	537.0	496.0	509.5	542.0	497.0
526.0	539.0	514.0	534.0	541.0	524.0	527.0	537.0	514.0	526.0	539.0	514.0	523.5	538.0	514.0
569.0	592.0	560.0	586.0	604.0	560.0	567.0	586.0	560.0	564.5	589.0	560.0	566.0	591.0	560.0
441.5	450.0	434.0	447.0	457.0	435.0	440.5	450.0	430.0	439.5	455.0	432.0	441.0	451.0	434.0
517.5	551.0	494.0	538.5	563.0	503.0	518.5	551.0	503.0	522.0	543.0	494.0	503.0	553.0	494.0
677.0	700.0	658.0	690.5	725.0	673.0	681.5	699.0	658.0	676.0	689.0	662.0	680.5	697.0	665.0
526.5	551.0	518.0	542.0	551.0	527.0	527.0	545.0	518.0	525.0	546.0	518.0	530.0	550.0	517.0
262.0	272.0	257.0	270.5	273.0	262.0	262.5	266.0	260.0	262.0	264.0	259.0	263.0	267.0	257.0
314.5	329.0	306.0	322.5	331.0	311.0	317.5	325.0	308.0	314.5	324.0	306.0	313.0	322.0	306.0
233.0	243.0	230.0	242.0	245.0	231.0	232.5	243.0	229.0	232.0	242.0	229.0	232.0	242.0	230.0
245.5	247.0	242.0	246.0	249.0	245.0	245.0	248.0	245.0	245.5	248.0	244.0	245.0	248.0	245.0
220.0	224.0	216.0	224.0	226.0	218.0	220.5	224.0	216.0	221.0	224.0	216.0	221.0	224.0	216.0
226.0	235.0	217.0	239.0	241.0	227.0	226.0	235.0	218.0	226.0	238.0	218.0	225.0	235.0	217.0
307.0	317.0	293.0	320.0	335.0	300.0	304.5	319.0	294.0	306.0	315.0	297.0	306.5	321.0	297.0
308.0	324.0	293.0	316.0	329.0	306.0	306.0	316.0	294.0	308.5	318.0	296.0	308.5	323.0	290.0
281.0	281.0	280.0	281.0	289.0	280.0	281.0	289.0	280.0	281.0	281.0	280.0	280.0	281.0	280.0
274.0	283.0	271.0	281.0	283.0	274.0	275.5	281.0	271.0	276.0	281.0	271.0	277.0	281.0	271.0
146.0	159.0	143.0	155.0	165.0	148.0	146.0	156.0	144.0	148.0	156.0	143.0	147.5	158.0	143.0
150.0	150.0	150.0	155.0	159.0	146.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0
151.5	155.0	149.0	155.0	159.0	151.0	152.5	155.0	148.0	153.0	155.0	148.0	152.0	155.0	148.0
135.0	137.0	133.0	136.0	138.0	134.0	134.5	137.0	133.0	135.0	138.0	133.0	135.0	137.0	133.0
170.0	172.0	165.0	171.0	171.0	171.0	169.5	173.0	165.0	171.0	172.0	165.0	171.0	172.0	165.0
276.0	281.0	271.0	281.0	283.0	274.0	276.0	281.0	271.0	276.0	281.0	271.0	277.0	281.0	273.0
265.5	269.0	260.0	268.0	273.0	263.0	265.0	268.0	262.0	265.0	269.0	260.0	265.0	268.0	262.0
243.5	253.0	235.0	252.0	258.0	244.0	244.5	253.0	236.0	243.0	254.0	234.0	242.5	252.0	237.0
244.0	250.0	241.0	249.0	253.0	241.0	244.0	253.0	244.0	244.0	249.0	241.0	244.0	249.0	241.0
239.0	240.0	237.0	240.0	242.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0
74.0	78.0	72.0	76.5	79.0	72.0	75.0	78.0	72.0	74.0	79.0	72.0	75.5	79.0	72.0
86.0	88.0	82.0	88.0	89.0	84.0	83.0	87.0	80.0	85.5	88.0	80.0	85.0	88.0	80.0
83.0	86.0	80.0	84.0	87.0	81.0	83.0	85.0	80.0	83.0	85.0	80.0	84.0	85.0	80.0
84.0	84.0	84.0	84.5	86.0	84.0	84.0	84.0	84.0	84.0	86.0	81.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	238.0	231.0	237.0	242.0	234.0	235.0	238.0	232.0	235.0	236.0	233.0	235.0	236.0	233.0
230.0	235.0	224.0	234.0	240.0	227.0	232.0	238.0	225.0	231.0	235.0	226.0	229.5	236.0	224.0
265.0	276.0	259.0	275.0	278.0	265.0	265.0	277.0	256.0	265.0	275.0	257.0	267.0	276.0	255.0
241.0	246.0	230.0	245.5	247.0	233.0	239.0	246.0	231.0	240.0	246.0	225.0	240.0	246.0	226.0
229.0	233.0	222.0	236.0	241.0	226.0	228.0	233.0	219.0	231.5	234.0	222.0	233.0	240.0	227.0
65.0	66.0	64.0	65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	64.0	65.0	66.0	61.0
70.0	70.0	69.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0
79.0	80.0	77.0	79.0	80.0	76.0	79.0	80.0	78.0	79.0	79.0	76.0	79.0	79.0	79.0
66.0	67.0	64.0	66.0	67.0	64.0	66.0	67.0	63.0	66.0	67.0	64.0	66.0	67.0	64.0
65.0	66.0	63.0	65.0	66.0	63.0	65.0	66.0	63.0	65.0	66.0	64.0	65.0	66.0	63.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
34.0	35.0	34.0	34.5	35.0	34.0	35.0	35.0	34.0	35.0	35.0	34.0	34.5	35.0	34.0
34.0	34.0	32.0	34.0	34.0	32.0	34.0	34.0	30.0	34.0	34.0	32.0	34.0	34.0	32.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	17.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	16.0	17.5	18.0	16.0	18.0	18.0	16.0	17.0	17.0	17.0	17.0	18.0	16.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
193.0	194.0	192.0	194.0	195.0	192.0	194.0	195.0	193.0	194.0	195.0	193.0	195.0	195.0	193.0
167.0	169.0	166.0	166.5	168.0	166.0	167.5	169.0	165.0	167.0	169.0	165.0	167.5	169.0	166.0
183.0	183.0	182.0	182.0	182.0	182.0	183.0	183.0	180.0	183.0	183.0	181.0	183.0	183.0	179.0
184.0	184.0	182.0	184.0	184.0	178.0	184.0	184.0	180.0	184.0	184.0	182.0	184.0	184.0	181.0
184.0	186.0	183.0	184.0	186.0	183.0	184.0	186.0	181.0	184.0	186.0	183.0	184.5	186.0	183.0
