On Heaviside Binary Logic: a Mathematical Characterization and its Applications

Giovanni Iacovelli , *Student Member, IEEE*, and Claudio Iacovelli , *Student Member, IEEE*

Abstract—Major advances in computer capabilities, together with a huge increase in the volume of data being produced every day, have made Artificial Intelligence an essential technology driving health and wellness growth. Main applications include speech recognition, recommendations systems, computer vision, and medical diagnosis from images. The theoretical concepts asserted by Alan Turing remain, in any case, at the basis of the computation and hence of machine intelligence. Turing Machine, the fundamental computational model, has been proven to be reducible to a logic circuit and, at the same time, portable into a computer program. In this work, the existence of a correspondence between fundamental programming language control structures and mathematical formulation is proven. The proposed interpretation is given through a well-defined logical circuit analytical expression. Relevant geometrical applications having wide implications in engineering branches are presented together with a new Penalty Method for constrained optimization problems handling.

Index Terms—Analytical models, Logic programming, Computational intelligence, Optimization, Mathematical programming

1 INTRODUCTION

The concept of algorithm [1] was first introduced when David Hilbert posed the *Entscheidungsproblem* in 1928. Still, it does not have a unique definition. Essentially, it can be regarded as a finite succession of elementary instructions which leads to a result in a finite amount of time. The first and most famous mathematical interpretation was proposed by Kurt Gödel in the early '30s who introduced the notion of recursive functions [2]. Algorithm first characterization took place in 1936 when Alan Turing created a theoretical model, called Touring Machine (TM) [3] in order to prove that Hilbert's problem was unsolvable.

Under specific conditions, any function computed by TM, deterministic or not, can be reduced to a circuit which is composed by a set of Logic Gates (LGs) [4]. Such digital devices implement a Boolean function that takes one or more binary inputs and provides a binary output. Further, patches of circuits can be synthetized into a complicated mathematical objects. For instance, this could either be the transfer function of the electronic circuit or an object that is associated to the truth table of it. Those objects can be inputs of other gates whose definition can be extended to accept and return not only numbers but also functions. The composition of this type of gates results in the description of the whole circuit. In mathematical terms, binary status can be expressed and LGs can be realized as results of specific composition of a certain number of Heaviside Step Functions (HSFs).

This idea can also be applied to programming languages, because they provide an high-level abstraction of the machine language that is mapped into logic circuits.

In fact, Böhm-Jacopini's Theorem [5] states that (i) every algorithm/function can be implemented through sequence, selection and iteration (ii) every TM, up to an extent, is equivalent to a program with only compositions and iterations. This means that it is possible to associate an algorithm to a specific problem, a TM to the algorithm, a logic circuit to the TM and a function to the logic circuit. The goal is to correlate an analytical function to the algorithm. HSF emulates the binary selection operation by design. Moreover, iterations can be executed based on essence of operations involved, while sequence is inherently guaranteed.

This concept is particularly suitable in case of analytical/geometrical problems especially in computer graphic, computer vision and geoscience applications. In these fields, one of the fundamental topics is polygons processing which is examined in several studies [6] [7] [8] [9] in literature. Moreover, in [10] two algorithms are designed to discover intersection points between simple polygons. The first is a set-based intersection algorithm while the second is binary-search-tree-based algorithm which allows the authors to briefly describe a set of possible merging and Boolean operations among polygons. These are also the focus of [11] where the proposed algorithm is also capable to deal with polygons with holes. [12] proposes a GPU-based rasterized polygon intersection area algorithm able to work with complex and large-scale ones. Further, in [13] an algorithm capable of performing clipping, Boolean union and difference between input polygons is presented. Based on entry/exit intersection point property, it uses an efficient data structure and can easily also be adapted to Boolean operations between regions composed by input sets. These kind of geometrical problems have also implications in medical imaging. For instance, in [14] a method is developed for the analytical calculation of the 3D shape and volume of volumes' intersection in order to reconstruct 3D Positron Emission Tomography data. In Robotics geometrical information about configuration [15], placement, collision [16] or

- G. Iacovelli is with the Department of Electrical and Information Engineering, Politecnico di Bari/CNIT, Bari, Italy.
E-mail: giovanni.iacovelli@poliba.it
- C. Iacovelli is with the Department of Physics, University of Trento, Trento, Italy.
E-mail: claudio.iacovelli@studenti.unitn.it

proximity [17] among objects are major concerns. In [18], an algorithm is proposed in order to avoid collisions between a manipulator and an obstacle. These are represented by cylinders and convex plane polygons respectively, while the intersection among these describes the physical contact. In [19], an algorithm is designed to address another challenging aspect: motion planning. This work takes into account planar polygonal rigid robots that have to move in a polyhedral environment and avoid collisions thanks to decomposability property. The process of determining the robot's position and the obstacles places involves Point-in-Polygon Problem. In [20], instead, the authors propose an algorithm capable of solving this issue and deal with concave and convex polygons of different shapes.

This work tackles these geometrical problems through a set of mathematical solutions. Some of them belong to a cost function which requires to be probed. Optimization techniques [21] and Artificial Intelligence Search Algorithms (SAs) [22], such as Genetic Algorithms (GAs) [23], are able to probe and pick the optimum or quasi-optimum solution in the output space. Therefore, to handle such non-convex and even constrained optimization problems, a new penalty function method, in combination with GAs [24], is studied.

The present work is organized as follows: Section 2 introduces mathematical definitions and properties of LGs realized through HSF. Section 3 discusses continuous representation of HSF adopted. Section 4 establish a relation between algorithms and proposed gates. With Section 5, formulation of the solutions to a set of envisioned problems is given. Section 6, instead, presents a new Penalty Method for constrained optimization problems based on examined notions. Section 7, analyses numerical results derived from theoretical concepts discussed. Finally, Section 8 concludes the work and draws future works and research perspectives.

2 HSF BOOLEAN LOGIC

In this Section, a mathematical characterization is addressed to achieve a proper definition of LGs functions realized by $\Theta(x)$, i.e. HSF. Information is carried by x , and its binary description is achieved through a discrimination with respect to a threshold, i.e. zero value. In the most complicated case $\Theta(x)$ could not be defined for $x = 0$. On the contrary, the upcoming concepts will results in a simplification.

2.1 HSF Circuit definition

Without loss of generality, let the possible inputs be described with a vector space over a real scalar field, i.e. $\mathbb{R}^D$, being $O = \{x \in \mathbb{R}^D \mid \exists i : 1 \dots D \ni x^i = 0\}$. Therefore, the definition of the function that encodes the real inputs in binary information is:

$$\begin{aligned} \vec{\Theta}_0 : \mathbb{R}^D \setminus O &\rightarrow \mathcal{B}^D \\ x \rightarrow \vec{\Theta}_0(x) &:= (\Theta(x_1), \dots, \Theta(x_D))^T, \end{aligned} \quad (1)$$

where $\mathcal{B} = \{0, 1\}$. Let $R \geq 1 \in \mathbb{N}$, $N_1 = D$ and $N_{R+1} = Q$. Then, it is necessary to define the following sets as:

$$\begin{aligned} \{N_r \mid r : 1 \dots R+1, N_r \in \mathbb{N}\} \\ \{\vec{g}_r : \mathcal{B}^{N_r} \rightarrow \mathbb{R}^{N_{r+1}}\}_{r:1 \dots R} \\ \{\vec{\Theta}_r : \mathbb{R}^{N_{r+1}} \setminus O_{r+1} \rightarrow \mathcal{B}^{N_{r+1}}\}_{r:1 \dots R}, \end{aligned}$$

where O_{r+1} is the analogous of O for any arbitrary input dimensionality $x \in \mathbb{R}^{N_{r+1}}$, $\vec{\Theta}_r$ acts in the same manner as $\vec{\Theta}_0$, and R is the rank, i.e. the number of compositions the HSF Circuit (HSFC) will be made of. Moreover, $\vec{g}_r, r : 1 \dots R$ must be a uniform continuous function. In presence of a chain of compositions, the inner argument of $\vec{\Theta}_{r+1}$ together with its kernel is:

$$\begin{aligned} \vec{h}_{r+1} &= \vec{g}_{r+1} \circ \vec{\Theta}_r \circ \vec{h}_r, \quad r : 1 \dots R-1, \quad \vec{h}_1 = \vec{g}_1, \\ K(\vec{h}_r) &= \{\vec{\Theta}_0(x) \in \mathcal{B}^D \mid \exists i : 1 \dots N_{r+1} \ni h_r^i(\vec{\Theta}_0(x)) = 0\}. \end{aligned} \quad (2)$$

