There and Back Again: Split and Prune to Tighten

Raazesh Sainudiin*†, Jennifer Harlow* and Warwick Tucker‡
†Laboratory for Mathematical Statistical Experiments & *Department of Mathematics and Statistics,
Private Bag 4800, University of Canterbury, Christchurch 8041, New Zealand.
*Email: jenny.harlow@canterbury.ac.nz, †Email: r.sainudiin@math.canterbury.ac.nz
‡Department of Mathematics, Uppsala University, Box 480, 751 06 Uppsala, Sweden
‡Email: warwick@math.uu.se

***Abstract*—A regular paving is a finite succession of bisections that partition a root box $\boldsymbol{x}$ in $\mathbb{R}^d$ into sub-boxes using a binary tree-based data structure. The sequence of splits that generate such a partition is given by the sub-boxes associated with the nodes of the tree. The leaf boxes, i.e., the sub-boxes associated with the leaf nodes, form a partition of $\boldsymbol{x}$. We provide algorithms to tightly enclose the range of a function $g : \boldsymbol{x} \to \mathbb{R}$ using its interval extension $\boldsymbol{g}$. Our idea is to (i) refine the regular paving partition of the domain $\boldsymbol{x}$ by splitting the leaf boxes, (ii) obtain range enclosures of g over them, (iii) propagate the range enclosures of the leaf boxes up the internal nodes of the tree and finally (iv) prune back the leaves to get a coarser partition with fewer leaf boxes but with tighter range enclosures. This approach allows one to obtain tighter range enclosures for interval inclusion functions.**

I. Introduction

Many operations with fuzzy numbers can be naturally implemented as operations with their α-cuts (i.e., intervals). A characterizing function $\xi_{\tilde{\boldsymbol{x}}}(x) : \mathbb{R} \to [0,1]$ of a fuzzy number $\tilde{\boldsymbol{x}}$ has the following property: $\forall \alpha \in (0,1]$ the set $\tilde{\boldsymbol{x}}^\alpha := \{x \in \mathbb{R} : \xi_{\tilde{\boldsymbol{x}}}(x) \geq \alpha\}$, the α-cut of $\tilde{\boldsymbol{x}}$, is a non-empty compact interval in $\mathbb{IR} := \{\boldsymbol{x} = [\underline{x}, \overline{x}] : \underline{x} \leq \overline{x}, (\underline{x}, \overline{x}) \in \mathbb{R}^2\}$. Since the α-cuts of a fuzzy number are intervals, it is natural to use interval arithmetic over $\mathbb{IR}$ to perform arithmetic with fuzzy numbers [1]. An interval vector or box $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_d) \in \mathbb{IR}^d$ can be used to represent the α-cuts of d fuzzy numbers in $\tilde{\boldsymbol{x}}^\alpha = (\tilde{\boldsymbol{x}}_1^\alpha, \tilde{\boldsymbol{x}}_2^\alpha, \ldots, \tilde{\boldsymbol{x}}_d^\alpha)$. In many situations we are interested in enclosing $g(\tilde{\boldsymbol{x}}^\alpha)$, the range of a function $g : \mathbb{R}^d \to \mathbb{R}$ over the indistinctness induced by $\tilde{\boldsymbol{x}}^\alpha$. Once again interval arithmetic can be used to enclose $g(\tilde{\boldsymbol{x}}^\alpha)$ by using an interval inclusion function of g given by $\boldsymbol{g} : \mathbb{IR}^d \to \mathbb{IR}$ where all real arithmetic operations in g are replaced by their interval counterparts. Such a naive range enclosure using interval arithmetic can be pessimistic and produce over enclosures of the range. In this article we introduce some novel algorithms to get tighter range enclosures using a space partitioning data structure based on binary trees.

Hierarchical data structures, such as trees, are commonly used to represent and organise multi-dimensional data. A wide range of such space partitioning trees have been developed to suit particular types of data and to meet the needs of different applications or uses of that data [2], [3]. A regular paving [4], [5] is a finite succession of bisections that partition a box $\boldsymbol{x}$ in $\mathbb{R}^d$ into sub-boxes using a tree-based data structure. The sequence of splits that generate such a partition is given by the sub-boxes associated with the nodes of the tree. The leaf boxes, i.e., the sub-boxes associated with the leaf nodes, form a partition of $\boldsymbol{x}$.

In this article we exploit the tree-based structure of regular pavings to calculate interval enclosures of real-valued functions with an interval inclusion function efficiently using operations on trees. We provide algorithms to tightly enclose the range of a function $g : \boldsymbol{x} \to \mathbb{R}$ using its interval extension $\boldsymbol{g}$ for a final specified partition size. Our idea is to (i) refine the regular paving partition of the domain $\boldsymbol{x}$ by splitting the leaf boxes, (ii) obtain range enclosures of g over them, (iii) propagate the range enclosures of the leaf boxes up the internal nodes of the tree and finally (iv) prune back the leaves to get a coarser partition with fewer leaf boxes but with tighter range enclosures. This approach allows one to obtain tighter range enclosures for interval inclusion functions while limiting the final size of the partition within a machine's internal memory constraints.

We briefly illustrate the core idea in this paper using the example in Figure 1. We show the range enclosures (dark gray boxes in Figure 1) over a regularly paved partition of the domain $[0.5, 1]$ for an oscillating function (curve in Figure 1) by doing a prioritized split up to 45 leaf boxes. Here we use the difference in the upper and lower Riemann sum over each leaf box (i.e., the enclosure area of the gray box over each leaf box in Figure 1) as the splitting priority so that the leaf box with the largest enclosure area is the next one to be split. We then continue splitting five more times until there are 50 leaf boxes. Then we propagate the range enclosures of these 50 leaf boxes up the internal nodes of the tree and thus pass information about the range from the finer partition to coarser ones. Finally, using the same priority rule to optimize the enclosure area, we prune back the leaves to get a range enclosure (light gray boxes in Figure 1) based on a coarser partition with 45 leaf boxes that are typically different from those obtained earlier by just splitting to 45 leaves. The tighter range enclosure that is still based on only 45 leaf boxes is shown in Figure 1 as the light gray boxes — from *having been there* to 50 leaves *and back again* to 45 leaves — that are superimposed on the dark gray boxes with 45 leaves.

This article is organized as follows: In Section II we describe regular pavings and in Section III we introduce $\mathbb{IR}$-mapped regular pavings and the algorithms to enclose the range of a real valued function using its interval inclusion function. We conclude in Section IV.