73.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	71.0	73.0	73.0	70.0	73.0	73.0	73.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	68.0	66.0	67.0	67.0	67.0	67.0	68.0	66.0	67.0	67.0	67.0	67.0	68.0	64.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 10: S_2 with all discretization functions

S_3														
D_1			D_2			D_3			D_4			D_5		
med	max	min	med	max	min	med	max	min	med	max	min	med	max	min
440.0	457.0	432.0	457.0	466.0	446.0	457.0	471.0	440.0	441.0	459.0	432.0	440.0	456.0	434.0
532.0	560.0	515.0	559.5	574.0	522.0	538.0	569.0	520.0	537.0	570.0	518.0	538.5	560.0	513.0
554.5	581.0	520.0	578.5	591.0	551.0	556.5	574.0	538.0	553.0	580.0	532.0	556.0	578.0	539.0
513.0	542.0	497.0	526.0	544.0	515.0	507.0	534.0	496.0	508.0	534.0	497.0	513.0	541.0	497.0
526.5	537.0	517.0	537.0	540.0	528.0	527.0	537.0	514.0	526.0	539.0	514.0	523.5	538.0	514.0
569.5	599.0	560.0	585.5	618.0	560.0	569.0	587.0	560.0	565.0	586.0	560.0	569.5	591.0	560.0
438.5	450.0	434.0	449.0	456.0	434.0	439.5	450.0	433.0	437.0	450.0	431.0	442.0	449.0	430.0
512.5	551.0	494.0	535.0	555.0	503.0	503.0	554.0	494.0	522.0	537.0	494.0	522.0	537.0	503.0
675.0	690.0	657.0	690.5	726.0	668.0	682.0	694.0	663.0	676.5	694.0	660.0	675.0	692.0	664.0
526.5	543.0	518.0	537.5	551.0	520.0	525.5	543.0	518.0	523.0	544.0	518.0	525.0	550.0	518.0
262.0	272.0	259.0	272.0	274.0	262.0	263.0	272.0	257.0	262.0	264.0	261.0	262.0	272.0	259.0
313.0	326.0	305.0	323.0	329.0	311.0	314.5	323.0	307.0	314.5	327.0	308.0	313.0	322.0	306.0
231.5	244.0	230.0	242.0	244.0	231.0	232.0	244.0	229.0	232.0	242.0	229.0	232.0	242.0	229.0
246.0	247.0	242.0	247.0	250.0	244.0	245.0	249.0	242.0	246.0	248.0	242.0	246.0	248.0	242.0
219.0	224.0	216.0	223.0	224.0	218.0	221.0	224.0	216.0	220.0	224.0	216.0	221.0	224.0	216.0
226.0	239.0	217.0	237.0	240.0	226.0	226.0	237.0	217.0	226.0	238.0	218.0	226.0	227.0	223.0
303.5	313.0	294.0	320.5	339.0	308.0	302.5	317.0	295.0	305.5	318.0	299.0	305.0	318.0	296.0
307.5	319.0	300.0	322.5	329.0	306.0	307.0	315.0	299.0	307.0	318.0	294.0	307.0	314.0	299.0
281.0	281.0	281.0	283.0	292.0	280.0	281.0	281.0	281.0	281.0	291.0	280.0	281.0	287.0	280.0
276.0	281.0	271.0	279.5	283.0	276.0	276.0	281.0	271.0	276.0	282.0	271.0	275.5	282.0	271.0
146.0	153.0	143.0	153.5	163.0	146.0	149.0	157.0	143.0	148.0	158.0	143.0	146.0	156.0	143.0
150.0	150.0	150.0	155.0	158.0	146.0	150.0	150.0	150.0	150.0	153.0	150.0	150.0	154.0	150.0
152.5	158.0	149.0	155.0	155.0	155.0	152.5	159.0	149.0	153.0	159.0	149.0	152.0	159.0	149.0
135.0	137.0	132.0	136.0	138.0	134.0	135.0	137.0	133.0	135.0	138.0	133.0	135.0	137.0	133.0
171.0	172.0	164.0	171.0	173.0	167.0	171.0	172.0	165.0	171.0	171.0	165.0	170.0	172.0	164.0
274.5	280.0	271.0	281.0	283.0	274.0	274.0	280.0	271.0	277.0	281.0	271.0	277.0	281.0	271.0
265.0	268.0	259.0	268.0	271.0	264.0	265.0	268.0	261.0	265.0	269.0	261.0	265.0	269.0	261.0
245.5	254.0	234.0	250.0	259.0	242.0	243.5	253.0	235.0	243.0	252.0	233.0	244.0	251.0	237.0
244.5	249.0	241.0	249.0	255.0	241.0	247.0	250.0	241.0	244.0	249.0	244.0	244.0	249.0	241.0
239.0	240.0	237.0	239.0	241.0	237.0	239.0	241.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0
75.0	77.0	72.0	78.0	79.0	73.0	74.0	78.0	72.0	74.0	78.0	71.0	75.0	78.0	70.0
85.0	88.0	81.0	88.0	89.0	85.0	85.5	88.0	81.0	85.0	88.0	80.0	85.0	88.0	80.0
83.0	85.0	80.0	85.0	87.0	83.0	83.0	86.0	80.0	83.0	86.0	80.0	82.0	85.0	80.0
84.0	84.0	84.0	85.0	87.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	236.0	231.0	238.0	242.0	235.0	235.0	238.0	233.0	235.0	238.0	233.0	235.0	237.0	233.0
229.0	239.0	225.0	234.0	240.0	227.0	230.5	234.0	224.0	230.5	237.0	225.0	231.0	238.0	225.0
264.0	277.0	255.0	274.0	278.0	261.0	266.0	274.0	254.0	265.0	276.0	257.0	264.0	275.0	254.0
245.0	247.0	238.0	245.5	247.0	234.0	240.0	246.0	232.0	240.0	246.0	230.0	238.5	244.0	229.0
235.5	242.0	228.0	235.0	242.0	228.0	232.5	234.0	226.0	232.0	235.0	223.0	232.0	234.0	221.0
66.0	66.0	64.0	65.0	66.0	61.0	65.0	65.0	65.0	65.0	66.0	64.0	65.0	65.0	65.0
70.0	70.0	70.0	70.0	70.0	69.0	70.0	70.0	70.0	70.0	70.0	69.0	70.0	70.0	70.0
79.0	81.0	78.0	79.0	79.0	79.0	79.0	79.0	79.0	79.0	79.0	79.0	79.0	79.0	79.0
67.0	68.0	64.0	66.0	67.0	63.0	66.0	67.0	63.0	66.0	67.0	63.0	66.0	67.0	64.0
66.0	66.0	65.0	65.0	66.0	64.0	66.0	66.0	64.0	65.0	66.0	64.0	65.0	66.0	64.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
35.0	35.0	34.0	34.5	35.0	34.0	34.0	35.0	34.0	35.0	35.0	34.0	34.0	35.0	34.0