These kernels have to be excluded, because $\vec{\Theta}_r(\vec{h}_r(\vec{\Theta}_0(x) \in \text{Ker}(h_r)))$ would not be defined. Obviously, the only argument to be checked is $\vec{h}_1$ if the rank is $R = 1$. Therefore, if $\mathcal{K}$ is the union of all kernels, the proposed definition of a HSFC $\mathcal{H}$ of rank R is:

$$\mathcal{H}^R : \mathcal{B}^D \setminus \mathcal{K} \rightarrow \mathcal{B}^Q \quad (3)$$

$$y \rightarrow \mathcal{H}^R(y) = \left(\bigcirc_{r=1}^R \vec{\Theta}_r \circ \vec{g}_r \right)(y), \quad (4)$$

where $\bigcirc_{r=1}^R (\cdot) = \vec{\Theta}_R \circ \vec{g}_R \circ \dots \circ \vec{\Theta}_1 \circ \vec{g}_1$, and $y = \vec{\Theta}_0(x)$ is the input string. If $\Theta(0)$ is defined, then the previous definition also holds in $\mathcal{K}$. Indeed, this is a Boolean function with input $\vec{\Theta}_0(x)$. The set of these functions does not have a group structure. However, the set is closed with respect to the composition operation. In the spirit of this notation, the most trivial circuit is the identity which is obtained by setting $R = 1$ and $D = Q$:

$$\begin{aligned} \mathcal{I} : \mathcal{B}^D &\rightarrow \mathcal{B}^D \\ y \rightarrow \vec{\Theta}_0(y - \xi) &= y, \end{aligned} \quad (5)$$

derived from Eq.(4) with:

$$\begin{aligned} \vec{g}_1 : \mathcal{B}^D &\rightarrow \mathbb{R}^D, \quad y \rightarrow y - \xi \\ \vec{\Theta}_1 : \mathbb{R}^D \setminus \mathcal{K} &\rightarrow \mathcal{B}^D, \quad z \rightarrow \vec{\Theta}_0(z), \end{aligned} \quad (6)$$

where $\xi \in]0, 1[^D$ being an arbitrary parameter which takes into account the fact that $\Theta(0)$ may not be defined or be different from zero. This translation allows to have $\mathcal{K} = K(\vec{h}_1) = K(y - \xi) = \emptyset$.

Remark 1. Similar considerations can be done for any set of functions $\vec{f}_i : \mathbb{R}^{N_{i+1}} \rightarrow \mathcal{B}^{N_{i+1}}$ $i : 0 \dots R$, that encode in binary the information, by substituting $\vec{\Theta}_i(\cdot)$ in the chain of composition. Indeed, $\vec{\Theta}_i(\cdot)$ can be another HSFC.

2.2 Fundamental HSF Gates

It is worth defining an equivalent LG standard basis $\Psi_L = \{NOT, AND, OR\}$ with the use of Eq.(4). The HSF representation of Ψ_L is defined as the unitary rank ($R = 1$) basis $\Psi_H = \{N, \Lambda, \Omega\}$, constituted by the fundamental HSF Gates (HSFGs). These particular circuits are:

- NOT:

$$\begin{aligned} \vec{g}_1 : \mathcal{B}^D &\rightarrow \mathbb{R}^D, \quad y \rightarrow \vec{g}_1(y) = (1 - y) - \xi \\ \vec{\Theta}_1 : \mathbb{R}^D &\rightarrow \mathcal{B}^D, \quad z \rightarrow \vec{\Theta}_0(z) \\ \vec{\Theta}_1 \circ \vec{g}_1 : \mathcal{B}^D &\rightarrow \mathcal{B}^D, \quad y \rightarrow \vec{\Theta}_0((1 - y) - \xi) = \mathcal{I}(1 - y) \\ \vec{N}(x) &:= 1 - \vec{\Theta}(x) = \vec{\Theta}(-x), \end{aligned} \quad (7)$$

with $\xi \in]0, 1[^D$.

- AND:

$$\begin{aligned} g_1 : \mathcal{B}^D &\rightarrow \mathbb{R}, & \mathbf{y} &\rightarrow g_1(\mathbf{y}) = \prod_{i=1}^D y_i - \xi \\ \Theta_1 : \mathbb{R} &\rightarrow \mathcal{B}, & z &\rightarrow \Theta(z) \\ \Theta_1 \circ g_1 : \mathcal{B}^D &\rightarrow \mathcal{B}, & \mathbf{y} &\rightarrow \Theta\left(\prod_{i=1}^D y_i - \xi\right) = \mathcal{I}\left(\prod_{i=1}^D y_i\right) \\ \Lambda(\mathbf{x}) &:= \prod_{i=1}^D \Theta(x_i). \end{aligned} \quad (8)$$

- OR:

$$\begin{aligned} g_1 : \mathcal{B}^D &\rightarrow \mathbb{R}, & \mathbf{y} &\rightarrow g_1(\mathbf{y}) = \sum_{i=1}^D y_i - \xi \\ \Theta_1 : \mathbb{R} &\rightarrow \mathcal{B}, & z &\rightarrow \Theta(z) \\ \Theta_1 \circ g_1 : \mathcal{B}^D &\rightarrow \mathcal{B}, & \mathbf{y} &\rightarrow \Theta\left(\sum_{i=1}^D y_i - \xi\right) \\ \Omega(\mathbf{x}) &:= \Theta\left(\sum_{i=1}^D \Theta(x_i) - \xi\right), \end{aligned} \quad (9)$$

with $\xi \in]0, 1[$.

As a matter of fact, for a given binary input $\mathbf{y}$, the output of any HSFG of Ψ_H is the same of its correspondent LG in Ψ_L . Again, the absence of a definition of $\Theta(0)$ is taken into account thanks to ξ .

Theorem 1. Let $\mathbf{y} \in \mathcal{B}^D \setminus \mathcal{K}$ be the input of a HSFG $\mathcal{H}^R$. Then, there must exist a logic circuit C such that $C(\mathbf{y}) = \mathcal{H}^R(\mathbf{y})$ where $C(\mathbf{y})$ is the output of C receiving $\mathbf{y}$.

Proof. According to the definition of Ψ_H , each function of this set has an I/O relation which has an exact correspondence with the truth table of its analogous in Ψ_L . Since Ψ_L can reproduce any truth table of any logic circuit C via composition of its LGs, then the same must hold for Ψ_H . This means that any HSFG $\mathcal{H}^R$ determines an I/O relation which must be reproducible through a composition of functions in Ψ_L . ■

Corollary 1. NAND and NOR alone are capable of constructing any HSFG via its composition, because it must be as fundamental as Ψ_H . Recalling Remark 1, these can be obtained by composing elements from the Ψ_H basis, in the following manner:

- NAND $\bar{\Lambda} := N \circ \Lambda$

$$\bar{\Lambda}(\mathbf{x}) := 1 - \prod_{i=1}^D \Theta(x_i). \quad (10)$$

- NOR $\bar{\Omega} := N \circ \Omega$

$$\bar{\Omega}(\mathbf{x}) := \Theta\left(\xi - \sum_{i=1}^D \Theta(x_i)\right) \quad (11)$$

with $\xi \in]0, 1[$.

Corollary 2. DeMorgan's Theorems are applicable to HSFGs which compose the basis Ψ_H .

$$\Lambda \circ \bar{N} = N \circ \Omega \quad \Omega \circ \bar{N} = N \circ \Lambda. \quad (12)$$

3 CONTINUOUS HSF GATES DEFINITION

The HSF is not suited for contexts and applications in which the differentiability of the functions is required. Several continuous approximations are available but, in this paper, the proposed substitution is:

$$\Theta(x) \rightarrow \sigma(\alpha x) \equiv \sigma_\alpha(x) := \frac{1}{1 + e^{-\alpha x}}, \quad (13)$$

where σ is the Sigmoid Function (SF) that has been widely used in engineering and physics application [25] [26]. Furthermore, α is a coefficient that should be set sufficiently high: hereafter it is shown that the difference between the two functions depends on it. Such difference is defined as:

$$\delta_\alpha(x) := \Theta(x) - \sigma(\alpha x) = \begin{cases} \frac{e^{-\alpha x}}{1 + e^{-\alpha x}} & \text{if } x > 0 \\ -\frac{1}{1 + e^{-\alpha x}} & \text{if } x < 0 \\ \vartheta_0 - \frac{1}{2} & \text{if } x = 0 \end{cases}, \quad (14)$$

where $\vartheta_0 = \Theta(0)$. It can be compactly reformulated as

$$\delta_\alpha(x) = \frac{e^{-\alpha x}}{1 + e^{-\alpha x}} \Theta(x) - \frac{1}{1 + e^{-\alpha x}} \Theta(-x), \quad (15)$$

provided $\delta_0 = \delta_\alpha(0)$ is not taken into account, since in

Fig. 1: $\delta_\alpha(x)$ for different values of α .

this formulation $\delta_0 = 0$ independently of ϑ_0 . δ_α is an odd function in $\mathbb{R}^*$ and it is odd in $\mathbb{R}$ only if $\delta_0 = 0$: this means that (15) is odd in $\mathbb{R}$. Moreover, δ_α goes exponentially to zero $\forall x \neq 0$ if $\alpha \rightarrow \infty$. At the same time, it goes exponentially to zero $\forall \alpha > 0$ when $x \rightarrow \pm\infty$ and its 1-norm is $\frac{2 \ln(2)}{\alpha}$. Therefore, the difference between $\Theta(\cdot)$ and $\sigma(\alpha \cdot)$ is small if α is large.