II. Regular Pavings

Let $\boldsymbol{x} := [\underline{x}, \overline{x}]$ be a compact real interval with lower bound $\underline{x}$ and upper bound $\overline{x}$ where $\underline{x} \leq \overline{x}$. Let the space of such intervals be $\mathbb{IR}$. We can define the width and midpoint of

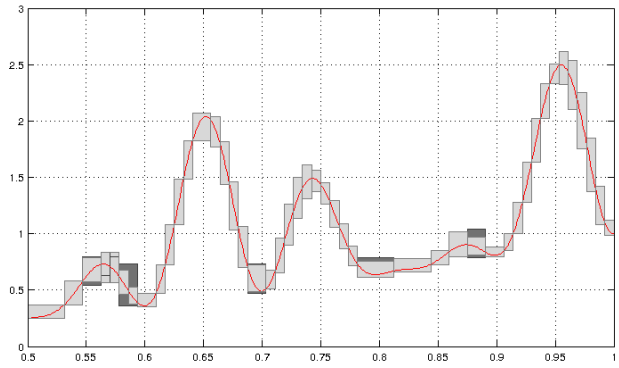


Fig. 1. The function $g(x) = x^2 + (x+1)\sin(10\pi x)^2\cos(3\pi x)^2$ over $x \in [1/2, 1]$ (red curve) with tighter range enclosures (light gray rectangles) superimposed on possibly wider range enclosures (dark gray rectangles) over two partitions of $[1/2, 1]$ made of 45 subintervals.

an interval $\boldsymbol{x}$ as $\mathrm{wid}\,(\boldsymbol{x}) := \overline{x} - \underline{x}$ and $\mathrm{mid}\,(\boldsymbol{x}) := \frac{\underline{x}+\overline{x}}{2}$, respectively. We can also define a box of dimension d with coordinates in $\Delta := \{1, 2, \ldots, d\}$ as an interval vector

$$\boldsymbol{x} := [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_d, \overline{x}_d] =: \bigboxtimes_{j\in\Delta} [\underline{x}_j, \overline{x}_j] \ .$$

Let $\mathbb{IR}^d$ be the set of all such boxes. Consider a box $\boldsymbol{x}$ in $\mathbb{IR}^d$. Let the index ι be the first coordinate of maximum width, i.e.

$$\iota = \min\left(\underset{i}{\mathrm{argmax}}(\mathrm{wid}\,(\boldsymbol{x}_i))\right) \ .$$

A *bisection* or *split* of $\boldsymbol{x}$ perpendicularly at the mid-point along this first widest coordinate ι gives us the left and right child boxes of $\boldsymbol{x}$ as follows:

$$\boldsymbol{x}_{\mathsf{L}} := [\underline{x}_1, \overline{x}_1]\times\ldots\times[\underline{x}_\iota, \mathrm{mid}\,(\boldsymbol{x}_\iota)]\times[\underline{x}_{\iota+1}, \overline{x}_{\iota+1}]\times\ldots\times[\underline{x}_d, \overline{x}_d] \ ,$$

$$\boldsymbol{x}_{\mathsf{R}} := [\underline{x}_1, \overline{x}_1]\times\ldots\times[\mathrm{mid}\,(\boldsymbol{x}_\iota), \overline{x}_\iota]\times[\underline{x}_{\iota+1}, \overline{x}_{\iota+1}]\times\ldots\times[\underline{x}_d, \overline{x}_d] \ .$$

Such a bisection is said to be *regular*. A recursive sequence of selective regular bisections of boxes along the first widest coordinate, starting from the root box $\boldsymbol{x}$ in $\mathbb{IR}^d$ is known as a *regular paving (RP)* [5] or n-tree [4] of $\boldsymbol{x}$.

An RP of $\boldsymbol{x}$ can also be seen as a binary tree formed by recursively bisecting the box $\boldsymbol{x}$ at the root node. Each node in the binary tree has either no children or two children. These trees are known as *plane binary trees* in enumerative combinatorics [6, Ex. 6.19(d), p. 220] and as *finite, rooted binary trees (frb-trees)* in geometric group theory [7, Ch. 10]. When the root box $\boldsymbol{x}$ is clear from the context we refer to an RP of $\boldsymbol{x}$ as merely an RP. Each node of an RP is associated with a sub-box of the root box that can be attained by a sequence of selective regular bisections.

Each node in an RP is distinctly labeled by the sequence of child node selections from the root node. We label these nodes and the associated boxes with strings composed of L and R for left and right, respectively.

We illustrate the relationship of trees, labels and partitions in Figure 2 with a simple 1-dimensional example. The root node associated with a 1-dimensional root interval $\boldsymbol{x}_\rho$ is labeled ρ. First, we split ρ into two child nodes and denote it

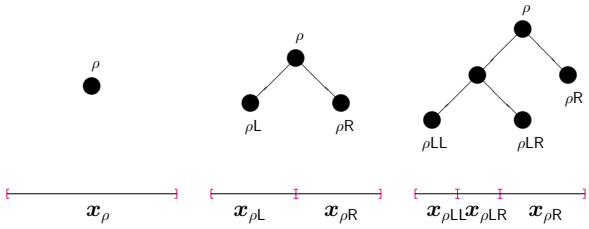


Fig. 2. A sequence of selective bisections of boxes (nodes) along the first widest coordinate, starting from the root box (root node), produces an RP.

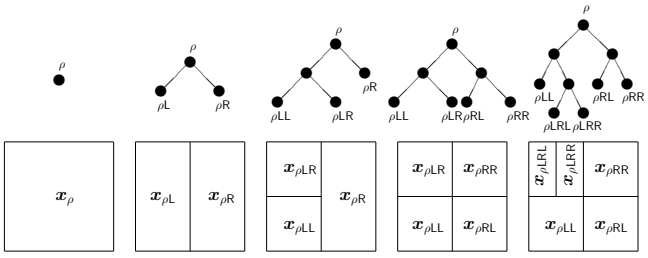


Fig. 3. A sequence of selective bisections of boxes (nodes) along the first widest coordinate, starting from the root box (root node), produces an RP.

by $\bigtriangledown(\rho) = \{\rho\mathsf{L}, \rho\mathsf{R}\}$. These left child and right child nodes are labeled by $\rho\mathsf{L}$ and $\rho\mathsf{R}$, respectively. The left half of $\boldsymbol{x}_\rho$ that is now associated with node $\rho\mathsf{L}$ is denoted by $\boldsymbol{x}_{\rho\mathsf{L}}$. Similarly, the right half of $\boldsymbol{x}_\rho$ that is associated with the right child node $\rho\mathsf{R}$ is denoted by $\boldsymbol{x}_{\rho\mathsf{R}}$. We say $\rho\mathsf{L}$ and $\rho\mathsf{R}$ are a pair of *sibling nodes* since they share the same parent node ρ. A node with no child nodes is called a *leaf node*. A *cherry node* is a sub-terminal node with a pair of child nodes that are both leaves. This pair of sibling nodes can be *reunited* or *merged* to its parent cherry node ρ, thereby turning the cherry node into a leaf node. Such a merging operation is denoted by $\bigtriangleup(\rho\mathsf{L}, \rho\mathsf{R}) = \rho$.

Returning to Figure 2, let us further split the left node $\rho\mathsf{L}$ to get its left and right child nodes $\rho\mathsf{LL}$ and $\rho\mathsf{LR}$ with associated sub-intervals $\boldsymbol{x}_{\rho\mathsf{LL}}$ and $\boldsymbol{x}_{\rho\mathsf{LR}}$ respectively, formed by the bisection of interval $\boldsymbol{x}_{\rho\mathsf{L}}$. Because the root interval $\boldsymbol{x}_\rho$ is 1-dimensional, each bisection is always on that single coordinate.

Figure 3 shows a sequence of bisections of a square (2-dimensional) root box. We start with the same sequence as in Figure 2, so the first three trees are identical to those in Figure 2 but in Figure 3 we can see the effect of always bisecting on the first widest coordinate. The first bisection, forming sub-boxes $\boldsymbol{x}_{\rho\mathsf{L}}$ and $\boldsymbol{x}_{\rho\mathsf{R}}$, takes place on the first widest coordinate of $\boldsymbol{x}_\rho$, which is the first coordinate because the box is square. The next bisection, of box $\boldsymbol{x}_{\rho\mathsf{L}}$ to form $\boldsymbol{x}_{\rho\mathsf{LL}}$ and $\boldsymbol{x}_{\rho\mathsf{LR}}$, takes place on the second coordinate because this is the first widest coordinate of $\boldsymbol{x}_{\rho\mathsf{L}}$.