34.0	34.0	32.0	32.0	34.0	32.0	34.0	34.0	31.0	32.5	34.0	32.0	34.0	34.0	32.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0	17.5	18.0	16.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
193.5	194.0	190.0	194.0	195.0	192.0	194.0	195.0	193.0	195.0	195.0	193.0	194.0	195.0	192.0
167.0	167.0	165.0	167.0	167.0	166.0	167.0	170.0	165.0	167.5	169.0	165.0	167.5	169.0	164.0
182.0	182.0	182.0	182.0	182.0	182.0	183.0	183.0	181.0	183.0	183.0	182.0	183.0	183.0	180.0
184.0	184.0	178.0	184.0	184.0	178.0	184.0	184.0	184.0	184.0	184.0	182.0	184.0	184.0	182.0
183.0	183.0	183.0	185.0	186.0	183.0	184.0	186.0	182.0	184.0	187.0	183.0	184.5	186.0	181.0
71.0	73.0	69.0	73.0	73.0	70.0	73.0	74.0	70.0	73.0	73.0	70.0	73.0	73.0	70.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	67.0	67.0	67.0	68.0	66.0	67.0	68.0	66.0	67.0	67.0	67.0	67.0	68.0	65.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 11: S_3 with all discretization functions

S_4														
D_1			D_2			D_3			D_4			D_5		
med	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**
440.5	465.0	432.0	455.5	465.0	439.0	444.0	465.0	434.0	444.5	458.0	432.0	440.0	461.0	434.0
535.5	555.0	516.0	554.5	574.0	533.0	536.5	560.0	516.0	530.0	564.0	517.0	530.5	557.0	516.0
554.5	579.0	535.0	573.0	591.0	538.0	556.5	580.0	535.0	564.0	583.0	522.0	554.0	581.0	536.0
514.5	539.0	497.0	528.5	541.0	503.0	508.5	540.0	497.0	510.5	538.0	499.0	507.0	541.0	496.0
523.5	537.0	514.0	537.5	540.0	519.0	526.0	538.0	514.0	523.5	539.0	514.0	525.0	538.0	514.0
568.5	600.0	560.0	587.0	621.0	564.0	564.5	602.0	560.0	564.5	591.0	560.0	564.5	581.0	560.0
439.0	449.0	434.0	449.0	458.0	435.0	437.0	451.0	431.0	437.0	450.0	431.0	437.5	446.0	434.0
522.0	560.0	494.0	534.5	563.0	503.0	522.0	554.0	494.0	522.0	553.0	492.0	522.0	551.0	503.0
676.0	690.0	660.0	683.0	711.0	661.0	676.0	694.0	660.0	675.5	699.0	654.0	675.0	689.0	661.0
528.0	544.0	518.0	542.0	551.0	520.0	527.0	541.0	518.0	526.0	544.0	518.0	525.0	542.0	518.0
262.5	272.0	260.0	265.0	276.0	262.0	262.0	263.0	261.0	262.0	266.0	260.0	262.5	264.0	260.0
315.0	327.0	308.0	322.5	330.0	310.0	314.5	323.0	307.0	314.5	322.0	306.0	315.5	324.0	306.0
232.0	244.0	228.0	242.0	245.0	231.0	232.0	244.0	226.0	233.0	244.0	229.0	232.0	244.0	229.0
246.0	248.0	242.0	246.0	249.0	245.0	245.0	248.0	242.0	246.0	248.0	242.0	245.0	249.0	242.0
221.0	224.0	216.0	224.0	224.0	218.0	221.0	224.0	216.0	221.0	224.0	216.0	219.0	224.0	216.0
226.0	238.0	217.0	238.0	243.0	223.0	226.0	239.0	215.0	225.0	236.0	213.0	224.0	227.0	217.0
307.0	322.0	298.0	320.0	338.0	301.0	306.5	318.0	297.0	304.5	317.0	294.0	307.0	320.0	296.0
305.0	318.0	290.0	318.5	329.0	311.0	306.0	313.0	300.0	308.5	323.0	300.0	309.0	317.0	294.0
281.0	281.0	280.0	282.0	291.0	281.0	281.0	281.0	280.0	281.0	281.0	280.0	281.0	289.0	280.0
274.0	281.0	271.0	279.5	283.0	269.0	274.0	280.0	271.0	276.0	281.0	271.0	274.0	281.0	273.0
146.0	158.0	143.0	154.0	164.0	149.0	149.0	154.0	143.0	147.5	156.0	144.0	146.0	152.0	143.0
150.0	150.0	150.0	155.0	158.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0	150.0	151.0	150.0
151.0	156.0	148.0	155.0	161.0	149.0	154.0	157.0	149.0	152.0	155.0	148.0	153.0	162.0	150.0
135.0	137.0	133.0	136.0	140.0	133.0	135.0	137.0	133.0	135.0	137.0	133.0	135.0	138.0	133.0
169.5	172.0	164.0	171.0	174.0	167.0	171.0	171.0	165.0	170.0	173.0	164.0	169.0	172.0	165.0
276.0	280.0	271.0	281.0	283.0	273.0	276.0	281.0	271.0	275.5	281.0	271.0	276.5	280.0	271.0
265.0	268.0	262.0	268.0	271.0	264.0	265.0	268.0	261.0	265.0	268.0	261.0	265.0	269.0	263.0
244.5	257.0	236.0	249.5	258.0	235.0	245.0	250.0	234.0	243.0	254.0	236.0	244.5	254.0	236.0
244.0	250.0	241.0	249.0	249.0	249.0	247.0	253.0	241.0	244.0	249.0	244.0	244.0	249.0	244.0
239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0
74.0	78.0	71.0	77.0	79.0	73.0	74.0	77.0	72.0	74.0	77.0	72.0	74.0	78.0	71.0
84.0	88.0	81.0	87.0	89.0	83.0	85.0	88.0	83.0	85.0	88.0	81.0	85.0	88.0	81.0
83.5	86.0	80.0	85.0	87.0	81.0	83.0	86.0	80.0	84.0	86.0	80.0	82.0	86.0	80.0
84.0	84.0	84.0	84.0	87.0	81.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	86.0	81.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	235.0	231.0	238.0	242.0	233.0	235.0	237.0	232.0	235.0	238.0	233.0	235.0	238.0	231.0
227.5	234.0	225.0	234.0	240.0	227.0	230.0	236.0	224.0	230.0	235.0	224.0	230.0	235.0	224.0
264.0	273.0	256.0	275.0	278.0	264.0	267.0	275.0	257.0	265.5	276.0	259.0	266.0	276.0	253.0
240.0	246.0	230.0	244.0	247.0	236.0	239.0	246.0	232.0	241.0	246.0	231.0	238.5	246.0	226.0