The continuous version of a circuit is now seen as a function of a real input:

$$\begin{aligned} \mathcal{H}_\alpha^R : \bar{\mathcal{B}}^D &\rightarrow \bar{\mathcal{B}}^Q \\ \mathbf{y} &\rightarrow \mathcal{H}_\alpha^R(\mathbf{y}) = \left(\bigcirc_{r=1}^R \vec{\sigma}_{r,\alpha} \circ \vec{g}_r \right)(\mathbf{y}), \end{aligned} \quad (16)$$

where $\bar{\mathcal{B}} =]0, 1[$, $\mathbf{y} = (\vec{\sigma}_{0,\alpha}(\mathbf{x}))$,

$$\begin{aligned} \vec{\sigma}_{0,\alpha}(\mathbf{x}) : \mathbb{R}^D &\rightarrow \bar{\mathcal{B}}^D \\ \mathbf{x} &\rightarrow \vec{\sigma}_{0,\alpha}(\mathbf{x}) = (\sigma_\alpha(x_1), \dots, \sigma_\alpha(x_D))^T, \end{aligned} \quad (17)$$

and $\{\vec{g}_r : \bar{\mathcal{B}}^{N_r} \rightarrow \mathbb{R}^{N_{r+1}}\}_r$ is a set of uniformly continuous functions. Since the logistic function is defined $\forall x \in \mathbb{R}$, the circuit can clearly accept any $\mathbf{x} \in \mathbb{R}^D$. Nonetheless, it is

convenient to recast kernels in the input space similar to those defined in Eq.(2):

$$\begin{aligned} \vec{h}_{r+1} &= \vec{g}_{r+1} \circ \vec{\Theta}_r \circ \vec{h}_r, \quad r : 0..R-1, \quad \vec{h}_0 = \mathbf{x}, \\ K(\vec{h}_r) &= \{\mathbf{x} \in \mathbb{R}^D \mid \exists i : 1..N_{r+1} \ni h_r^i(\mathbf{x}) = 0\}, \end{aligned} \quad (18)$$

as well their union $\mathcal{K}$. Until the end of this Section, the circuits will be now considered as functions of $\mathbf{x}$, i.e. $\mathcal{H}^R(\mathbf{x}) \equiv (\mathcal{H}^R \circ \vec{\Theta}_0)(\mathbf{x})$ and $\mathcal{H}_\alpha^R \equiv (\mathcal{H}_\alpha^R \circ \vec{\sigma}_{0,\alpha})(\mathbf{x})$.

Theorem 2. *Let $\{\vec{g}_r\}_r$ be a set of uniformly continuous functions, $\mathcal{H}^R$ a HSFC and $\mathcal{H}_\alpha^R$ a SF Circuit (SFC). Let X be any subset of $\mathbb{R}^D$ whose closure $\bar{X} \subset \mathbb{R}^D \setminus \mathcal{K}$. Then, $\mathcal{H}_\alpha^R$ converges in X uniformly to $\mathcal{H}^R$ for $\alpha \rightarrow \infty$.*

Proof. See **Appendix A**. ■

To get an estimate of the error committed in the evaluation of $\mathcal{H}_\alpha^R(\mathbf{x})$ in place of a given $\mathcal{H}^R(\mathbf{x})$, the circuits may be regarded as functionals with respect to either σ_α or Θ . As a consequence, the notation $\mathcal{H}[\Theta](\mathbf{x})$ describes Eq.(4) while $\mathcal{H}[\sigma_\alpha](\mathbf{x})$ describes Eq.(16). Now, the equality $\mathcal{H}^R[\Theta] = \mathcal{H}^R[\sigma_\alpha + \delta_\alpha](\mathbf{x})$ is trivial, and the variational Taylor series reads:

$$\begin{aligned} \mathcal{H}^R[\Theta] &= \mathcal{H}^R[\sigma_\alpha + \epsilon \delta_\alpha] \Big|_{\epsilon=1} = \\ &= \mathcal{H}^R[\sigma_\alpha] + \sum_{k=1}^{\infty} \frac{d^k}{d\epsilon^k} \mathcal{H}^R[\sigma_\alpha + \epsilon \delta_\alpha] \Big|_{\epsilon=0} \frac{\epsilon^k}{k!} \quad @\epsilon = 1. \end{aligned} \quad (19)$$

This series is convergent, and it collects both the error and the discontinuity in the functional derivatives. In the case of interest, the parameter ϵ is set to one for the equality to hold, but the variation δ_α has been proved to be small, so the expectation is that the majority of the discrepancy has to be caused by the first orders. Finally, thanks to this analysis, it is possible to define Ψ_σ basis which is equivalent to Ψ_H but exploiting the logistic function. The fundamental SF Gates (SFGs) that constitutes this new basis are $N_\alpha, \Lambda_\alpha, \Omega_\alpha$ for which Corollary 2 still holds.

4 HSP CONTROL STRUCTURES

Once defined the analytical structure of a general SFG, a correlation can be established between them and algorithms.

John E. Savage proved the following, which appears as Theorem 3.8.1 of [4]

Theorem 3. *Any computation performed by a one-tape Turing machine M , deterministic or nondeterministic, on an input string w in T steps using m b -bit memory cells can be simulated by a circuit $\mathcal{C}_{M,T}$ over the standard complete basis Ω of size and depth $O(ST)$ and $O(T \log S)$, respectively, where $S = mb$ is the storage capacity in bits of M 's tape. For the deterministic TM the inputs to this circuit consist of the values of w . For the nondeterministic TM the inputs consist of w and the Boolean choice input variables whose values are not set in advance.*

Based on the above Theorem, the following is proved.

Theorem 4. *Let $\mathcal{P}$ be a program or an algorithm which computes a function in a finite time. Then, there must exist an equivalent HSFG $\mathcal{H}$.*

Proof. Recalling Böhm-Jacopini's Theorem, $\mathcal{P}$ is equivalent to TM M and can be expressed through a combination of the three fundamental control structures: sequence, selection and iteration. Therefore, reminding Theorem 3 it is possible to state that there exists a logic circuit C equivalent to M . Finally, considering Theorem 1 can be claimed that there must exist a HSFC $\mathcal{H}$ which is equivalent to C . Hence, control structures must also have a correspondent expression in these terms. ■

A slight interpretation is presented hereby. However, it does not pretend to be mathematically rigorous as it will be considered for future investigations. The proposed characterization is proved to be sufficient for the description of control structures which compose a finite-time algorithm. For this reason the state of all variables and operations can be expressed through Boolean strings and HSFCs, respectively. This means that a set of commands is translated into a set of HSFCs $\{\mathcal{H}_r^{R_r}\}_{r:1..S}$, and to execute this set in sequence means:

$$\mathbf{x}_{OUT} = (\bigcirc_{r=1}^S \mathcal{H}_r^{R_r})(\mathbf{x}_{IN}), \quad (20)$$

where $\mathbf{x}_{OUT}$ and $\mathbf{x}_{IN}$ are the state variable before and after the sequence.

The selection, instead, is inherently addressed by HSF. In fact, given logical conditions which has to be verified, this control structure can be described through a HSFC $\mathcal{H}^R$. This returns 1 if the conditions are not satisfied and 0 otherwise. The mathematical expression is:

$$\mathbf{x}_{OUT} = (f_1 \mathcal{H}^R + f_0 (1 - \mathcal{H}^R))(\mathbf{x}_{IN}), \quad (21)$$

where f_1 and f_0 are the statements executed in false and true cases, respectively.

Finally, for what concern finite iteration, the definition of a recursive function turns to be necessary:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{x}_{IN} \mathcal{H}^R(\mathbf{x}_{IN}) + f(\mathbf{x}_{IN})(1 - \mathcal{H}^R(\mathbf{x}_{IN})) \\ \mathbf{x}_2 &= \mathbf{x}_1 \mathcal{H}^R(\mathbf{x}_1) + f(\mathbf{x}_1)(1 - \mathcal{H}^R(\mathbf{x}_1)) \\ &\vdots \\ \mathbf{x}_n &= \mathbf{x}_{n-1} \mathcal{H}^R(\mathbf{x}_{n-1}) + f(\mathbf{x}_{n-1})(1 - \mathcal{H}^R(\mathbf{x}_{n-1})). \end{aligned}$$

This synthetizes into

$$\begin{aligned} \mathbf{x}_n &= \bigcirc_{i=1}^{n-1} f(\mathbf{x}_{IN}) \mathcal{H}^R(\bigcirc_{i=1}^{n-1} f(\mathbf{x}_{IN})) + \\ &+ \left(1 - \mathcal{H}^R(\bigcirc_{i=1}^{n-1} f(\mathbf{x}_{IN}))\right) \bigcirc_{i=1}^n f(\mathbf{x}_{IN}). \end{aligned} \quad (22)$$

In fact, once the n -th step is reached, all of the others have to be

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) = \bigcirc_{i=1}^{k-1} f(\mathbf{x}_{IN}), \quad \forall k : 1..n-1. \quad (23)$$

If the iteration ends at the $n+1$ -th step, then the output is:

$$\mathbf{x}_{n+1} = \mathbf{x}_n = \bigcirc_{i=1}^n f(\mathbf{x}_{IN}), \quad (24)$$

and this happens if the following condition is satisfied:

$$\forall \mathbf{x}_{IN} \in \Omega \exists n \in \mathbb{N} \ni \mathcal{H}^R(\bigcirc_{i=1}^{n-1} f(\mathbf{x}_{IN})) = 1, \quad (25)$$

with $\Omega \subseteq \mathcal{B}^D$ being the domain of the inputs.