We then extend the sequence with two further bisections. First we split the right child node $\rho\mathsf{R}$ into its child nodes $\rho\mathsf{RL}$ and $\rho\mathsf{RR}$, respectively (again bisecting on the second coordinate of $\boldsymbol{x}_{\rho\mathsf{R}}$). Then we select $\rho\mathsf{LR}$ to do a final split and obtain its child nodes $\rho\mathsf{LRL}$ and $\rho\mathsf{LRR}$. $\boldsymbol{x}_{\rho\mathsf{LR}}$ is square and is bisected on its first coordinate to form the sub-boxes $\boldsymbol{x}_{\rho\mathsf{LRL}}$ and $\boldsymbol{x}_{\rho\mathsf{LRR}}$.

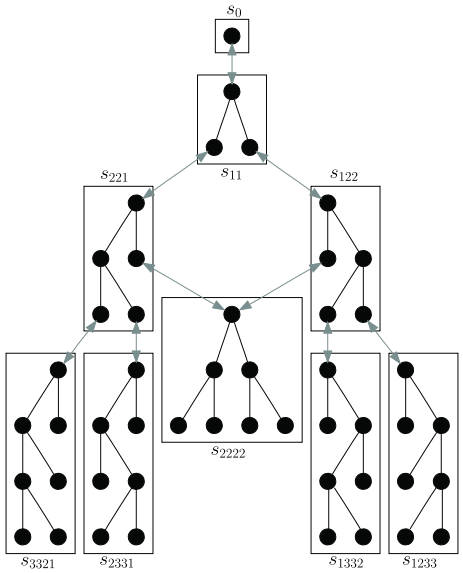


Fig. 4. Transition diagram over $\mathbb{S}_{0:3}$ with split/reunion transitions from one RP state to another.

Let the j-th interval of a box $\boldsymbol{x}_{\rho\mathsf{v}}$ be $[\underline{x}_{\rho\mathsf{v},j}, \overline{x}_{\rho\mathsf{v},j}]$. Then the volume of a d-dimensional box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with the node $\rho\mathsf{v}$ of an RP of $\boldsymbol{x}_\rho$ is the product of the side-lengths of the box, i.e.

$$\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right) = \prod_{j=1}^{d} (\overline{x}_{\rho\mathsf{v},j} - \underline{x}_{\rho\mathsf{v},j}) \ .$$

The volume may also be associated with the *depth* of a node. A node has depth k if it can be reached by k splits from the root node. Then, the volume of any d-dimensional box $\boldsymbol{x}_{\rho\mathsf{v}}$ associated with node $\rho\mathsf{v}$ having depth k is $\mathrm{vol}\left(\boldsymbol{x}_{\rho\mathsf{v}}\right) = 2^{-k}\mathrm{vol}\left(\boldsymbol{x}_\rho\right)$. This is because we will always split a box exactly in half.

We use the nodes of the final RP in Figure 3 for illustration purposes. Assume that the root box $\boldsymbol{x}_\rho$ is a unit hypercube. Then the root node ρ has depth 0 and $\mathrm{vol}\left(\boldsymbol{x}_\rho\right) = 1$, the nodes $\rho\mathsf{L}$ and $\rho\mathsf{R}$ have depth 1 and volume 2^{-1}, the nodes $\rho\mathsf{LL}, \rho\mathsf{LR}, \rho\mathsf{RL}, \rho\mathsf{RR}$ have depth 2 and volume 2^{-2}, and finally the nodes $\rho\mathsf{LRL}$, $\rho\mathsf{LRR}$ have depth 3 and volume 2^{-3}.

We can now label each leaf node of a tree by its depth. The leaf nodes of the final RP in Figure 3, listed in left-right order, are $[\rho\mathsf{LL}, \rho\mathsf{LRL}, \rho\mathsf{LRR}, \rho\mathsf{RL}, \rho\mathsf{RR}]$. We can then also uniquely identify or label an RP with its *ordered leaf-depth string*. The final RP in Figure 3 has 23322 as its *ordered leaf-depth string*. Thus this RP can be denoted by s_{23322}.

We denote the label set of all nodes of a regular paving by $\mathbb{V} := \rho \cup \{\rho\{\mathsf{L},\mathsf{R}\}^j : j \in \mathbb{N}\}$ and the set of all leaf nodes of a regular paving by $\mathbb{L}$. For the final regular paving in the sequence represented in Figure 3, $\mathbb{L}(s_{23322}) = \{\rho\mathsf{LL}, \rho\mathsf{RL}, \rho\mathsf{RR}, \rho\mathsf{LRL}, \rho\mathsf{LRR}\}$ and $\mathbb{V}(s_{23322}) = \{\rho, \rho\mathsf{L}, \rho\mathsf{R}, \rho\mathsf{LL}, \rho\mathsf{LR}, \rho\mathsf{RL}, \rho\mathsf{RR}, \rho\mathsf{LRL}, \rho\mathsf{LRR}\}$. The list of cherry nodes of s_{23322} is $c(s_{23322}) := [\rho\mathsf{LR}, \rho\mathsf{R}]$ and $\boldsymbol{x}_{c(s_{23322})} = \{\boldsymbol{x}_{\rho\mathsf{LR}}, \boldsymbol{x}_{\rho\mathsf{R}}\}$ is the set of boxes associated with them.

Having seen a particular RP s_{23322} let us study the space of all RPs. Let $\mathbb{S}_k$ be the set of all RPs of $\boldsymbol{x}_\rho$ made of k splits. Note that $|\mathbb{L}(s)| = k+1$ if $s \in \mathbb{S}_k$. The number of distinct binary trees with k splits is equal to the Catalan number

$$C_k = \frac{1}{k+1}\binom{2k}{k} = \frac{(2k)!}{(k+1)!(k!)} \ . \tag{1}$$

For $i, j \in \mathbb{Z}_+$, where $\mathbb{Z}_+ := \{0, 1, 2, \ldots\}$ and $i \le j$, let $\mathbb{S}_{i:j}$ be the set of RPs with k splits where $k \in \{i, i+1, \ldots, j\}$. The space of all RPs is then $\mathbb{S}_{0:\infty} := \lim_{j\to\infty} \mathbb{S}_{0:j}$. Figure 4 displays the transition diagram over $\mathbb{S}_{0:3}$ where the gray arrows represent the transition from one RP state to another through a split or reunion. Each sequence of splits and merges of an RP s with root node ρ returns a partition of its root box $\boldsymbol{x}_\rho$ given by the set of its leaf boxes $\boldsymbol{x}_{\mathbb{L}(s)}$.

There may be more than one path from the root node to a particular RP in $\mathbb{S}_k$, i.e. more than one distinct sequence of k splits may result in the same RP in $\mathbb{S}_k$. In Figure 4, for example, there are two paths to s_{2222}.

Randomised algorithms of interest here are Markov chains on $\mathbb{S}_{0:\infty}$. In Section III we shall describe two such algorithms that use a randomised priority queue to rigorously approximate a real-valued function that has an interval inclusion function.