230.5	233.0	222.0	233.0	242.0	227.0	229.0	233.0	226.0	230.5	239.0	224.0	233.0	233.0	224.0
65.0	66.0	61.0	65.0	66.0	61.0	65.0	65.0	65.0	65.0	66.0	60.0	65.0	66.0	64.0
70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	69.0
79.0	79.0	78.0	79.0	80.0	76.0	79.0	80.0	76.0	79.0	79.0	76.0	79.0	80.0	76.0
66.0	67.0	63.0	66.0	67.0	64.0	66.0	67.0	64.0	66.0	67.0	63.0	66.0	67.0	64.0
65.0	66.0	63.0	65.5	66.0	64.0	65.0	66.0	63.0	65.0	66.0	64.0	65.0	66.0	64.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
35.0	35.0	34.0	34.0	35.0	34.0	34.5	35.0	34.0	35.0	35.0	34.0	35.0	35.0	34.0
34.0	34.0	32.0	34.0	34.0	31.0	34.0	34.0	32.0	34.0	34.0	32.0	34.0	34.0	32.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
194.0	195.0	193.0	195.0	195.0	193.0	194.0	195.0	193.0	194.0	195.0	191.0	194.0	195.0	192.0
167.0	169.0	165.0	167.0	169.0	165.0	167.0	169.0	166.0	167.5	169.0	165.0	167.0	170.0	166.0
183.0	184.0	181.0	182.5	183.0	182.0	182.0	183.0	182.0	182.0	183.0	181.0	183.0	183.0	180.0
184.0	184.0	181.0	184.0	184.0	182.0	184.0	184.0	182.0	184.0	184.0	184.0	184.0	184.0	182.0
184.0	186.0	183.0	184.0	186.0	183.0	185.0	186.0	183.0	184.0	186.0	182.0	184.0	186.0	183.0
73.0	73.0	71.0	73.0	73.0	70.0	72.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	71.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	68.0	66.0	67.0	67.0	67.0	67.0	68.0	65.0	67.0	68.0	66.0	67.0	68.0	64.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 12: S_4 with all discretization functions

$\mathbf{V_1}$														
$\mathbf{D_1}$			$\mathbf{D_2}$			$\mathbf{D_3}$			$\mathbf{D_4}$			$\mathbf{D_5}$		
med	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**
440.0	450.0	434.0	444.0	458.0	432.0	440.0	458.0	434.0	442.0	456.0	432.0	441.5	455.0	432.0
533.0	567.0	514.0	531.5	563.0	516.0	534.0	563.0	516.0	529.0	559.0	517.0	533.5	555.0	518.0
560.0	586.0	532.0	557.0	574.0	535.0	555.5	580.0	536.0	554.0	576.0	543.0	556.0	578.0	522.0
521.5	541.0	499.0	512.5	535.0	495.0	511.5	534.0	498.0	521.5	542.0	499.0	510.5	537.0	496.0
525.0	538.0	514.0	526.0	539.0	516.0	526.5	539.0	515.0	526.0	540.0	514.0	525.5	540.0	514.0
565.0	597.0	560.0	565.0	587.0	560.0	570.0	589.0	560.0	568.0	591.0	560.0	565.0	586.0	560.0
441.5	450.0	434.0	440.0	449.0	434.0	437.0	455.0	430.0	437.0	450.0	434.0	441.0	455.0	433.0
519.0	554.0	503.0	522.0	553.0	503.0	522.0	553.0	503.0	522.0	553.0	503.0	522.0	553.0	503.0
675.5	691.0	661.0	678.5	693.0	660.0	677.0	703.0	660.0	675.5	688.0	660.0	676.0	688.0	665.0
526.0	544.0	518.0	527.0	545.0	518.0	527.0	538.0	518.0	520.0	540.0	517.0	527.0	549.0	518.0
262.0	264.0	258.0	262.0	270.0	257.0	262.0	264.0	260.0	263.0	267.0	256.0	263.0	272.0	261.0
313.5	323.0	306.0	314.5	324.0	309.0	316.0	324.0	309.0	313.0	324.0	305.0	314.0	322.0	304.0
232.0	242.0	229.0	232.0	243.0	229.0	232.0	243.0	229.0	232.0	244.0	229.0	232.0	246.0	229.0
245.0	248.0	245.0	246.0	249.0	242.0	245.5	248.0	242.0	245.0	249.0	245.0	245.5	249.0	242.0
219.0	224.0	216.0	219.0	224.0	216.0	221.0	224.0	216.0	221.0	224.0	216.0	221.0	224.0	216.0
226.0	243.0	217.0	226.0	235.0	217.0	224.0	227.0	217.0	226.0	238.0	217.0	226.0	238.0	217.0
305.5	317.0	293.0	305.5	323.0	293.0	307.0	319.0	297.0	303.0	325.0	296.0	307.5	320.0	298.0
310.0	325.0	299.0	310.0	320.0	294.0	307.5	321.0	299.0	305.5	318.0	299.0	305.5	316.0	299.0
281.0	281.0	281.0	281.0	287.0	280.0	281.0	290.0	280.0	281.0	281.0	280.0	281.0	281.0	280.0
276.0	281.0	271.0	277.0	281.0	271.0	276.0	281.0	271.0	276.5	283.0	268.0	277.0	281.0	271.0
146.0	159.0	144.0	148.0	153.0	143.0	147.0	154.0	143.0	146.0	156.0	144.0	146.5	154.0	143.0
150.0	150.0	150.0	150.0	150.0	150.0	150.0	151.0	150.0	150.0	151.0	150.0	150.0	151.0	150.0
153.0	162.0	148.0	151.5	159.0	149.0	153.0	155.0	149.0	152.0	156.0	149.0	152.5	158.0	151.0
135.0	138.0	133.0	135.0	138.0	133.0	135.0	137.0	133.0	135.0	138.0	133.0	135.0	135.0	135.0
170.0	172.0	164.0	171.0	172.0	164.0	169.5	172.0	165.0	171.0	172.0	165.0	168.0	172.0	165.0
274.0	278.0	271.0	274.0	279.0	271.0	276.5	281.0	269.0	276.0	282.0	271.0	276.5	281.0	271.0
265.0	268.0	262.0	265.0	268.0	262.0	265.5	269.0	263.0	265.0	269.0	261.0	265.0	268.0	261.0
247.0	254.0	235.0	241.5	253.0	236.0	244.0	254.0	234.0	245.0	253.0	234.0	244.0	255.0	235.0
244.0	249.0	241.0	247.0	249.0	241.0	244.0	249.0	244.0	244.0	250.0	241.0	244.0	250.0	241.0
239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0
74.0	78.0	72.0	74.0	77.0	70.0	74.0	78.0	70.0	74.0	78.0	72.0	74.0	78.0	72.0