5 COMPUTATION OF VOLUME MEASURE COVERED BY INTERSECTING GEOMETRICAL LOCI

In this Section, an interesting geometrical application is studied. The whole mathematical process is deeply described hereby and it is concluded with the equivalent interpretation in terms of HSFGs.

Consider a finite set of geometrical loci, described with

$$\{f_n(\mathbf{x}, \mathbf{q}_n) = 0 | n : 1..N\}, \quad (26)$$

where $\mathbf{x}$ is the generic coordinate in the space $\mathbb{R}^D$ in which the loci are embedded and $\mathbf{q}_n$ is a point in the the vector space of parameters which specify the locus. Addressing the point $\mathbf{q}_n$ with index n is equal to consider such a point and its dimensionality to be independent from all the others. It is also assumed that f_n identifies a closed and compact volume

$$V_n = \{\mathbf{x} | f_n(\mathbf{x}, \mathbf{q}_n) \leq 0\}. \quad (27)$$

The aim is to find an expression which describes the volume that is effectively encompassed by the set of loci, namely

$$V = \bigcup_{n=1}^N V_n = \left\{ \mathbf{x} \left| \bigvee_{n=1}^N f_n(\mathbf{x}, \mathbf{q}_n) \leq 0 \right. \right\}. \quad (28)$$

Clearly, the major difficulty stands in the fact that the sum of the measure of single volumes

$$\mu_S = \sum_n \mu(V_n) = \sum_n \int_{V_n} d\mathbf{x}, \quad (29)$$

counts the intersections among different V_n a certain number of times which will be later studied. Obviously, in the trivial case in which no intersections are present

$$\mu_S = \mu(V) \iff V_i \cap V_j = \emptyset \forall i, j : 1..N, i \neq j. \quad (30)$$

Otherwise, in order to obtain $\mu(V)$ from μ_S , each intersecting region has to be subtracted in an appropriate way. In other words, intersections have to be counted just once. For instance, consider a $D = 2$ scenario in which three squares $S_i, i : 1..3$ intersect in region $\mathcal{R}_i, i : 1..3$ as shown in Fig.2. Assuming v as the measure of the unit square, μ_S is equal to

Fig. 2: Example scenario of intersecting squares.

48v while $\mu(V) = 30v$. Defining $\mathcal{R}_1 = S_1 \cap S_2$, $\mathcal{R}_2 = S_2 \cap S_3$ and $\mathcal{R}_3 = S_1 \cap S_2 \cap S_3$, it results (in v units) that

$$\begin{aligned} \mu(V) &= \mu_S - \mu(\mathcal{R}_1) - \mu(\mathcal{R}_2) - 2\mu(\mathcal{R}_3) \\ &= 48 - 5 - 5 - 2 \times 4 = 30. \end{aligned} \quad (31)$$

As previously stated, the measure of each intersection region has to be subtracted a certain number of times that is equal to their multiplicity κ . But first, the characteristic function of the n -th volume has to be introduced:

$$\mathcal{X}[f_n](\mathbf{x}) := \Theta(-f_n(\mathbf{x}, \mathbf{q}_n)) \begin{cases} 1 & \text{if } f_n(\mathbf{x}, \mathbf{q}_n) < 0 \\ 0 & \text{if } f_n(\mathbf{x}, \mathbf{q}_n) > 0 \\ \vartheta_0 & \text{if } f_n(\mathbf{x}, \mathbf{q}_n) = 0 \end{cases}. \quad (32)$$

According to the definition of the Heaviside used in this formula, the points verifying $f_n = 0$ count as ϑ_0 that is usually equal to 1 or 1/2, but they also may be not counted at all if ϑ_0 is 0 or NaN. Then, the multiplicity of a point $\mathbf{x}$ is defined as follows:

$$\kappa(\mathbf{x}) := \sum_n \mathcal{X}[f_n](\mathbf{x}) - 1. \quad (33)$$

In particular, the multiplicity of a point $\mathbf{x}^*$ which belongs to one volume only is $\kappa(\mathbf{x}^*) = 0$.

Consider a subspace $\mathcal{M} \subseteq \mathbb{R}^D \ni V_n \subseteq \mathcal{M} \forall n$. Analytically, the contribution that has to be subtracted from (29) is

$$\int_{\mathcal{M}} \kappa(\mathbf{x}) d\mathbf{x}. \quad (34)$$

However, an issue emerges in this formulation since

$$\forall \mathbf{x} \in \mathcal{M} \ni \mathbf{x} \notin V : \kappa(\mathbf{x}) = -1.$$

That is, each point of $\mathcal{M}$ that does not belong to any V_n has -1 multiplicity, thus incorrectly increasing μ_S . A third addendum which rectifies this behaviour is needed:

$$\int_{\mathcal{M}} \Theta\left(\xi - \sum_n \mathcal{X}[f_n](\mathbf{x})\right) d\mathbf{x}. \quad (35)$$

As a matter of fact, the integrand is equal to 1 $\forall \mathbf{x} \ni f_n > 0 \forall n$. The parameter ξ is a shift that imposes the aforementioned behaviour. Indeed, points which do not belong to any volume return a value which is different from 0 according to the adopted definition of Heaviside:

$$\xi \in \begin{cases}]0, 1[& \text{if } \vartheta_0 = 1 \\]0, 1/2[& \text{if } \vartheta_0 = 1/2 \\ [0, 1[& \text{if } \vartheta_0 = 0 \end{cases} \quad (36)$$

Basically, as the results in (29)(34)(35) are brought together:

$$\mu(V) = \mu_S - \int_{\mathcal{M}} \kappa(\mathbf{x}) d\mathbf{x} - \int_{\mathcal{M}} \Theta\left(\xi - \sum_n \mathcal{X}[f_n](\mathbf{x})\right) d\mathbf{x}. \quad (37)$$

The expression above can be rewritten in a more compact manner. It is worth noting that (34) is equivalent to

$$\sum_n \int_{\mathcal{M}} \mathcal{X}[f_n](\mathbf{x}) d\mathbf{x} - \int_{\mathcal{M}} d\mathbf{x} = \sum_n \int_{V_n} d\mathbf{x} - \int_{\mathcal{M}} d\mathbf{x}. \quad (38)$$

The exchange between the sum and integral signs can be always done if N is finite, and also for $N \rightarrow \infty$ under the hypothesis of Fubini's Theorem. Moreover, the first

integrand is equal to 1 for every point belonging to V_n at fixed n and it is 0 elsewhere. Therefore, the whole first addendum is equivalent to (29). Hence (37) becomes:

$$\int_{\mathcal{M}} \left(1 - \Theta \left(\xi - \sum_n \mathcal{X}[f_n](\mathbf{x}) \right) \right) d\mathbf{x}. \quad (39)$$

Recalling results achieved in Section 2 it is clear that (39) is the following HSGs composition:

$$(N \circ \bar{\Omega} \circ \vec{N})(f_n(\mathbf{q}_n))_n(\mathbf{x}) = (\Omega \circ \vec{N})(f_n(\mathbf{q}_n))_n(\mathbf{x}), \quad (40)$$

which does not correspond to any element of the Ψ_H basis. However, it can be described by Algorithm 1 thanks to Theorem 4. Finally, all this leads to:

$$\mu(V) := \mathcal{V} = \int_{\mathcal{M}} \Omega[\{N(f_n)\}](\mathbf{x}) d\mathbf{x} \quad (41)$$

$$= \int_{\mathcal{M}} \Theta \left(\sum_n \mathcal{X}[f_n](\mathbf{x}) - \xi \right) d\mathbf{x}, \quad (42)$$

which perfectly describe (28). This expression, as stated in Corollary 2, is also equivalent to

$$\mathcal{V} = \int_{\mathcal{M}} \bar{\Lambda}[\{f_n\}](\mathbf{x}) d\mathbf{x}. \quad (43)$$

Remark 2. The characteristic function may not be defined on the locus $f_n = 0$ due to the definition of ϑ_0 . Nonetheless, the measure of a volume does not depend on the value assumed by ϑ_0 because such value concerns the border which is a zero measure set.