III. $\mathbb{IR}$-Mapped regular pavings

Let $s \in \mathbb{S}_{0:\infty}$ be an RP with root node ρ and root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$. Let $\mathbb{V}(s)$ and $\mathbb{L}(s)$ denote the set of all nodes and leaf nodes of s, respectively. Let $\boldsymbol{f} : \mathbb{V}(s) \to \mathbb{IR}$ map each node of s to an element in $\mathbb{IR}$ as follows:

$$\{\rho\mathsf{v} \mapsto \boldsymbol{f}_{\rho\mathsf{v}} : \rho\mathsf{v} \in \mathbb{V}(s), \boldsymbol{f}_{\rho\mathsf{v}} \in \mathbb{IR}\} \ .$$

Such a map $\boldsymbol{f}$ is called an $\mathbb{IR}$-*mapped regular paving* ($\mathbb{IR}$-MRP). Thus, an $\mathbb{IR}$-MRP $\boldsymbol{f}$ is obtained by augmenting each node $\rho\mathsf{v}$ of the RP tree s with an additional data member $\boldsymbol{f}_{\rho\mathsf{v}} \in \mathbb{IR}$.

Let $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ and $g : \boldsymbol{x}_\rho \to \mathbb{R}$ be a continuous function that is known point-wise for any $x \in \boldsymbol{x}_\rho$, i.e., we have a procedure that returns $g(x)$ for a given $x \in \boldsymbol{x}_\rho$. Our objective is to obtain an $\mathbb{IR}$-MRP $\boldsymbol{f}$ to enclose the range of g. In order to be able to rigorously bound the range we need the notion of inclusion functions from interval analysis.

Let $\boldsymbol{x}_\rho$ and $\boldsymbol{y}$ be complete lattices. Typically, $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ and $\boldsymbol{y} \in \mathbb{IR}^c$. Let $\mathcal{G}$ be the set of all functions from $\mathbb{I}\boldsymbol{x}_\rho$ to $\mathbb{I}\boldsymbol{y}$. Then, $(\mathcal{G}, \le)$ is a complete lattice under the order relation $f \le g \iff \forall x \in \boldsymbol{x}_\rho, f(x) \le g(x)$ and an interval in $\mathcal{G}$ is $\boldsymbol{f} = [\underline{f}, \overline{f}]$ such that $\underline{f} \le \overline{f}$. Now suppose we are given a function $g : \boldsymbol{x}_\rho \to \boldsymbol{y}$. Then an *inclusion function* of g is a function $\boldsymbol{g} : \mathbb{I}\boldsymbol{x} \to \mathbb{I}\boldsymbol{y}$ that satisfies:

$$\forall \boldsymbol{x} \in \mathbb{I}\boldsymbol{x}_\rho, \qquad g(\boldsymbol{x}) := \{g(x) : x \in \boldsymbol{x}\} \subseteq \boldsymbol{g}(\boldsymbol{x}) \ , \tag{2}$$

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{I}\boldsymbol{x}_\rho, \qquad \boldsymbol{x} \subseteq \boldsymbol{x}' \implies g(\boldsymbol{x}) \subseteq \boldsymbol{g}(\boldsymbol{x}') \ . \tag{3}$$

We say that $g : \boldsymbol{x}_\rho \to \mathbb{R}$ has a *well-defined natural interval extension* $\boldsymbol{g} : \mathbb{I}\boldsymbol{x}_\rho \to \mathbb{IR}$ if $\boldsymbol{g}(\boldsymbol{x})$ that is obtained by replacing the sub-expressions in g with their interval counterparts satisfies $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{IR}$ for each $\boldsymbol{x} \in \mathbb{I}\boldsymbol{x}_\rho$. Note that by the fundamental theorem of interval analysis [8] a well-defined interval extension of g is indeed an inclusion function that satisfies (2) and (3).

Theorem III.1 (Enclosing range of functions)**.** *Let $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ and $g : \boldsymbol{x}_\rho \to \mathbb{R}$ be a function with a well-defined inclusion function $\boldsymbol{g} : \mathbb{I}\boldsymbol{x}_\rho \to \mathbb{IR}$. Then, for a given $\epsilon > 0$: $\texttt{UniformApprox}(\rho, g, \boldsymbol{g}, \epsilon)$ of Algorithm 1 will produce an $\mathbb{IR}$-MRP $\boldsymbol{f}$ that uniformly encloses g using intervals of diameter no larger than ϵ, i.e.,*

$$\forall x \in \boldsymbol{x}_\rho, \boldsymbol{f}(x) := \sum_{\rho\mathsf{v} \in \mathbb{L}(s)} \mathbb{1}_{\boldsymbol{x}_{\rho\mathsf{v}}}(x) \boldsymbol{f}_{\rho\mathsf{v}} \supset g(x) \ ,$$

and

$$\text{wid}\,(\boldsymbol{f}(x)) \leq \epsilon \ .$$

Proof: The proof follows from the fundamental theorem of interval analysis by an induction argument on the sub-expressions of the arithmetical expression defining g [8]. ■

Algorithm 1: $\texttt{UniformApprox}(\rho, g, \boldsymbol{g}, \epsilon)$

input : ρ, the root node of $\mathbb{IR}$-MRP f with root box $\boldsymbol{x}_\rho$, function g to be approximated, its inclusion function $\boldsymbol{g}$, and tolerance ϵ.
output : $\mathbb{IR}$-MRP $\boldsymbol{f}$ such that for any $x \in \boldsymbol{x}_\rho$, $g(x) \in \boldsymbol{f}(x)$, $\text{wid}\,(\boldsymbol{f}(x)) \leq \epsilon$.

```
if !IsLeaf(ρ) then
    UniformApprox(ρL, g, g, ε)
    UniformApprox(ρR, g, g, ε)
end
else
    f_ρ ← g(x_ρ)
    if wid (g(x_ρ)) > ε then
        Split ρ: ▽(ρ) = {ρL, ρR}
        UniformApprox(ρL, g, g, ε)
        UniformApprox(ρR, g, g, ε)
    end
end
```

Using Algorithm 1 and Theorem III.1 we can get an $\mathbb{IR}$-MRP $\boldsymbol{f}$ approximation of g such that for any $x \in \boldsymbol{x}_\rho$, $g(x) \in \boldsymbol{f}(x)$ and $\text{wid}\,(\boldsymbol{f}(x)) \leq \epsilon$ for some specified tolerance ϵ. This means that we can, in theory, get an $\mathbb{IR}$-MRP $\boldsymbol{f}$ that encloses g as tightly as we want.

A practical disadvantage of Algorithm 1 is that it is ultimately limited by available machine memory to store the MRP tree used to approximate $g : \boldsymbol{x}_\rho \to \mathbb{R}$, $\boldsymbol{x}_\rho \in \mathbb{IR}^d$, especially for large d. In some cases it may be impossible to run Algorithm 1 for a specified range tolerance ϵ because the required tree would be too large (have too many leaves). One could experimentally increase the tolerance to produce an MRP approximation that can be held in machine memory. A more sensible strategy is to produce an MRP approximation that tries to meet the range tolerance condition, or some other suitable criterion for the tightness of the enclosure, in an optimal manner for a given maximum number of leaves.