85.0	88.0	81.0	85.0	88.0	80.0	84.5	88.0	81.0	84.0	88.0	79.0	85.0	88.0	81.0
83.0	85.0	80.0	83.5	86.0	80.0	83.0	85.0	80.0	83.5	86.0	80.0	84.0	86.0	81.0
84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	238.0	232.0	235.0	235.0	235.0	235.0	236.0	233.0	234.5	239.0	233.0	235.0	235.0	232.0
230.0	235.0	224.0	231.0	235.0	221.0	231.5	234.0	226.0	228.5	239.0	225.0	231.0	235.0	224.0
265.5	277.0	255.0	265.5	276.0	257.0	265.0	274.0	258.0	266.0	276.0	253.0	265.0	275.0	258.0
240.0	246.0	230.0	239.0	246.0	232.0	240.5	246.0	232.0	241.0	246.0	230.0	240.0	246.0	232.0
228.0	233.0	222.0	231.0	233.0	221.0	229.0	239.0	222.0	228.0	239.0	222.0	232.0	234.0	222.0
65.0	66.0	64.0	65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	60.0	65.0	66.0	61.0
70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0
79.0	80.0	78.0	79.0	80.0	78.0	79.0	79.0	78.0	79.0	79.0	77.0	79.0	79.0	78.0
66.0	67.0	64.0	66.5	67.0	64.0	66.0	67.0	64.0	65.0	67.0	63.0	66.0	67.0	64.0
65.0	66.0	64.0	65.0	66.0	64.0	65.0	66.0	64.0	65.0	66.0	64.0	65.0	66.0	64.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
35.0	35.0	32.0	34.5	35.0	34.0	34.5	35.0	34.0	35.0	35.0	34.0	35.0	35.0	34.0
34.0	34.0	31.0	34.0	34.0	31.0	34.0	34.0	31.0	34.0	34.0	32.0	34.0	34.0	31.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
17.0	18.0	16.0	17.0	18.0	17.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
194.0	195.0	192.0	194.0	195.0	192.0	194.0	195.0	192.0	194.5	195.0	192.0	194.0	195.0	189.0
167.0	169.0	165.0	167.0	169.0	165.0	167.5	169.0	165.0	167.0	169.0	165.0	167.0	169.0	164.0
183.0	183.0	182.0	183.0	183.0	182.0	182.0	183.0	181.0	182.0	183.0	181.0	183.0	183.0	181.0
184.0	187.0	180.0	184.0	184.0	184.0	184.0	184.0	182.0	183.5	184.0	180.0	184.0	184.0	182.0
185.0	186.0	182.0	184.0	186.0	180.0	184.0	186.0	181.0	184.5	186.0	182.0	184.0	186.0	181.0
72.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	71.0	73.0	73.0	71.0	73.0	73.0	70.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	67.0	67.0	67.0	68.0	66.0	67.0	68.0	66.0	67.0	68.0	65.0	67.0	67.0	67.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 13: V_1 with all discretization functions

V_2														
D_1			D_2			D_3			D_4			D_5		
med	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**
441.0	464.0	432.0	441.0	454.0	434.0	440.0	459.0	432.0	442.0	459.0	432.0	440.0	456.0	432.0
536.5	555.0	519.0	534.0	559.0	519.0	536.0	557.0	517.0	528.0	555.0	517.0	532.0	555.0	519.0
553.5	578.0	539.0	554.0	580.0	523.0	552.0	584.0	523.0	558.0	590.0	542.0	558.5	581.0	521.0
510.5	538.0	497.0	511.5	537.0	496.0	509.0	541.0	497.0	508.0	537.0	497.0	511.0	538.0	496.0
526.0	537.0	514.0	527.0	539.0	516.0	526.5	537.0	512.0	523.0	539.0	514.0	528.0	538.0	514.0
568.0	591.0	560.0	565.0	597.0	560.0	564.0	580.0	560.0	566.0	586.0	560.0	563.0	591.0	560.0
439.0	454.0	434.0	440.5	449.0	434.0	438.5	455.0	434.0	441.5	455.0	434.0	436.0	447.0	434.0
522.0	551.0	503.0	522.0	554.0	494.0	511.5	537.0	494.0	512.5	553.0	494.0	522.0	553.0	503.0
676.0	693.0	664.0	675.0	685.0	665.0	678.0	694.0	661.0	678.5	696.0	661.0	676.5	709.0	657.0
530.0	545.0	518.0	527.0	546.0	518.0	527.0	540.0	518.0	527.0	543.0	518.0	527.0	546.0	518.0
262.0	271.0	261.0	263.0	272.0	257.0	262.0	266.0	260.0	262.5	269.0	260.0	262.0	272.0	257.0
315.5	326.0	306.0	314.0	324.0	309.0	315.0	327.0	308.0	313.0	326.0	308.0	312.5	324.0	307.0
232.0	243.0	229.0	233.0	243.0	229.0	231.5	242.0	229.0	232.0	242.0	230.0	232.0	242.0	229.0
245.5	248.0	242.0	246.0	248.0	242.0	245.0	248.0	242.0	245.5	248.0	243.0	245.0	247.0	242.0
219.0	224.0	218.0	221.0	224.0	216.0	219.0	222.0	216.0	221.0	224.0	216.0	221.0	224.0	216.0
225.5	235.0	213.0	226.0	239.0	217.0	226.0	239.0	217.0	223.5	240.0	216.0	226.0	239.0	213.0
302.5	315.0	294.0	304.5	320.0	294.0	305.0	315.0	296.0	304.5	321.0	297.0	308.0	328.0	300.0
305.5	320.0	300.0	305.0	315.0	294.0	307.0	319.0	300.0	305.0	320.0	290.0	307.0	325.0	293.0
281.0	282.0	280.0	281.0	281.0	281.0	281.0	292.0	280.0	281.0	284.0	280.0	281.0	290.0	280.0
277.0	281.0	271.0	274.5	281.0	271.0	276.0	282.0	271.0	275.0	281.0	271.0	276.0	283.0	271.0
147.0	162.0	143.0	145.0	152.0	143.0	146.0	156.0	143.0	146.0	156.0	144.0	148.0	158.0	144.0
150.0	150.0	150.0	150.0	153.0	148.0	150.0	153.0	150.0	150.0	153.0	150.0	150.0	150.0	150.0
152.5	157.0	149.0	152.0	155.0	149.0	153.0	155.0	149.0	151.5	155.0	149.0	153.0	155.0	148.0
135.0	137.0	133.0	135.0	138.0	133.0	135.0	138.0	133.0	135.0	138.0	133.0	135.0	136.0	133.0