Remark 3. It is worth noting that is sufficient to change the sign of the argument of all HSFs in Eq.(42) to characterize the dual formula for the computation of the intersection among all volumes, that is:

$$V^* = \bigcap_{n=1}^N V_n = \left\{ \mathbf{x} \mid \bigwedge_{n=1}^N f_n(\mathbf{x}, \mathbf{q}_n) \leq 0 \right\}. \quad (44)$$

$$\begin{aligned} \mu(V^*) &:= \mathcal{V}^* = \int_{\mathcal{M}} \Theta \left(\xi - \sum_n \Theta(f_n(\mathbf{x}, \mathbf{q}_n)) \right) d\mathbf{x} \\ &= \int_{\mathcal{M}} \bar{\Omega}[\{f_n\}](\mathbf{x}) d\mathbf{x}. \end{aligned} \quad (45)$$

Again, recalling Corollary 2, an equivalent expression is

$$\mathcal{V}^* = \int_{\mathcal{M}} \Lambda[\{N(f_n)\}](\mathbf{x}) d\mathbf{x}, \quad (46)$$

that well describes (44).

Algorithm 1 Measure of the Volume

Input: $\{f_n\}$

Output: $\mathcal{V}$

- 1: **for** $i = 1 \dots \mathcal{M}$ **do**
 - 2: **if** $f_1(x_i) < 0$ or $f_2(x_i) < 0 \dots$ or $f_n(x_i) < 0$ **then**
 - 3: $\mathcal{V} \leftarrow \mathcal{V} + \rho^2$ $\triangleright \rho$ stands for mesh precision
 - 4: **end if**
 - 5: **end for**
-

5.1 Error Analysis

The error which is introduced by the use of the logistic function in place of Heaviside is addressed hereby. First, it is necessary to introduce the expression of the error:

$$\begin{aligned} \delta\mathcal{V} &:= \int_{\mathcal{M}} \left[\Theta \left(\sum_n \Theta(-f_n(\mathbf{x}, \mathbf{q}_n)) - \xi \right) + \right. \\ &\quad \left. - \sigma \left(\alpha \left(\sum_n \sigma(-\alpha f_n(\mathbf{x}, \mathbf{q}_n)) - \xi \right) \right) \right] d\mathbf{x}. \end{aligned} \quad (47)$$

Considering (19), introduce the variable $z = \alpha \sum_n (\sigma_\alpha + \epsilon \delta_\alpha)(-f_n) - \xi$ and consider that any term $\frac{d^k}{d\epsilon^k} z$, $k \geq 2$ is zero, therefore it will not be written. As the same time, it can be reasonably assumed that the perturbation is zero at the border and as a consequence terms which are border-related are zero too. Hence it can be written:

$$\begin{aligned} \frac{d}{d\epsilon} \mathcal{V} &= \int d\mathbf{x} \left(\frac{d}{dz} \sigma_\alpha(z) \frac{d}{d\epsilon} z + \delta_\alpha(z) + \epsilon \frac{d}{dz} \delta_\alpha(z) \frac{d}{d\epsilon} z \right), \\ \frac{d^2}{d\epsilon^2} \mathcal{V} &= \int d\mathbf{x} \left(\frac{d^2}{dz^2} \sigma_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^2 + 2 \frac{d}{dz} \delta_\alpha(z) \frac{d}{d\epsilon} z + \right. \\ &\quad \left. + \frac{d^2}{dz^2} \delta_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^2 \right), \\ \frac{d^3}{d\epsilon^3} \mathcal{V} &= \int d\mathbf{x} \left(\frac{d^3}{dz^3} \sigma_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^3 + 2 \frac{d^2}{dz^2} \delta_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^2 + \right. \\ &\quad \left. + \frac{d^3}{dz^3} \delta_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^3 \right), \\ \frac{d^k}{d\epsilon^k} \mathcal{V} &= \int d\mathbf{x} \left(\frac{d^k}{dz^k} \sigma_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^k + 2 \frac{d^{k-1}}{dz^{k-1}} \delta_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^{k-1} + \right. \\ &\quad \left. + \frac{d^k}{dz^k} \delta_\alpha(z) \left(\frac{d}{d\epsilon} z \right)^k \right). \end{aligned} \quad (48)$$

Moreover, $\frac{d^n}{dz^n} \delta_\alpha(z) = -\frac{d^n}{dz^n} \sigma_\alpha(z)$ because of the definition of δ_α , so the first and third addenda for $k \geq 2$ simplify at every term. Let z' be equal to $z(\epsilon = 0) = \sum_n \sigma_\alpha(-f_n) - \xi$ and manipulate the following:

$$\frac{d}{d\epsilon} \mathcal{V} + \frac{d^2}{d\epsilon^2} \mathcal{V} \frac{1}{2} \bigg|_{\epsilon=0} = \int d\mathbf{x} \left(\delta_\alpha(z) + 2 \frac{d}{dz} \delta_\alpha(z) \frac{d}{d\epsilon} z \right). \quad (49)$$

Finally, as $\frac{d}{d\epsilon} z = \alpha \sum_n \delta(-f_n)$ and the index of the Taylor series is moved back, it results that

$$\delta\mathcal{V} = \delta_\alpha(z') - 2 \sum_{j=1}^{\infty} \frac{\sigma_\alpha^{(j)}(z')}{j!} \left(\alpha \sum_n \delta_\alpha(-f_n) \right)^j. \quad (50)$$

5.2 Custom D-dimensional volumes

Another interesting application of Heaviside is the possibility to represent customizable volumes. In fact, the technique presented in previous section is not limited just to analytical functions. Without loss of generality, consider $D = 2$. Suppose that this desired area is defined by a closed polygonal chain of N segments, each of them lying on a given line described by f_n (or the vertical line $x = c$, where c is a constant value). Clearly, $\Theta(f_n)$ is a binary representation of the inequality $f_n \geq 0$, because the equation $\Theta(f_n) = 1$ is satisfied only in a subspace of $\mathbb{R}^2$. Capitalizing the concept

of AND HSFG described in Section 2, the customizable area is cast with the following characteristic function:

$$\mathcal{X}_\Theta = \prod_{n=1}^N \Theta(f_n) = \Lambda[\{f_n\}]. \quad (51)$$

This concept does not rely on choice of f_n as these can be anything, as long as $\mathcal{X}_\Theta = 1$ encompasses a non-zero finite area.

Remark 4. Reminding results of Section 3, also for this application is possible to define an equivalent SF form $\mathcal{X}_\sigma$, taking advantage of Λ_σ .

Remark 5. It should be noticed that

$$\Omega(\{N(f_n)\}_n, \{\Lambda_q(\{f_m\}_{m(q)})\}_q) \quad \forall n, q, m(q), \quad (52)$$

represents a general partition of input domain into a certain number of disjointed sub-domains of arbitrary shape.

5.3 Custom D-dimensional borders

According to the formulation proposed so far, a further important result is achieved when the objective is not to calculate the volume but to have an expression of its border. Once again, it is sufficient to exploit the AND HSFG. It is necessary that $\vartheta_0 \neq 0$ to distinguish the locus of interest in the following manner:

$$\Gamma : \Theta(\mathcal{Y}_\Theta - \vartheta_0)\Theta(\vartheta_0 - \mathcal{Y}_\Theta) - \vartheta_0^2 = 0, \quad (53)$$

where $\mathcal{Y}_\Theta$ is the generic characteristic function which can be composed in one of the several way previous discussed. In practise, when two face-to-face Heavisides are multiplied the only geometrical locus that has non-zero value is the desired border. In other words, an equality constraint is imposed as will be seen in the next section. In particular, if $\vartheta_0 \neq \{0, 1\}$ then it is just needed:

$$\Gamma : \Theta(\mathcal{Y}_\Theta - \vartheta_0) - \vartheta_0 = 0. \quad (54)$$

However, it must be noticed that if $\mathcal{Y}_\Theta$ is composed of more than one function, and their borders intersect, then Eq.(53) and (54) are not satisfied in the intersections, except for Eq.(53) when $\vartheta_0 = 1$.

Another way to realize this locus is the substitution defined in (13) so that previous formulation is equivalent to:

$$\Gamma : \lim_{\alpha \rightarrow \infty} \sigma(\alpha(\mathcal{Y}_\sigma - 1/2)) - 1/2 = 0, \quad (55)$$

where $\mathcal{Y}_\sigma$ is the characteristic function expressed via SF. Still, the intersection are not counted. If the limit is not taken, an interesting behaviour arises in the proximity of intersection due to the analyticity of σ :

$$\Gamma_\alpha : \sigma(\alpha(\mathcal{Y}_\sigma - 1/2)) - 1/2 = 0. \quad (56)$$

For instance, consider the configuration in Fig.3. It consists of a square centered in $[2 \ 2]^T$ and a circle in $[1 \ 0]^T$ which intersect in $[1 \ 1]^T$. Therefore, the $\mathcal{Y}_\sigma$ in Eq.(56) is the sum of the characteristic functions of the square and the circle. The former is made with Eq.(51) and the usual substitution (13), while the latter is done with Eq.(32). As α increases, Γ_α gets closer to the real intersection. The presence of the error owed to α results to be negligible, and even advantageous:

Fig. 3: A custom border as function of α .

the proposed formulation represents an optimum analytical approximation of the union border, which in principle (i.e. the exact border) should have no defined gradient at the intersection points.