One way to create such a strategy is to specify an appropriate priority function $\psi : \mathbb{L}(s) \to \mathbb{R}$ on the set of leaves of the current MRP with RP s and split a leaf node that is uniformly chosen at random from $\text{argmax}_{\rho\mathsf{v} \in \mathbb{L}(s)}\, \psi(\rho\mathsf{v})$, the set of leaf nodes of s which are equally 'large' when measured using ψ. In Examples 1 and 2 we use

$$\psi(\rho\mathsf{v}) = \text{vol}\,(\rho\mathsf{v}) \text{wid}\,(\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{v}})) \ . \tag{4}$$

This priority function measures the *volume of interval enclosure* of the leaf node $\rho\mathsf{v}$ (the product of the volume of the box $\boldsymbol{x}_{\rho\mathsf{v}}$ and the width of its image interval under $\boldsymbol{g}$). This ψ prioritises the splitting of leaf nodes with the largest volume of interval enclosure.

Recall that once we have chosen a leaf box to split, its bisection proceeds perpendicularly at the mid-point along the first widest side of the box. This ensures that our partitions remain in the space of regular pavings of the root box that are closed under refinements, i.e., we can overlay one RP partition over another and obtain a refined RP partition that contains the smallest leaf boxes from both partitions. This closure under refinements or overlays gives us various useful properties, including efficient algorithms to perform arithmetic operations directly over $\mathbb{IR}$-MRPs (this arithmetic is introduced in [9]). The algorithms here thus focus on finding the "best" leaf node to split and cherry node to merge using randomized priority queues over the current set of leaf and cherry nodes, respectively. We purposely avoid making other choices regarding the coordinates and points for bisection in anticipation of further arithmetical operations one may want to perform with two or more $\mathbb{IR}$-MRPs that are constructed with the randomized priority queues introduced here.

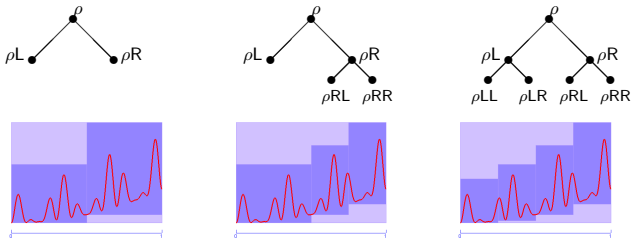
Example 1 (prioritised splitting)**:** Let us illustrates this with a simple example in one dimension when the function to be enclosed is $g : [0, 1] \to \mathbb{R}$ where $g(x) = x^2 + (x + 1)\sin(10\pi x)^2 \cos(3\pi x)^2$ with inclusion function $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}^2 + (\boldsymbol{x} + 1)\sin(10\pi \boldsymbol{x})^2 \cos(3\pi \boldsymbol{x})^2$. We use the priority function in (4).

We start with a single (root) node and root box [0.0, 1.0]. Since this is the only node it is trivially the node with the largest $\psi(\rho\mathsf{v})$, and so will be split. The leaf nodes are now $\rho\mathsf{R}$ and $\rho\mathsf{L}$ with range enclosures $\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{R}})$ and $\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{L}})$ respectively.

Figure 5 shows the next few steps:

1) Figure5(a) shows interval enclosures of the function on $\boldsymbol{x}_{\rho\mathsf{L}} = [0.0, 0.5)$ and $\boldsymbol{x}_{\rho\mathsf{R}} = [0.5, 1.0]$. The volume of the interval enclosure of $\boldsymbol{x}_{\rho\mathsf{R}}$ is the largest, i.e. $\psi(\rho\mathsf{R}) > \psi(\rho\mathsf{L})$, and so $\rho\mathsf{R}$ is the next node to be split.
2) Figure 5(b) shows interval enclosures of the function on $\boldsymbol{x}_{\rho\mathsf{L}} = [0.0, 0.5)$, $\boldsymbol{x}_{\rho\mathsf{RL}} = [0.5, 0.75)$ and $\boldsymbol{x}_{\rho\mathsf{RR}} = [0.75, 1.0]$. The volume of the interval enclosure of $\boldsymbol{x}_{\rho\mathsf{L}}$ is the largest ($\text{argmax}_{\rho\mathsf{v} \in \mathbb{L}(s)}\, \psi(\rho\mathsf{v}) = \{\rho\mathsf{L}\}$) and $\rho\mathsf{L}$ is the next node to be split.
3) Figure 5(c) shows interval enclosures of the function on $\boldsymbol{x}_{\rho\mathsf{LL}} = [0.0, 0.25)$, $\boldsymbol{x}_{\rho\mathsf{LR}} = [0.25, 0.5)$, $\boldsymbol{x}_{\rho\mathsf{RL}} = [0.5, 0.75)$ and $\boldsymbol{x}_{\rho\mathsf{RR}} = [0.75, 1.0]$. $\text{argmax}_{\rho\mathsf{v} \in \mathbb{L}(s)}\, \psi(\rho\mathsf{v}) = \{\rho\mathsf{RR}\}$ and $\rho\mathsf{RR}$ would be the next node to be split if we carried the example further.

In Example 1 we only have a single 'largest' node at each step, but it is possible that there may be more than one such largest node. Once the priority function is motivated we can implement such a sequential bisection procedure using a *randomised priority queue* over the current set of leaf nodes of



(a) Two leaves (b) Three leaves (c) Four leaves

Fig. 5. Priority split.

the MRP. The randomised priority queue will choose the node to be split uniformly at random from $\operatorname{argmax}_{\rho\mathsf{v}\in\mathbb{L}(s)}\psi(\rho\mathsf{v})$.

Another natural priority function is the following:

$$\psi(\rho\mathsf{v}) = \operatorname{wid}\left(\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{v}})\right) \ . \tag{5}$$

This priority function measures the *width of interval enclosure* of the leaf node $\rho\mathsf{v}$ and prioritises the splitting of leaf nodes with the largest enclosure width.

We also need a rule that tells us when to stop splitting (a 'stopping rule'). A simple rule is to stop splitting when $|\mathbb{L}(s)|$, the total number of leaf nodes in the MRP, reaches some maximum limit $\bar{\ell}$ that is usually imposed by internal machine memory.

Finally, we are equipped with the necessary ingredients to produce a memory-efficient $\mathbb{IR}$-MRP $\boldsymbol{f}$ with RP s and root box $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ that encloses the function $g : \boldsymbol{x}_\rho \to \mathbb{R}$ that has an inclusion function $\boldsymbol{g} : \mathbb{I}\boldsymbol{x}_\rho \to \mathbb{IR}$ using at most $\bar{\ell} = |\mathbb{L}(s)|$ many leaves, such that $\operatorname{vol}(\boldsymbol{x}_{\rho\mathsf{v}}) \cdot \operatorname{wid}(\boldsymbol{f}_{\rho\mathsf{v}})$ for each leaf node $\rho\mathsf{v} \in \mathbb{L}(s)$ is made as small as possible under a heuristic sequential splitting strategy based on a randomised priority queue of leaves.