170.0	172.0	165.0	170.0	172.0	165.0	171.0	173.0	165.0	171.0	172.0	165.0	170.5	172.0	165.0
277.0	281.0	271.0	276.0	281.0	271.0	276.0	280.0	271.0	276.0	280.0	273.0	275.0	282.0	271.0
265.0	268.0	261.0	265.0	269.0	262.0	265.0	268.0	261.0	266.0	268.0	263.0	265.0	268.0	260.0
243.0	255.0	235.0	242.5	253.0	236.0	244.0	252.0	235.0	244.5	252.0	235.0	242.5	254.0	235.0
244.0	249.0	241.0	247.0	253.0	237.0	245.5	250.0	241.0	244.0	251.0	241.0	244.5	250.0	241.0
239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0
74.0	78.0	70.0	74.0	78.0	70.0	74.0	79.0	72.0	75.5	78.0	72.0	74.0	78.0	70.0
85.0	89.0	80.0	84.5	88.0	80.0	85.0	88.0	79.0	85.0	88.0	82.0	85.0	87.0	80.0
83.0	86.0	80.0	83.0	86.0	80.0	83.0	86.0	80.0	83.0	86.0	80.0	83.0	86.0	81.0
84.0	84.0	84.0	84.0	88.0	81.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	236.0	231.0	235.0	235.0	233.0	235.0	235.0	232.0	235.0	238.0	231.0	235.0	235.0	231.0
229.0	235.0	225.0	231.0	239.0	224.0	229.5	234.0	225.0	231.0	238.0	224.0	231.5	234.0	221.0
266.0	276.0	259.0	267.0	274.0	259.0	266.0	275.0	253.0	265.0	276.0	259.0	264.5	274.0	256.0
238.0	244.0	232.0	240.0	246.0	232.0	240.0	246.0	233.0	239.5	246.0	230.0	242.0	247.0	234.0
232.0	234.0	221.0	232.0	234.0	225.0	232.5	234.0	222.0	230.0	233.0	222.0	233.0	234.0	227.0
65.0	66.0	61.0	65.0	66.0	60.0	65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	61.0
70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	69.0
79.0	79.0	78.0	79.0	79.0	76.0	79.0	79.0	76.0	79.0	80.0	77.0	79.0	79.0	76.0
66.0	67.0	64.0	66.0	67.0	64.0	66.0	67.0	64.0	67.0	67.0	64.0	66.0	67.0	63.0
65.0	66.0	64.0	65.0	66.0	64.0	65.0	66.0	63.0	65.0	66.0	64.0	66.0	66.0	64.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
34.0	34.0	34.0	34.5	35.0	34.0	35.0	35.0	34.0	34.0	35.0	34.0	34.0	35.0	34.0
34.0	34.0	32.0	34.0	34.0	30.0	34.0	34.0	32.0	34.0	34.0	30.0	34.0	34.0	32.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
17.0	18.0	16.0	17.0	18.0	17.0	17.0	18.0	15.0	17.0	18.0	15.0	17.0	18.0	16.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
194.0	195.0	192.0	194.0	195.0	193.0	194.0	195.0	192.0	194.5	195.0	193.0	194.0	195.0	192.0
168.0	169.0	166.0	167.0	169.0	166.0	168.0	169.0	166.0	168.0	169.0	164.0	168.0	169.0	163.0
183.0	183.0	179.0	183.0	183.0	181.0	183.0	183.0	181.0	182.5	183.0	182.0	183.0	184.0	180.0
184.0	184.0	182.0	184.0	184.0	179.0	184.0	184.0	182.0	184.0	185.0	180.0	184.0	184.0	180.0
184.0	186.0	180.0	184.5	188.0	182.0	185.0	186.0	184.0	184.0	186.0	183.0	184.0	184.0	184.0
73.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	70.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	68.0	65.0	67.0	68.0	66.0	67.0	67.0	67.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 14: V_2 with all discretization functions

V_3														
D_1			D_2			D_3			D_4			D_5		
med	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**
440.0	452.0	432.0	444.5	466.0	433.0	442.0	457.0	432.0	442.0	459.0	434.0	442.0	456.0	434.0
531.5	556.0	516.0	537.0	554.0	519.0	534.5	564.0	518.0	533.5	561.0	518.0	534.0	559.0	517.0
554.5	588.0	520.0	561.0	581.0	529.0	556.0	573.0	531.0	557.0	578.0	548.0	552.5	582.0	538.0
513.5	537.0	497.0	509.5	537.0	497.0	511.5	534.0	497.0	518.5	538.0	499.0	513.0	538.0	498.0
527.5	538.0	515.0	526.0	539.0	514.0	525.0	538.0	514.0	527.0	539.0	514.0	526.0	537.0	514.0
569.5	593.0	560.0	566.0	589.0	560.0	565.0	591.0	560.0	565.0	597.0	560.0	564.5	593.0	560.0
435.0	447.0	434.0	435.5	451.0	433.0	442.0	450.0	434.0	441.5	449.0	431.0	441.0	455.0	433.0
522.0	537.0	494.0	503.0	554.0	494.0	507.5	553.0	494.0	522.0	551.0	492.0	522.0	553.0	494.0
679.5	700.0	658.0	676.5	695.0	661.0	675.5	695.0	658.0	676.5	695.0	661.0	675.5	693.0	660.0
525.0	545.0	518.0	524.0	541.0	518.0	526.0	546.0	518.0	527.5	537.0	518.0	526.0	551.0	518.0
262.0	264.0	261.0	262.0	269.0	257.0	262.0	267.0	257.0	262.0	264.0	258.0	263.0	265.0	261.0
315.5	324.0	303.0	314.0	323.0	304.0	313.5	321.0	306.0	314.0	327.0	304.0	314.0	324.0	310.0
234.0	243.0	229.0	232.0	243.0	229.0	232.0	243.0	229.0	232.0	244.0	230.0	232.5	244.0	229.0
246.0	248.0	242.0	245.0	248.0	242.0	245.0	248.0	242.0	246.0	249.0	245.0	245.5	248.0	243.0
221.0	224.0	216.0	221.0	224.0	216.0	219.5	224.0	216.0	219.0	224.0	218.0	220.0	222.0	216.0
226.0	239.0	213.0	226.0	239.0	214.0	226.0	238.0	215.0	226.0	238.0	217.0	226.0	238.0	216.0
306.0	321.0	298.0	306.5	316.0	297.0	306.0	314.0	297.0	308.0	316.0	296.0	304.5	321.0	296.0