6 HSF PENALTY METHOD

In this Section, an alternative representation of constrained-optimization problems is proposed. This is possible thanks to binary logic granted by HSF. Let us have a cost function $\mathcal{A}(\mathbf{q})$ that has to be minimized and it is subject to some equality and inequality constraints

$$\begin{aligned} \min \mathcal{A}(\mathbf{q}) \\ h_j(\mathbf{q}) &= 0, \quad \forall j \in (1..\ell) \\ g_i(\mathbf{q}) &\leq 0, \quad \forall i \in (1..m). \end{aligned}$$

Let's pretend, for a while, that the minima of $\mathcal{A}$ are different from zero. This hypothesis does not represent a restriction and the lack of it will be handled hereafter. Indeed, usually the cost function has some practical and physical meaning, therefore those minima reasonably lie within a known range. Assuming also that $\vartheta_0 = 1$, it is possible to write the following expression:

$$\mathcal{F}(\mathbf{q}) = \mathcal{A}(\mathbf{q}) \prod_{i=1}^m \Theta(-g_i(\mathbf{q})) \prod_{j=1}^{\ell} \Theta(h_j(\mathbf{q})) \Theta(-h_j(\mathbf{q})). \quad (57)$$

As in Penalty Methods, the problem is reformulated as an unconstrained one by taking into account a so-called penalty function [27]. The proposed approach takes care of constraints by exploiting AND HSFG. Indeed, previous formulation is equivalent to:

$$\begin{aligned} \mathcal{F}(\mathbf{q}) &= \mathcal{A}(\mathbf{q}) \Lambda[\{N(g_i)\}] \Lambda[\{h_j\}] \Lambda[\{N(h_j)\}] \\ &= \mathcal{A}(\mathbf{q}) \bar{\Omega}[\{g_i\}] \Lambda[\{h_j\}] \bar{\Omega}[\{h_j\}]. \end{aligned} \quad (58)$$

As a matter of fact, the resultant function is zero across the whole space where they are not fulfilled, and it coincides with the cost function itself when they are satisfied. Moreover, each equality constraint is enabled through an AND HSFG having in input two face-to-face inequalities. The requirement $\vartheta_0 = 1$ which was stated before is now clear. With this assumption, gates return 1 when the weak inequality $g_i(\mathbf{q}) \leq 0$ is satisfied and it holds at the equality as well. As a consequence, it also allows the equality constraint $h_j(\mathbf{q})$

to actually work. Otherwise, in both cases zero is provided. This behaviour reproduces the binary logic where the two levels are 0 and $\mathcal{A}(\mathbf{q})$ itself. Thus, the usage of the first level has to be denied. Indeed, if some minima are expected to be zeroes of $\mathcal{A}$, it is just needed to shift $\mathcal{F}$ sending $\mathcal{A} \rightarrow \mathcal{A} + c$. Clearly, the minimal obtained values have to be shifted back after they have been found.

The composition $\mathcal{F}$ cannot be globally convex because any of its second-order partial derivatives involves the second derivative of SF with respect to its argument. Therefore, convex optimization methods are not as well-suited as Heuristic SAs. However, these could need to move in a smooth manner between the two spaces (in which the constraints are satisfied or not). To this aim, once again, substitution (13) is necessary and hence SFGs have to be used instead of corresponding HSF forms. In this manner, an exponential slope is given to $\mathcal{F}$, and it will depend on α . In particular, each equality constraint has to be multiplied by a factor 4 because the two σ of which is composed of would give 1/2 each when it is satisfied. The obtained function $\mathcal{F}$ and the optimization process are described by pseudo-code in Algorithm 2.

Algorithm 2 Optimization Process

Input: $\mathcal{A}, \mathbb{Q}, g, h$
Output: $\mathbf{q}^*$

- 1: $\text{SA}(\mathcal{F}, \mathbb{Q})$
- 2:
- 3: **function** $\mathcal{F}(\mathcal{A}, \mathbf{q} \in \mathbb{Q}, g, h)$
- 4: **if** $g_i(\mathbf{q}) < 0 \dots \text{and } g_m(\mathbf{q}) < 0 \text{ and } h_j(\mathbf{q}) = 0 \dots \text{and } h_\ell(\mathbf{q}) = 0$ **then**
- 5: **return** $\mathcal{A}(\mathbf{q})$
- 6: **end if**
- 7: **return** 0 ▷ or asymptotically equal to $\mathcal{A}(\mathbf{q})e^{-\alpha}$
- 8: **end function**

This method represents an alternative with respect to standard optimization, because the constraints are included in such a way that they are not constraints anymore, but a part of the whole function. Therefore, it is possible to avoid the check of the Karush-Kuhn-Tucker necessary conditions since they rely on the construction of the Lagrangian while $\mathcal{F}$ is formally unconstrained. Thus, computation of weights concerning the classical inclusion of constraints is avoided. Most importantly, as a final remark, this method allows constraints also to be put in any other logical relation.

6.1 Curves intersections

A useful geometrical application of HSF Penalty Method (HSFPM) proposed is to find how many and which intersections there are when two or more curves lie in the same locus. The methodology proposed hereby is suitable in the case of two curves, but can be iterated in case of more. Given f_1 and f_2 , $\mathbb{X} = \{x_i \in \mathbb{R} \forall i : 1..N_{MAX}\}$ and $\mathbb{Y} = \{y_i \in \mathbb{R} \forall i : 1..N_{MAX}\}$ where N_{MAX} is the maximum intersection number (see Bézout's Theorem [28]),

the optimization problem can be stated as follows

$$(P1) \max_{\mathbb{X}, \mathbb{Y}} \sum_{i=1}^{N_{MAX}} f_1(x_i, y_i)^2 + f_2(x_i, y_i)^2 \text{ s.t.} \\ \left\| [x_k \ y_k]^T - [x_j \ y_j]^T \right\| \leq \zeta \quad \forall k \in (1, N_{MAX} - 1) \\ \forall j \in (k + 1, N_{MAX})'$$

where ζ is a user-defined minimum distance to distinguish intersection points. This is a non-convex optimization problem that is hard to solve even with mathematical approximation. The technique proposed in this section can be used to reformulate the problem as:

$$(P2) \max_{\mathbb{X}, \mathbb{Y}} \sum_{i=1}^{N_{MAX}} \Theta(-f_i^*) \prod_{k < j}^{N_{MAX}} \mathcal{J}_{kj},$$

where

$$\mathcal{J}_{kj} = \Theta\left(\left\| [x_k \ y_k]^T - [x_j \ y_j]^T \right\| - \zeta\right), \\ f_i^* = f_1^2(x_i, y_i) + f_2^2(x_i, y_i).$$

A search algorithm can be used to solve this problem.

Basically, every intersection point corresponds to a maxima of the objective function. Moreover, a $\Theta(\cdot)$ has been introduced in the objective function derived from (P1) in order to assure that local maxima are avoided in (P2), and the function value will now be equal to the number of intersections which the algorithm finds. In fact, every other point which is not a real intersection would not change the discrete function value. But as it was previously remarked, $\Theta(\cdot)$ has to be substituted with $\sigma(\alpha \cdot)$ in order for the algorithm to decently move in the solution domain and α has to be relatively small. As a consequence, the points in excess would increase the function value in an undefined manner. So, it is necessary to introduce a new integer parameter N which counts the real intersections. As the previous sum and products now extends up to N , one more addendum is introduced as a penalty cost P for having increased N , but in such a way it must be convenient to pay it if a true intersection is found. To this aim it is necessary to also multiply the objective function for a weight W , such that:

$$W\sigma(-\alpha\zeta) < P. \quad (59)$$

The worst case scenario is when the algorithm tries to count a point at a distance ζ from a true intersection as if it was a real one. The requirement in Eq.(59) prevents such unwanted behaviour, as it imposes that when a fake intersection is found the reward gained has to be lower than inflicted penalty. For instance, to achieve this goal, (P2) can be reformulated as:

$$(P3) \max_{\mathbb{X}, \mathbb{Y}, N} 2N_{MAX} \sum_{i=1}^N \sigma(-\alpha f_i^*) \prod_{k < j}^N \mathcal{J}_{kj} - (N_{MAX} - N).$$

where $\mathcal{J}_{kj}$ is now made with the substitution (13). Despite aforementioned solution, others can be investigated.