Algorithm 2: $\texttt{RPQEnclose}^{\nabla}(\rho, \boldsymbol{g}, \psi, \bar{\ell})$

input : ρ, the root node of $\mathbb{IR}$-MRP $\boldsymbol{f}$ with RP s, root box $\boldsymbol{x}_\rho$ and $\boldsymbol{f}_\rho = \boldsymbol{g}(\boldsymbol{x}_\rho)$, $\psi : \mathbb{L}(s) \to \mathbb{R}$ given by say (4) or (5), $\bar{\ell}$ the maximum number of leaves.
output : $\boldsymbol{f}$ with modified RP s such that $|\mathbb{L}(s)| = \bar{\ell}$.

if $|\mathbb{L}(s)| < \bar{\ell}$ **then**
  $\rho\mathsf{v} \leftarrow \texttt{random\_sample}\left(\operatorname{argmax}_{\rho\mathsf{v}\in\mathbb{L}(s)}\psi(\rho\mathsf{v})\right)$
  Split $\rho\mathsf{v}$: $\nabla(\rho\mathsf{v}) = \{\rho\mathsf{vL}, \rho\mathsf{vR}\}$ $\boldsymbol{f}_{\rho\mathsf{vL}} \leftarrow \boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{vL}})$
  $\boldsymbol{f}_{\rho\mathsf{vR}} \leftarrow \boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{vL}})$
  $\texttt{RPQEnclose}^{\nabla}(\rho, \psi, \bar{\ell})$
end

The result of this sequential splitting that is informed by the randomised priority queue with priority function ψ is an $\mathbb{IR}$-MRP $\boldsymbol{f}$ with RP $s \in \mathbb{S}_{0:\bar{\ell}-1}$. Note that we can supply any reasonable priority function ψ to Algorithm 2 in order to find an enclosure satisfying the desired priority criterion. For instance, $\psi(\rho\mathsf{v}) = \operatorname{wid}\left(\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{v}})\right)$ will prioritise the splitting of leaf nodes with the widest range enclosure.

Recall that our aim in developing this procedure was to find a way to make an $\mathbb{IR}$-MRP f that encloses a target simple function g "in an optimal manner for a given maximum number of leaves". The priority function ψ reflects what is meant by 'optimal': $\psi(\rho\mathsf{v}) = \operatorname{wid}\left(\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{v}})\right)$ aims for the smallest maximum range enclosures on any sub-box of the partition; $\psi(\rho\mathsf{v}) = \operatorname{vol}(\rho\mathsf{v})\operatorname{wid}\left(\boldsymbol{g}(\boldsymbol{x}_{\rho\mathsf{v}})\right)$ aims to minimise the total volume of the interval enclosures of the sub-boxes $\sum_{\rho\mathsf{v}\in\mathbb{L}(s)}\left(\operatorname{vol}(\boldsymbol{x}_{\rho\mathsf{v}})\operatorname{wid}(\boldsymbol{f}_{\rho\mathsf{v}})\right)$.

We now describe a refinement to our heuristic splitting strategy that can be guaranteed to give a range enclosure that is at least as good as that achieved by the use of $\texttt{RPQEnclose}^{\nabla}$ alone, and in many cases better.

This refinement is based on the observation that the more we split (the larger $\bar{\ell}$ is) the smaller the boxes associated with the leaves and (by (3)) the tighter the range enclosures of the inclusion function $\boldsymbol{g}$ over these boxes. This suggests that we could benefit by splitting to a larger number of leaves than we really want, using the information from the range enclosures at the leaves to adjust (tighten) the range enclosures of the internal nodes, and then perform a 'prioritised prune' to reduce the tree to some ultimate target number of leaves. This is the idea behind Algorithms 3 and 4.

Let $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$ be two intervals, then the smallest interval containing the union of $\boldsymbol{x}$ and $\boldsymbol{y}$ is called their *interval hull* and given by $\boldsymbol{x} \sqcup \boldsymbol{y} := [\min(\underline{x}, \underline{y}), \max(\overline{x}, \overline{y})]$.

Algorithm 3: $\texttt{HullPropagate}(\rho)$

input : ρ, the root node of $\mathbb{IR}$-MRP $\boldsymbol{f}$ with RP s.
output : Modify input MRP $\boldsymbol{f}$.

if $\texttt{!IsLeaf}(\rho)$ **then**
  $\texttt{HullPropagate}(\rho\mathsf{L})$
  $\texttt{HullPropagate}(\rho\mathsf{R})$
  $\boldsymbol{f}_\rho \leftarrow \boldsymbol{f}_{\rho\mathsf{L}} \sqcup \boldsymbol{f}_{\rho\mathsf{R}}$
end

$\texttt{HullPropagate}(\rho)$ works from the leaves of $\boldsymbol{f}$ upwards to replace the intervals mapped onto the internal nodes with the interval hull of the intervals mapped onto their children. Thus, $\operatorname{wid}(\boldsymbol{f}_{\rho\mathsf{v}}) \leq \operatorname{wid}\left(\boldsymbol{g}(x_{\rho\mathsf{v}})\right)$ for each internal node $\rho\mathsf{v} \in \mathbb{V}(s) \setminus \mathbb{L}(s)$ after an $\boldsymbol{f}$ produced by $\texttt{RPQEnclose}^{\nabla}$ has been modified by a call to $\texttt{HullPropagate}(\rho)$.

The 'prioritised prune' to fewer leaves is achieved by Algorithm 4. When we prune, we select a cherry node from $\mathbb{C}(s)$ the set of cherry nodes $\mathbb{IR}$-MRP $\boldsymbol{f}$ and prune off both children so that the selected node becomes a leaf node. Algorithm 4 uses some appropriate priority function $\psi : \mathbb{C}(s) \to \mathbb{R}$ and chooses the next cherry node for pruning uniformly at random from $\operatorname{argmin}_{\rho\mathsf{v}\in\mathbb{C}(s)}\psi(\rho\mathsf{v})$, i.e. from the cherries that are the *smallest* measured using ψ.

In Example 2 we use

$$\psi(\rho\mathsf{v}) = \operatorname{vol}(\rho\mathsf{v})\operatorname{wid}(\boldsymbol{f}_{\rho\mathsf{v}}) \ , \tag{6}$$

the priority function that measures the volume of the interval enclosure of a node $\rho\mathsf{v}$. This ψ prioritises the pruning of cherry nodes with the smallest volume of interval enclosure. The priority function that measures the width of the interval

enclosure can also be used

$$\psi(\rho \mathsf{v}) = \text{wid}\left(\boldsymbol{f}_{\rho \mathsf{v}}\right) \ . \tag{7}$$

A simple stopping-rule is to keep pruning cherries back into leaves until $|\mathbb{L}(s)|$, the total number of leaf nodes in the MRP, has decreased to some maximum limit $\bar{\ell}'$.

Algorithm 4: $\texttt{RPQEnclose}^{\Delta}(\rho, \psi, \bar{\ell}')$

input : ρ, the root node of $\mathbb{IR}$-MRP $\boldsymbol{f}$ with RP s, box $\boldsymbol{x}_\rho$,
$\psi : \mathbb{L}(s) \to \mathbb{R}$ given by say (6) or (7),
$\bar{\ell}'$ the maximum number of leaves.