305.5	319.0	299.0	307.0	314.0	298.0	308.0	320.0	293.0	306.0	320.0	295.0	307.0	319.0	290.0
281.0	281.0	281.0	281.0	284.0	280.0	281.0	281.0	280.0	281.0	291.0	280.0	281.0	281.0	280.0
276.0	281.0	271.0	276.0	281.0	271.0	276.0	280.0	271.0	276.0	281.0	271.0	276.0	281.0	271.0
146.0	158.0	143.0	146.0	154.0	143.0	146.5	158.0	143.0	146.0	158.0	144.0	145.5	151.0	143.0
150.0	155.0	146.0	150.0	150.0	150.0	150.0	152.0	150.0	150.0	151.0	150.0	150.0	150.0	150.0
152.5	159.0	149.0	152.0	158.0	147.0	151.5	155.0	148.0	153.0	156.0	149.0	153.0	157.0	151.0
135.0	135.0	135.0	135.0	137.0	133.0	135.0	137.0	133.0	135.0	137.0	133.0	135.0	137.0	133.0
170.5	172.0	164.0	169.5	172.0	165.0	171.0	172.0	165.0	170.0	172.0	165.0	170.0	172.0	165.0
275.5	281.0	271.0	276.5	281.0	271.0	276.0	281.0	271.0	276.0	280.0	271.0	275.5	282.0	271.0
265.0	268.0	260.0	265.0	269.0	259.0	265.0	269.0	261.0	265.0	268.0	263.0	265.0	268.0	260.0
243.0	253.0	235.0	241.5	257.0	238.0	243.5	254.0	234.0	245.0	255.0	238.0	245.0	252.0	235.0
247.0	249.0	244.0	244.0	249.0	239.0	245.5	249.0	244.0	244.0	249.0	244.0	244.0	249.0	241.0
239.0	242.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.5	240.0	237.0
75.0	78.0	71.0	74.5	78.0	72.0	75.0	79.0	70.0	74.0	78.0	72.0	74.0	77.0	71.0
86.0	88.0	79.0	85.0	88.0	82.0	84.0	88.0	80.0	85.0	88.0	80.0	85.0	88.0	82.0
83.0	86.0	80.0	83.0	85.0	80.0	83.0	86.0	80.0	83.0	85.0	80.0	84.0	86.0	80.0
84.5	86.0	81.0	84.0	84.0	84.0	84.0	86.0	81.0	84.0	84.0	84.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	238.0	231.0	235.0	235.0	235.0	235.0	237.0	232.0	235.0	236.0	233.0	235.0	235.0	235.0
230.5	237.0	224.0	230.0	234.0	225.0	231.0	235.0	225.0	231.0	234.0	224.0	232.0	234.0	225.0
266.0	273.0	258.0	265.0	277.0	258.0	266.0	274.0	258.0	267.0	277.0	259.0	265.0	277.0	255.0
239.0	246.0	233.0	241.0	246.0	230.0	240.0	246.0	228.0	241.0	246.0	232.0	240.0	246.0	229.0
228.0	233.0	224.0	229.0	233.0	222.0	229.0	233.0	224.0	228.0	233.0	226.0	231.5	233.0	222.0
65.0	66.0	60.0	65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	64.0
70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	69.0	70.0	70.0	69.0
79.0	79.0	79.0	79.0	80.0	76.0	79.0	79.0	77.0	79.0	79.0	78.0	79.0	79.0	76.0
66.0	67.0	63.0	66.0	67.0	64.0	66.0	67.0	64.0	66.0	67.0	64.0	66.0	67.0	63.0
65.5	66.0	63.0	65.0	66.0	63.0	65.0	65.0	65.0	65.0	66.0	64.0	65.0	66.0	64.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
35.0	35.0	34.0	34.0	35.0	34.0	35.0	35.0	34.0	34.0	35.0	34.0	35.0	35.0	34.0
32.5	34.0	30.0	34.0	34.0	30.0	33.5	34.0	32.0	34.0	34.0	32.0	34.0	34.0	31.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
17.0	18.0	17.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	16.0	17.0	18.0	17.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
194.5	195.0	191.0	195.0	195.0	193.0	194.5	195.0	193.0	194.0	195.0	190.0	194.0	195.0	192.0
168.0	169.0	166.0	167.0	169.0	165.0	169.0	169.0	166.0	167.0	169.0	166.0	168.0	169.0	166.0
182.5	183.0	182.0	183.0	183.0	182.0	183.0	183.0	182.0	183.0	183.0	181.0	183.0	183.0	181.0
184.0	184.0	180.0	184.0	184.0	179.0	184.0	184.0	180.0	184.0	184.0	178.0	184.0	184.0	182.0
184.0	186.0	182.0	184.0	186.0	183.0	184.0	186.0	183.0	185.0	190.0	181.0	184.0	186.0	183.0
73.0	74.0	70.0	73.0	73.0	70.0	73.0	73.0	70.0	73.0	73.0	72.0	73.0	73.0	71.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	68.0	67.0	67.0	68.0	65.0	67.0	67.0	67.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 15: V_3 with all discretization functions

V4														
D1			**D2**			**D3**			**D4**			**D5**		
med	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**	**med**	**max**	**min**
442.0	456.0	434.0	457.0	468.0	440.0	440.0	461.0	432.0	444.0	456.0	434.0	441.0	465.0	432.0
527.5	555.0	515.0	541.0	567.0	515.0	539.0	559.0	517.0	533.0	559.0	515.0	532.5	555.0	515.0
554.5	580.0	535.0	557.0	574.0	531.0	554.5	573.0	528.0	554.0	582.0	538.0	551.5	569.0	531.0
511.0	538.0	497.0	513.0	541.0	498.0	511.5	542.0	497.0	507.0	535.0	497.0	515.5	541.0	497.0
526.5	537.0	514.0	527.0	539.0	517.0	528.0	539.0	516.0	526.0	538.0	516.0	526.0	539.0	514.0
566.0	589.0	560.0	569.0	589.0	560.0	567.0	597.0	560.0	564.5	585.0	560.0	566.0	587.0	560.0
440.5	450.0	434.0	441.5	450.0	434.0	440.5	451.0	434.0	438.5	449.0	434.0	441.5	458.0	434.0
522.0	553.0	494.0	522.0	553.0	503.0	522.0	553.0	503.0	522.0	543.0	494.0	503.0	537.0	494.0
676.0	708.0	661.0	676.0	689.0	665.0	674.0	693.0	660.0	675.0	688.0	662.0	675.0	692.0	660.0
524.0	542.0	517.0	527.5	543.0	518.0	527.5	542.0	517.0	526.5	543.0	518.0	527.0	543.0	518.0