7 NUMERICAL EVALUATION

In this Section, theoretical results discussed in past sections are evaluated. In particular, a wide experimental campaign has been carried out to analyze each possible application

derived from discussed previous notions. Genetic Algorithm implemented in MATLAB R2020a is used as search algorithm on a desktop computer equipped with Intel i7 970 and 16GB RAM. For the sake of generality, measurement units are simplified to pure numbers.

7.1 Volumes measure error analysis

In order to evaluate results obtained in Section 5.1, a reference scenario is taken into account. Given the general equation of a circle

$$\mathcal{C}(x_0, y_0, r) : (x - x_0)^2 + (y - y_0)^2 - r^2 = 0, \quad (60)$$

the aim is to compute the union area of $\mathcal{C}_1 = \mathcal{C}(-1, 0, 1)$ and $\mathcal{C}_2 = \mathcal{C}(1, 0, 1)$, which are external tangent, placed in a mesh 4×2 . This task can be done exploiting Eq.(41):

$$\int_{\mathcal{M}} \Omega[N(\mathcal{C}_1), N(\mathcal{C}_2)](\mathbf{x}) d\mathbf{x} = \quad (61)$$

$$= \int_{-1}^1 \int_{-2}^2 \Theta(\Theta(-((x+1)^2 + y^2 - 1)) + \Theta(-((x-1)^2 + y^2 - 1)) - 0.1) dx dy. \quad (62)$$

First, the integral has been calculated in MATLAB through the Riemann Summation, with a mesh precision $\rho = 10^{-4}$. The exact result of the area should be $2\pi R^2 \simeq 6.28318$ but, due to the precision used, the experimental one is $\simeq 6.28313$. This evaluation considers HSFGs that have a discontinuity in 0 that does not represent a problem as stated in Remark 2. However, when SFGs are used, α has a non-negligible impact on the error as discussed in Section 5.1. As expected

Fig. 4: Comparison between experimental error and theoretical one in a reference scenario.

Fig.4 shows that, fixed the mesh precision, the error of $\sigma(\cdot)$ with respect to $\Theta(\cdot)$ decreases as α increases. Moreover, the first two orders shown demonstrate that (i) the high-order derivatives accounts for lower α error (ii) for $\alpha \geq 10$ the error is negligible.

7.2 Custom volume measure

Thanks to this analysis and results obtained in Section 5.2 it is possible to solve problem in the scenario proposed in Section 5. First, the function that describes the Square #1, namely $\mathcal{S}_1$, centered in $[3 \ 3]^T$, can be easily derived from Eq.(51):

$$\mathcal{S}_1 = \Lambda_\alpha [x - 1, -(x - 5), y - 1, -(y - 5)]. \quad (63)$$

Fig. 5: Intersection of $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$.

Similarly, it is possible to build the other two squares $\mathcal{S}_2$ and $\mathcal{S}_3$ thus obtaining an integrand shown in Fig.5. As previously mentioned, the edges of the volume have a value of $1/2$ that is exactly $\sigma(0)$. Finally, exploiting Eq.(41), with $\rho = 10^{-3}$, $\alpha = 10^3$ and a mesh 8×8 , the calculated area is 29.99998.

7.3 HSF Penalty Method

To probe results of Section 6 two simulations of different complexity have been done.

The first involves two fixed circles $\mathcal{C}_1 = \mathcal{C}(-2, 0, 1)$ and $\mathcal{C}_2 = \mathcal{C}(2, 0, 1)$ placed in a 8×8 mesh. A third one, defined as $\mathcal{C}_3 = \mathcal{C}(x, y, 1)$ has to be positioned in order to maximize the total covered area. Taking advantage of Eq.(41) and (13) $\mathcal{V}_\sigma$ is defined as the SF variant of the Heaviside's one. The task is accomplished by the objective function:

$$\mathcal{V}_\sigma[\{\mathcal{C}_n\}] \quad \forall n \in (1, 3), \quad (64)$$

where $\alpha = 10^3$ and $\rho = 5 \times 10^{-2}$. The locus in which the center $[x \ y]^T$ of $\mathcal{C}_3$ has to be placed to maximize the total covered area is shown in Fig.6. As expected, the center should not be in proximity of the two pre-positioned circles.

Fig. 6: Total covered area as a function of $\mathcal{C}_3$ center.

Fig. 9: Intersection points of two circles found via GA.

Fig. 10: Intersection points of a circle and a triangle found via GA.

with custom volumes and, specifically, for what concern oblique segments. On a trial basis, outcomes of simulations confirm that α related to these pieces of volumes should be set not greater than 10 with the aim of speed up the search algorithm operations.

8 CONCLUSION

The present work discussed a mathematical characterization of binary logic granted by the HSF. Then, an analytical representation through logistic function has been studied. Furthermore, the existence of a correlation between the conceived mathematical structure and a finite-time program has been proved.

Based on these concepts, different geometrical applications and a Penalty Method for constrained optimization problems has been deeply investigated. First, the measure of binary operations among D-volumes is presented. These can also be designed in a customer manner so that a compact region is encompassed by a set of loci. Hence, it is even possible to obtain the border analytical expression of the desired D-volume. Finally, a novel Penalty Method has been

realized allowing the reformulation of a constrained non-convex optimization problem in an unconstrained one.

A wide simulation campaign demonstrated the effectiveness of proposed methodologies which can be employable in many engineering areas such as computer vision and graphics, medical imaging, robotics and automation.

However, several aspects can be further investigated. First of all, a solid correspondence between control structures and proposed HSFGs/SFGs will be addressed. Then, a method to compute the border length of a custom volume will be probed. Current formulation to find intersections could also be improved. Finally, the proposed Penalty Method will be tested with more SAs to settle a proper benchmark with others solutions.

APPENDIX A

The family of functions $\{\delta_\alpha | \alpha > 0\}$ pointwise converges [29] in $\mathbb{R}$ for $\alpha \rightarrow +\infty$ to

$$\delta_\infty(x) = \begin{cases} 0 & \text{if } x \neq 0 \\ \vartheta_0 - \frac{1}{2} & \text{if } x = 0. \end{cases} \quad (66)$$

Therefore, it means that $\{\sigma(\alpha \cdot)\}_\alpha$ pointwise converges in $\mathbb{R}^*$ to $\Theta(\cdot)$ and in $\mathbb{R}$ to $\Theta(\cdot)$ only if $\vartheta_0 = 1/2$. Instead $\{\delta_\alpha\}_\alpha$ uniformly converges in $A_1 =]-\infty, -a] \cup [a, +\infty[\forall a > 0$ for $\alpha \rightarrow +\infty$ to δ_∞ , meaning that uniform convergence of $\{\sigma(\alpha \cdot)\}_\alpha$ to $\Theta(\cdot)$ is only achieved in the same interval. This phenomenon propagates up to Eq.(4), and the following explanation tells how. $\vec{\sigma}_{0,\alpha}$ converges to $\vec{\Theta}_0$ in a component-wise manner and with the same caveats as before, therefore the convergence of the whole vector function can be established with the metric $d := \sum_i d_{\infty,i}(\cdot, \cdot)$ in $A = A_1^D$. Consider a composition of the form $\mathcal{H}^1 = g_1 \circ \vec{\Theta}$ and the family $\mathcal{H}_\alpha^1 = g_1 \circ \vec{\sigma}_{0,\alpha}$. Define $L = \vec{\sigma}_{0,\alpha}(A)$ and its closure $\bar{L}$, then the uniform continuity of g_1 means

$$\forall \epsilon > 0 \exists \delta > 0 \exists' \forall y_1, y_2 \in \bar{L}, \\ d(y_1, y_2) < \delta : d(g_1(y_1), g_1(y_2)) < \epsilon,$$

while definition of uniform convergence of $\vec{\sigma}_{1,\alpha}$ in A :

$$\exists N \exists' \forall \alpha \geq N : d(\vec{\sigma}_{0,\alpha}(\mathbf{x}), \vec{\Theta}_0(\mathbf{x})) < \delta \forall \mathbf{x} \in A.$$

So if $\mathbf{x} \in A$, $y_1 = \vec{\sigma}_{0,\alpha}(\mathbf{x}) \in \bar{L}$ and $y_2 = \vec{\Theta}_0(\mathbf{x}) \in \bar{L}$. Thus, when $\alpha \geq N$

$$d(g_1(\vec{\sigma}_{0,\alpha}(\mathbf{x})), g_1(\vec{\Theta}_0(\mathbf{x}))) < \epsilon \forall \mathbf{x} \in A. \quad (67)$$

Therefore $\mathcal{H}_\alpha^1$ converges uniformly in A to $\mathcal{H}^1$.