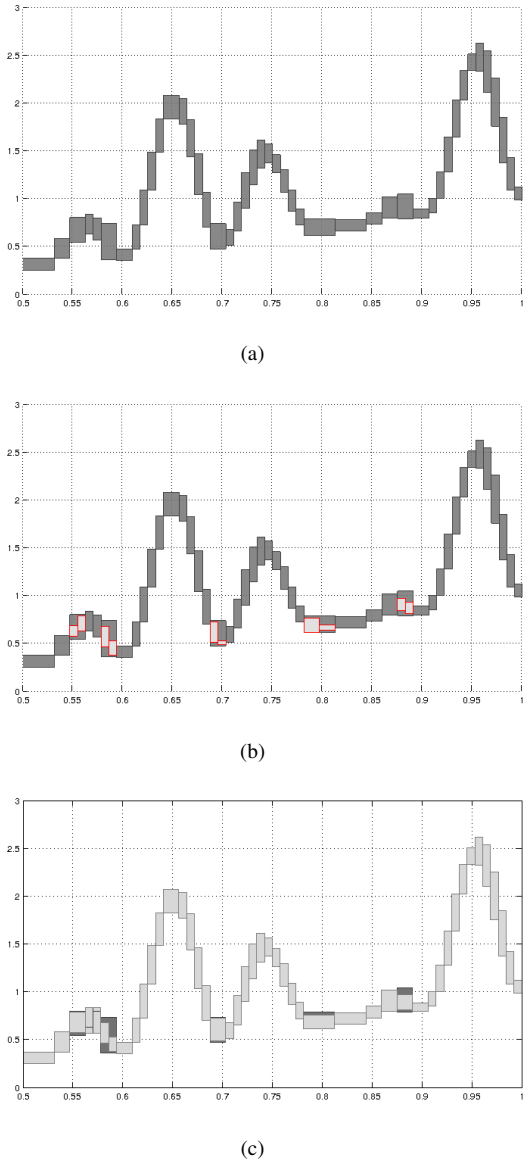
output : modified $\boldsymbol{f}$ with RP s such that $|\mathbb{L}(s)| = \bar{\ell}'$ or $\mathbb{C}(s) = \emptyset$.

if $|\mathbb{L}(s)| \geq \bar{\ell}' \ \ \& \ \ \mathbb{C}(s) \neq \emptyset$ **then**
  $\rho \mathsf{v} \leftarrow \texttt{random\_sample}\left(\text{argmin}_{\rho \mathsf{v} \in \mathbb{C}(s)}\, \psi(\rho \mathsf{v})\right)$
  $\texttt{Prune}(\rho \mathsf{L})$
  $\texttt{Prune}(\rho \mathsf{R})$
  $\texttt{RPQEnclose}^{\Delta}(\rho,\ \psi,\ \bar{\ell}')$
end

Notice that `HullPropagate` and $\texttt{RPQEnclose}^{\Delta}$ can be nicely separated because of the tree structure. This allows a much clearer, simpler and more flexible implementation.

We can use Algorithm 2, Algorithm 3 and Algorithm 4 in succession to obtain an interval-based approximation for the function $g : \boldsymbol{x}_\rho \to \mathbb{R}$ that has an inclusion function $\boldsymbol{g}$. We start with $\mathbb{IR}$-MRP $\boldsymbol{f}$ consisting of just the root node ρ with an appropriate root box $\boldsymbol{x}_\rho$ where $\boldsymbol{f}_\rho = \boldsymbol{g}(\boldsymbol{x}_\rho)$. First we use $\texttt{RPQEnclose}^{\nabla}$ of Algorithm 2, specifying some priority function ψ and some maximum number of leaves $\bar{\ell}$ that is larger than we want in our final approximation. The result is $\boldsymbol{f}$ with RP $s \in \mathbb{S}_{0:\bar{\ell}-1}$. Then we perform $\texttt{HullPropagate}(\rho)$ of Algorithm 3 on the root node ρ of $\boldsymbol{f}$ to tighten the range enclosures on the internal nodes of $\boldsymbol{f}$. Finally we reduce the number of leaves to $\bar{\ell}' < \bar{\ell}$ using $\texttt{RPQEnclose}^{\Delta}$ of Algorithm 4, again specifying some priority function for the pruning.

Example 2 (split, propagate hull & prune)**:** In this example we will explain the process leading to the better enclosure in Figure 1. Let us reconsider the function $g : [0,1] \to \mathbb{R}$ where $g(x) = x^2 + (x+1)\sin(10\pi x)^2\cos(3\pi x)^2$ with the well-defined inclusion function $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}^2 + (\boldsymbol{x}+1)\sin(10\pi \boldsymbol{x})^2\cos(3\pi \boldsymbol{x})^2$ of Example 1. Suppose we first try to approximate g on domain $[0.5,\ 1.0]$ with $\mathbb{IR}$-MRP $\boldsymbol{f}$ made by $\texttt{RPQEnclose}^{\nabla}(\rho, \boldsymbol{g}, \psi, \bar{\ell})$ that encloses the range of g with intervals over an RP s with $|\mathbb{L}(s)| = \bar{\ell} = 45$ leaves and $\psi(\rho \mathsf{v}) = \text{vol}\,(\rho \mathsf{v})\text{wid}\,(\boldsymbol{g}(\boldsymbol{x}_{\rho \mathsf{v}}))$. Figure 6(a) displays such an $\boldsymbol{f}$. We can measure the tightness of the range enclosure of this $\mathbb{IR}$-MRP $\boldsymbol{f}$ by finding the volume of the enclosure over the leaf nodes $\Xi(\boldsymbol{f}) := \sum_{\rho \mathsf{v} \in \mathbb{L}(s)} \text{wid}\,(\boldsymbol{f}_{\rho \mathsf{v}})\text{vol}\,(\boldsymbol{x}_{\rho \mathsf{v}}) = 0.1215$. Figure 6(b) displays the $\mathbb{IR}$-MRP $\boldsymbol{f}'$ made by $\texttt{RPQEnclose}^{\nabla}(\rho, \boldsymbol{g}, \psi, \bar{\ell})$ with $\bar{\ell} = 50$ leaves. With these 5 new splits we have 10 new leaves with tighter range enclosures (light gray boxes with red edges in Figure 6(b)). With these additional leaves, the volume of the enclosure over the leaf nodes is smaller, $\Xi(\boldsymbol{f}') = 0.1102$. Thus, with this partition that has 5 more leaves than the one in Figure 6(a), we have reduced the total volume of the enclosure by $\Xi(\boldsymbol{f}) - \Xi(\boldsymbol{f}') = 0.1215 - 0.1102 = 0.0113$. However, by calling $\texttt{HullPropagate}(\rho')$ upon ρ', the root node of $\boldsymbol{f}'$ (50 leaves), we can propagate the hull of the range enclosures from the leaf nodes of $\boldsymbol{f}'$ up through its internal nodes. Finally, we can call $\texttt{RPQEnclose}^{\Delta}(\rho', \psi, \bar{\ell}')$ with $\bar{\ell}' = 45$ and $\psi(\rho \mathsf{v}) = \text{vol}\,(\rho \mathsf{v})\text{wid}\,(\boldsymbol{f}_{\rho \mathsf{v}})$. We obtain a modified $\mathbb{IR}$-MRP $\boldsymbol{f}'$ that has only 45 leaves as shown in Figure 6(c) with $\Xi(\boldsymbol{f}') = 0.1159$. Now $\boldsymbol{f}'$ has the same number of leaves as $\boldsymbol{f}$ but has a different partition with a tighter range enclosure: $\Xi(\boldsymbol{f}) - \Xi(\boldsymbol{f}') = 0.1215 - 0.1159 = 0.0056$.



(a) (b) (c)

Fig. 6. $\mathbb{IR}$-MRPs to enclose the function g of Example 2 using $\texttt{RPQEnclose}^{\nabla}(\rho, \boldsymbol{g}, \psi, 45)$ in (a), $\texttt{RPQEnclose}^{\nabla}(\rho', \boldsymbol{g}, \psi, 50)$ in (b) and $\texttt{HullPropagate}(\rho')$ followed by $\texttt{RPQEnclose}^{\Delta}(\rho', \psi, 45)$ in (c).