263.0	272.0	261.0	262.0	264.0	258.0	262.0	268.0	259.0	262.0	267.0	257.0	262.0	268.0	257.0
314.0	323.0	303.0	313.0	325.0	306.0	314.0	323.0	309.0	312.5	324.0	309.0	313.5	324.0	304.0
232.0	243.0	229.0	232.5	244.0	229.0	233.0	242.0	229.0	231.5	242.0	229.0	233.0	244.0	229.0
245.0	248.0	242.0	245.5	248.0	245.0	245.0	248.0	242.0	245.5	248.0	245.0	245.5	248.0	242.0
220.5	224.0	218.0	221.0	224.0	216.0	221.0	224.0	216.0	219.5	224.0	216.0	219.5	224.0	218.0
226.0	239.0	217.0	226.0	239.0	217.0	225.5	238.0	213.0	226.0	238.0	213.0	225.5	237.0	213.0
304.0	319.0	296.0	302.5	317.0	294.0	303.0	317.0	299.0	306.5	317.0	295.0	307.0	322.0	298.0
307.0	323.0	288.0	310.0	315.0	295.0	306.5	319.0	299.0	305.0	321.0	294.0	306.5	319.0	296.0
281.0	289.0	280.0	281.0	281.0	280.0	281.0	281.0	280.0	281.0	287.0	280.0	281.0	281.0	280.0
276.0	281.0	271.0	276.0	283.0	271.0	275.0	281.0	271.0	276.0	281.0	271.0	276.0	281.0	269.0
146.0	158.0	143.0	149.0	155.0	143.0	148.0	154.0	143.0	145.5	158.0	143.0	146.0	158.0	143.0
150.0	153.0	150.0	150.0	151.0	150.0	150.0	150.0	150.0	150.0	153.0	150.0	150.0	150.0	150.0
152.0	155.0	149.0	151.5	155.0	148.0	152.0	158.0	149.0	153.0	155.0	149.0	153.0	156.0	148.0
135.0	137.0	133.0	135.0	138.0	133.0	135.0	137.0	133.0	135.0	137.0	133.0	135.0	138.0	133.0
170.0	173.0	165.0	170.0	172.0	164.0	168.5	171.0	164.0	171.0	172.0	165.0	170.0	172.0	165.0
276.0	282.0	271.0	275.0	280.0	271.0	274.5	283.0	269.0	274.0	283.0	271.0	275.0	281.0	271.0
265.0	268.0	262.0	265.0	269.0	262.0	265.0	269.0	261.0	265.0	270.0	262.0	265.5	269.0	261.0
244.5	257.0	236.0	246.0	256.0	235.0	246.0	254.0	235.0	244.0	251.0	236.0	245.5	251.0	235.0
244.0	249.0	241.0	244.0	251.0	244.0	246.0	250.0	241.0	244.0	249.0	237.0	245.0	249.0	244.0
239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.0	240.0	237.0	239.5	240.0	237.0
75.0	78.0	70.0	74.0	77.0	72.0	74.0	79.0	73.0	74.0	79.0	72.0	74.0	78.0	72.0
84.0	88.0	80.0	85.0	89.0	82.0	84.5	88.0	82.0	85.0	88.0	82.0	84.0	88.0	81.0
82.5	86.0	80.0	83.0	85.0	80.0	83.0	85.0	80.0	82.5	85.0	80.0	83.0	85.0	80.0
84.0	84.0	84.0	84.0	84.0	84.0	84.0	86.0	82.0	84.0	84.0	84.0	84.0	84.0	84.0
73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0	73.0
235.0	237.0	232.0	235.0	238.0	231.0	235.0	238.0	231.0	235.0	238.0	231.0	235.0	236.0	233.0
233.5	235.0	223.0	231.0	235.0	224.0	230.5	238.0	226.0	230.0	234.0	224.0	229.0	236.0	225.0
266.5	274.0	258.0	264.0	275.0	257.0	266.0	277.0	258.0	264.0	273.0	256.0	266.0	277.0	255.0
239.0	246.0	228.0	239.5	246.0	227.0	240.0	246.0	229.0	240.5	246.0	232.0	237.0	246.0	232.0
232.0	234.0	221.0	231.5	233.0	222.0	232.0	236.0	224.0	228.0	233.0	222.0	229.0	236.0	222.0
65.0	66.0	61.0	65.0	66.0	61.0	65.0	66.0	60.0	65.0	66.0	61.0	65.0	66.0	61.0
70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0	70.0
79.0	79.0	79.0	79.0	80.0	77.0	79.0	79.0	76.0	79.0	79.0	79.0	79.0	79.0	79.0
66.5	67.0	64.0	67.0	67.0	63.0	66.0	67.0	64.0	66.5	67.0	63.0	66.0	67.0	64.0
65.0	66.0	64.0	65.0	66.0	63.0	65.0	66.0	63.0	65.0	66.0	64.0	65.0	66.0	63.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
35.0	35.0	34.0	35.0	35.0	34.0	35.0	35.0	34.0	34.0	35.0	34.0	35.0	35.0	34.0
34.0	34.0	30.0	34.0	34.0	29.0	33.5	34.0	32.0	32.0	34.0	29.0	34.0	34.0	31.0
33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	33.0	32.0	33.0	33.0	33.0
30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	17.0	18.0	18.0	18.0
17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0	17.0
17.0	18.0	16.0	17.0	17.0	17.0	17.0	18.0	15.0	17.0	17.0	17.0	17.0	18.0	16.0
16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
195.0	195.0	193.0	194.0	195.0	192.0	194.5	195.0	192.0	194.0	195.0	193.0	194.0	195.0	192.0
167.0	169.0	165.0	167.0	169.0	166.0	168.0	169.0	166.0	168.0	169.0	166.0	168.0	169.0	166.0
183.0	183.0	181.0	182.0	184.0	182.0	183.0	183.0	182.0	183.0	183.0	182.0	183.0	183.0	182.0
183.5	184.0	182.0	184.0	184.0	181.0	184.0	184.0	182.0	184.0	184.0	181.0	184.0	186.0	182.0
184.5	187.0	183.0	186.0	187.0	183.0	185.0	186.0	183.0	185.5	186.0	183.0	184.5	186.0	183.0
73.0	73.0	70.0	73.0	73.0	70.0	72.0	73.0	70.0	73.0	73.0	71.0	73.0	73.0	71.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0	67.0
69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0	69.0
67.0	67.0	67.0	67.0	67.0	67.0	67.0	68.0	66.0	67.0	68.0	66.0	67.0	68.0	66.0
61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0	61.0

Table 16: V_4 with all discretization functions