Without loss of generality, the next composition will be executed with a scalar σ_α . Now, consider some $\mathcal{H}_\alpha^1 : \mathbb{R}^D \rightarrow \mathbb{R}$ and $\sigma_\alpha : \mathbb{R} \rightarrow]0, 1[$. Define a subset X of the domain of $\mathcal{H}_\alpha^1$ whose closure $\bar{X}$ is contained in the complementary of $K = \{\mathbf{x} \in \mathbb{R}^D | \mathcal{H}_\alpha^1(\mathbf{x}) = 0\}$ with respect to A , and define $U = \mathcal{H}_\alpha^1(X)$. Let $\epsilon > 0$ be fixed again, and choose $\epsilon/2$.

$$\exists c_1 \in \mathbb{R} \exists' \forall \alpha_1 \geq c_1 : d(\sigma_{\alpha_1}(z), \Theta(z)) < \epsilon/2 \forall z \in A_1, \quad (68)$$

$$\exists \delta > 0 \exists' \forall z_1, z_2 \in U, \\ d(z_1, z_2) < \delta : d(\Theta(z_1), \Theta(z_2)) < \epsilon/2, \quad (69)$$

$$\exists c_2 \in \mathbb{R} \exists' \forall \alpha_2 \geq c_2 : d(\mathcal{H}_{\alpha_2}^1(\mathbf{x}), \mathcal{H}^1(\mathbf{x})) < \delta \forall \mathbf{x} \in X. \quad (70)$$

Eq.(68) is the uniform convergence of σ_α , (69) is the uniform continuity of Θ that is granted in U , and (70) is the uniform convergence of $\mathcal{H}_\alpha^1$ to $\mathcal{H}^1$ that has been previously proved. Take $c = \max\{c_1, c_2\}$ and $\alpha \geq c$. Then, if $\mathbf{x} \in X$, $z_1 = \mathcal{H}_\alpha^1(\mathbf{x}) \in U$ and $z_2 = \mathcal{H}^1(\mathbf{x}) \in U$. Thus, the following is verified:

$$d(\sigma_\alpha(\mathcal{H}_\alpha(\mathbf{x})), \Theta(\mathcal{H}^1(\mathbf{x}))) \leq d(\sigma_\alpha(\mathcal{H}_\alpha^1(\mathbf{x})), \Theta(\mathcal{H}_\alpha^1(\mathbf{x}))) + d(\Theta(\mathcal{H}_\alpha^1(\mathbf{x})), \Theta(\mathcal{H}^1(\mathbf{x}))) < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon \quad \forall \mathbf{x} \in X. \quad (71)$$

So, it results that $\sigma_\alpha \circ \mathcal{H}_\alpha^1$ uniformly converges to $\Theta \circ \mathcal{H}^1$ in X . This generalises in the following conclusion: the composition which defines any SFC $\mathcal{H}_\alpha^R$ in Eq.(16) uniformly converges to its correspondent in Eq.(4) in any subset X of $\mathbb{R}^D$ whose closure $\bar{X} \subset \mathbb{R}^D \setminus \mathcal{K}$, with $\mathcal{K}$ being the union of kernels defined in Eq.(18).

ACKNOWLEDGMENTS

The authors would like to thank Dr. Pietro Boccadoro for his valuable and constructive suggestions. Special thanks also go to Prof. Monica Lazzo for revising the mathematical part.

REFERENCES

- [1] Y. N. Moschovakis, "What is an algorithm?" in *Mathematics unlimited—2001 and beyond*. Springer, 2001, pp. 919–936.
- [2] S. Bauer-Mengelberg, "Review: Kurt Godel, b. meltzer, on formally undecidable propositions of principia mathematica and related systems i," *Journal of Symbolic Logic*, vol. 30, no. 3, pp. 359–362, 09 1965.
- [3] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *J. of Math*, vol. 58, no. 345–363, p. 5, 1936.
- [4] J. E. Savage, *Models of computation: Exploring the Power of Computing*. Addison-Wesley Reading, MA, 1998, vol. 136.
- [5] C. Böhm and G. Jacopini, "Flow diagrams, turing machines and languages with only two formation rules," *Communications of the ACM*, vol. 9, no. 5, pp. 366–371, 1966.
- [6] C. Zhou, Z. Chen, and M. Li, "A parallel method to accelerate spatial operations involving polygon intersections," *International Journal of Geographical Information Science*, vol. 32, no. 12, pp. 2402–2426, 2018.
- [7] S. C. Park and H. Shin, "Polygonal chain intersection," *Computers & Graphics*, vol. 26, no. 2, pp. 341–350, 2002.
- [8] F. Chin *et al.*, "Optimal algorithms for the intersection and the minimum distance problems between planar polygons," *IEEE Transactions on Computers*, vol. 12, pp. 1203–1207, 1983.
- [9] D. P. Dobkin and D. G. Kirkpatrick, "Fast detection of polyhedral intersection," *Theoretical Computer Science*, vol. 27, no. 3, pp. 241–253, 1983.
- [10] B. Žalik, "Two efficient algorithms for determining intersection points between simple polygons," *Computers & Geosciences*, vol. 26, no. 2, pp. 137–151, 2000.
- [11] F. Martínez, A. J. Rueda, and F. R. Feito, "A new algorithm for computing boolean operations on polygons," *Computers & Geosciences*, vol. 35, no. 6, pp. 1177–1185, 2009.
- [12] Y. Gao, B. Wu, J. Luo, and H. Qiu, "Gpu-based arbitrary polygon intersection area algorithm," *DEStech Transactions on Engineering and Technology Research*, 2017.
- [13] Y. K. Liu, X. Q. Wang, S. Z. Bao, M. Gomboši, and B. Žalik, "An algorithm for polygon clipping, and for determining polygon intersections and unions," *Computers & Geosciences*, vol. 33, no. 5, pp. 589–598, 2007.
- [14] J. J. Scheins, F. Boschen, and H. Herzog, "Analytical calculation of volumes-of-intersection for iterative, fully 3-d pet reconstruction," *IEEE Transactions on Medical Imaging*, vol. 25, no. 10, pp. 1363–1369, 2006.
- [15] C. Liu, S. Yu, and M. Yim, "A fast configuration space algorithm for variable topology truss modular robots," in *2020 International Conference on Robotics and Automation (ICRA)*, Paris, France, May 2020.
- [16] Y. J. Kim, M. C. Lin, and D. Manocha, *Collision Detection*. Dordrecht: Springer Netherlands, 2019, pp. 1933–1956.
- [17] J. Xu, Z. Liu, C. Zhang, C. Yang, and Y. Pei, "Minimal distance calculation between the industrial robot and its workspace based on circle/polygon-slices representation," *Applied Mathematical Modelling*, vol. 87, pp. 691 – 710, 2020.
- [18] J. Xu, Z. Liu, C. Yang, L. Li, and Y. Pei, "A pseudo-distance algorithm for collision detection of manipulators using convex-plane-polygons-based representation," *Robotics and Computer-Integrated Manufacturing*, vol. 66, p. 101993, 2020.
- [19] B. Zhou, Y.-J. Chiang, and C. Yap, "Soft subdivision motion planning for complex planar robots," *Computational Geometry*, vol. 92, p. 101683, 2021.
- [20] M. El-Salamony and A. Guaily, "Enhanced modified-polygon method for point-in-polygon problem," in *Recent Advances in Engineering Mathematics and Physics*, M. H. Farouk and M. A. Hassanein, Eds. Cham: Springer International Publishing, 2020, pp. 47–61.
- [21] S. S. Rao, *Engineering Optimization: Theory and Practice*, 5th ed. John Wiley & Sons, 2019.
- [22] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., ser. Global Edition. Pearson Education, 2016.
- [23] J. H. Holland and J. S. Reitman, "Cognitive systems based on adaptive algorithms," in *Pattern-directed inference systems*. Elsevier, 1978, pp. 313–329.
- [24] Ö. Yeniyay, "Penalty function methods for constrained optimization with genetic algorithms," *Mathematical and computational Applications*, vol. 10, no. 1, pp. 45–56, 2005.
- [25] N. Kyurkchiev and S. Markov, "Sigmoid functions: some approximation and modelling aspects," *LAP LAMBERT Academic Publishing, Saarbrücken*, 2015.
- [26] A. Iliev, N. Kyurkchiev, and S. Markov, "On the approximation of the step function by some sigmoid functions," *Mathematics and Computers in Simulation*, vol. 133, pp. 223–234, 2017.
- [27] A. E. Smith and D. W. Coit, "Constraint-handling techniques - penalty functions," *Handbook of Evolutionary Computation*, pp. C5.2:1–C5.2:6, 1997.
- [28] L. Blum, F. Cucker, M. Shub, and S. Smale, *Bézout's Theorem*. New York, NY: Springer New York, 1998, pp. 187–200.
- [29] V. A. Zorich, *Mathematical analysis II*. Springer, 2016.

Giovanni Iacovelli Giovanni Iacovelli received the Dr. Eng. degree (with honors) in information engineering from Politecnico di Bari, Bari, Italy, in July 2019. Since November 2019 he is a Ph.D. Student at the Department of Electrical and Information Engineering, Politecnico di Bari.

Claudio Iacovelli Claudio Iacovelli received the Bachelor's degree (with honors) in Physics from Università degli Studi di Bari, Bari, Italy, in September 2019. Since October 2019 he is a Master's degree student in Physics at Università degli Studi di Trento.