Our method can be applied to functions in higher dimensions as well. Figure 7 shows an $\mathbb{IR}$-MRP enclosure of the Rosenbrock function $f(x) : \mathbb{R}^2 \to \mathbb{R}$, $f(x) = (1-x_1)^2 + 100(x_2 - x_1^2)^2$ on $[-1,1]^2$.

Our method naturally suffers from the curse of dimen-

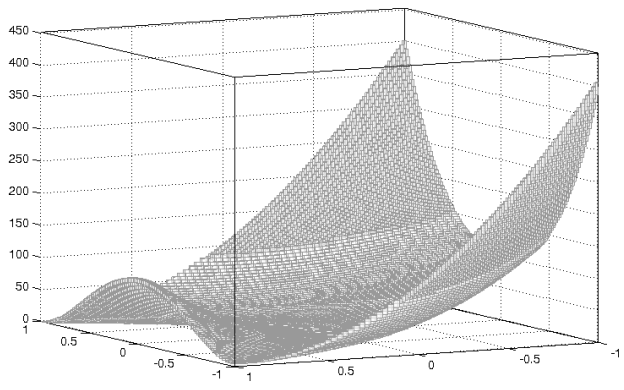
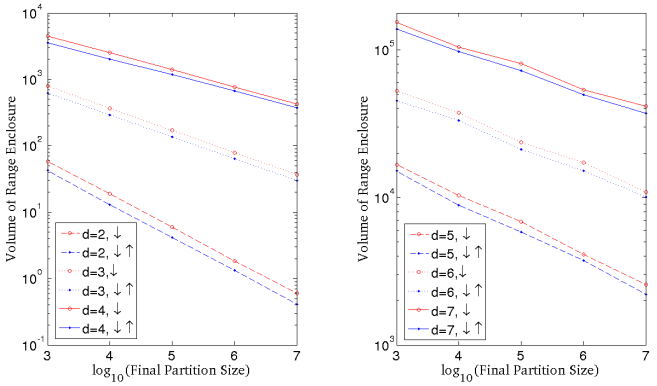


Fig. 7. $\mathbb{IR}$-MRP enclosure of the Rosenbrock function with domain $x_\rho = [-1,1]^2$ and maximum number of leaves $\hat{\ell}' = 7000$.



(a) $d = 2, 3, 4$ (b) $d = 5, 6, 7$

Fig. 8. Volumes of range enclosures for the multi-dimensional Rosenbrock function over $[-1,1]^d$ based on just splitting (↓) to final parition size versus splitting further, propagating hull and pruning back to final parition size (↓↑). See text for description.

sionality. However, for a given final partition size $\bar{\ell}'$ in any dimension we can improve on the range enclosures obtained from an RP s using $\texttt{RPQEnclose}^{\nabla}$ alone by continuing to split using $\texttt{RPQEnclose}^{\nabla}$ to more leaves, say, $\bar{\ell} = 2 \times \bar{\ell}'$, then calling $\texttt{HullPropagate}(\rho)$ and finally pruning back to an RP s' with $\bar{\ell}'$ many leaves using $\texttt{RPQEnclose}^{\Delta}$. We found an improvement (summarized in Figure 8) in the volume of the range enclosures using the priority function ψ in (4) for the multivariate Rosenbrock function (see for e.g. [9, Eqn. (30)]) over $x_\rho = [-1,1]^d$ and $d \in \{2,3,4,5,6,7\}$. We allowed the $\bar{\ell}'$ to range in $\{10^3, 10^4, 10^5, 10^6, 10^7\}$ and set $\bar{\ell} = 2 \times \bar{\ell}'$. We also found tighter maximal diameters for the same setting when the priority function in (5) was used (results not shown). The curse of dimensionality is evident in Figure 8 since the slopes of the lines that relate partition size to enclosure volumes decrease in steepness as the dimension increases.

IV. Discussion

We have developed novel randomised algorithms that provide memory-efficient enclosures of a real-valued function with a well-defined inclusion function by forming $\mathbb{IR}$-mapped regular pavings using randomised priority queues of their leaf nodes. We show how the range enclosure for a number of leaf nodes can be tightened by making an initial priority queue split to some larger number of leaves and then using the tree structure to propagate the hull of the range enclosures from the leaves up to the root and then using a similar randomised priority queue to prune the tree to the required final size.

The structures and algorithms described in this paper are implemented in *MRS: a C++ class library for statistical set processing* and publicly available under the terms of the GNU General Public License from http://www.math.canterbury.ac.nz/~r.sainudiin/codes/mrs/.

Although we have focused here on enclosing functions from $\boldsymbol{x}_\rho \in \mathbb{IR}^d$ to $\mathbb{R}$ with well-defined inclusion functions, we can easily allow for differentiation arithmetic or Taylor arithmetic instead of range arithmetic by making further assumptions on the class of functions being enclosed. These would lead to tighter enclosures that can be further tightened by our splitting and pruning algorithms.

Obtaining $\mathbb{IR}$-MRPs is only the first step. Since the underlying trees are closed under refinements we can efficiently perform arithmetic operations over $\mathbb{IR}$-MRPs (see [9]). For example, we can add, subtract, multiply or divide two $\mathbb{IR}$-MRPs or apply elementary transformations, such as $\exp, \sin, \cos, \ldots$, to an $\mathbb{IR}$-MRP and obtain the result as another $\mathbb{IR}$-MRP. Such operations with interval mapped regular pavings and their compositions can be useful.

Acknowledgements

This research was partly supported by RS's external consulting revenues from the New Zealand Ministry of Tourism, WT's Swedish Research Council Grant 2008-7510 that enabled RS's visits to Uppsala in 2006 and 2009, Erskine grant from University of Canterbury that enabled WT's visit to Christchurch in 2011 and University of Canterbury MSc Scholarship to JH.

References

[1] A. Kaufmann and M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications*. New York, New York: Van Nostrand Reinhold Company, 1985.

[2] H. Samet, "Multidimensional spatial data structures," in *Handbook of Data Structures and Applications*, D. P. Mehta and S. Sahni, Eds. Boca Raton: Chapman and Hall/CRC, 2004, ch. 16.

[3] ——, *Foundations of multidimensional and metric data structures*. San Francisco: Morgan Kaufman, 2006.

[4] ——, *The design and analysis of spatial data structures*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1990.

[5] M. Kieffer, L. Jaulin, I. Braems, and E. Walter, "Guaranteed set computation with subpavings," in *Scientific Computing, Validated Numerics, Interval Methods, Proceedings of SCAN 2000*, W. Kraemer and J. Gudenberg, Eds. Kluwer Academic Publishers, 2001, pp. 167–178.

[6] R. P. Stanley, *Enumerative Combinatorics, Vol. 2*. Cambridge: Cambridge university press, 1999.

[7] J. Meier, *Groups, graphs and trees: an introduction to the geometry of infinite groups*, ser. London Mathematical Society student texts. Cambridge: Cambridge University Press, 2008.

[8] R. E. Moore, *Interval analysis*. Englewood Cliffs, New Jersey: Prentice-Hall, 1967.

[9] J. Harlow, R. Sainudiin, and W. Tucker, "Mapped regular pavings," *Reliable Computing*, vol. 16, pp. 252–282, 2012.