■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... • Encoding/Representation. The most basic encoding used in a GA for representing the parameter(s) defining fitness is binary or bit-string encoding. It is assumed that the value(s) of a parameter(s) has a certain (finite) range(s), i.e., a certain associated interval(s) in which the value(s) can vary. In the simplest case of a univariate fitness function, one must only encode the allowed range interval of a single independent parameter via a set of bit-strings of a fixed length, defining specific values in the interval. Of course, a fitness function is generally a multivariate function, where the fitness depends on several parameters and multiple range intervals are encoded via bit--strings (with possibly different lengths). In an even more general setting, the fitness function can be a multi-/vector-valued function, consisting of multiple individual objective functions to be considered in the overall fitness function optimization. In our box tree scenario, the parameters of a single box are the center point and the D basis vectors, each defined by D coordinate values.

•••

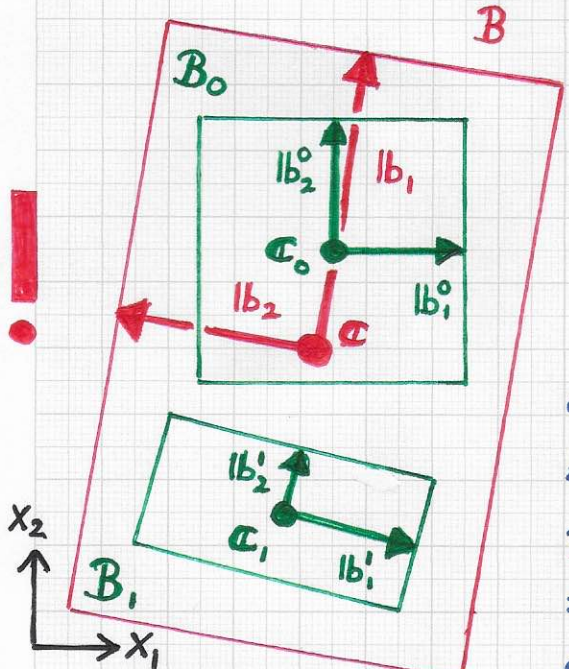
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Further, we can consider a classification performance fitness function that is single-/scalar-valued or multi-/vector-valued: We can measure performance with a single **P**-value (scalar-valued case) or, for example, with the values of five classification outcome types, i.e., **TN**, **TP**, **FN**, **FP** and **MC** (mis-classification). Thus, one can use the relative occurrence values of these five specific outcome types to establish a vector-valued fitness function.

!

• Constraints. Typically, the GA must ensure that certain constraints that apply to the parameters to be optimized are satisfied. Such constraints can be numerical, algebraic, geometrical or topological constraints, for example. The left figure shows the essential geometrical parameters to be optimized for a box tree: box center points $\mathbb{c}, \mathbb{c}_0, \mathbb{c}_1, \ldots$ and box basis vectors $\mathbb{b}_1, \mathbb{b}_2, \mathbb{b}_1^0, \mathbb{b}_2^0, \mathbb{b}_1^1, \mathbb{b}_2^1, \ldots$ for tree boxes $\mathbb{B}, \mathbb{B}_0, \mathbb{B}_1, \ldots$

!



Geometrical data defining box tree.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks: ... Considering only the geometrical parameters, the scalar-valued performance function $\mathbf{P}$ to be maximized by the GA can be written as

!

$$P = P(\mathbb{c}, \mathbb{c}_0, \mathbb{c}_1, \mathbb{c}_{00}, \mathbb{c}_{01}, \mathbb{c}_{10}, \mathbb{c}_{11}, \ldots, \mathbb{b}_1, \mathbb{b}_2, \mathbb{b}_1^0, \mathbb{b}_2^0, \mathbb{b}_1^1, \mathbb{b}_2^1, \mathbb{b}_1^{00}, \mathbb{b}_2^{00}, \ldots).$$

The vector-valued function that considers the relative occurrences of classification result types TN, TP, FN, FP and MC, for example, can be written as

!

$$(TN, TP, FN, FP, MC) = \\ = (TN, TP, FN, FP, MC) \\ (\mathbb{c}, \mathbb{c}_0, \mathbb{c}_1, \mathbb{c}_{00}, \ldots, \mathbb{b}_1, \mathbb{b}_2, \mathbb{b}_1^0, \mathbb{b}_2^0, \mathbb{b}_1^1, \mathbb{b}_2^1, \ldots).$$

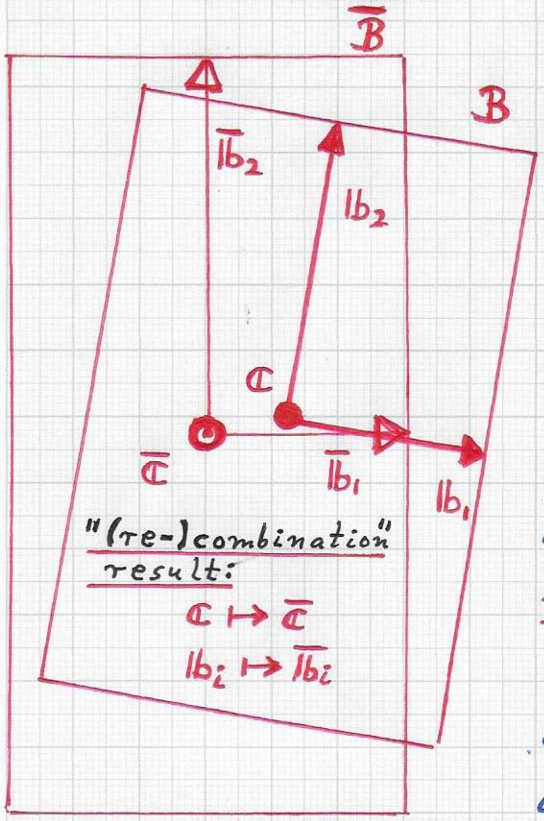
As a consequence, one must define constraints for box centers and basis vectors — if necessary or desirable. One must be careful when defining and imposing such potential constraints. The optimization problem involves and must be solved for a high-dimensional domain. One must NOT assume that one's "intuition" regarding "appropriate constraints" is correct. The GA must be permitted to explore the high-dimensional domain as necessary to ultimately terminate with a near-optimal solution — that might not be "intuitive."

!

•••

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...


$\overline{\mathbb{B}}$
$\mathbb{B}$
$\overline{\mathbb{b}}_2$
$\mathbb{b}_2$
$\mathbb{c}$
$\overline{\mathbb{c}}$
$\overline{\mathbb{b}}_1$
$\mathbb{b}_1$
"(re-)combination" result:
$\mathbb{c} \mapsto \overline{\mathbb{c}}$
$\mathbb{b}_i \mapsto \overline{\mathbb{b}}_i$

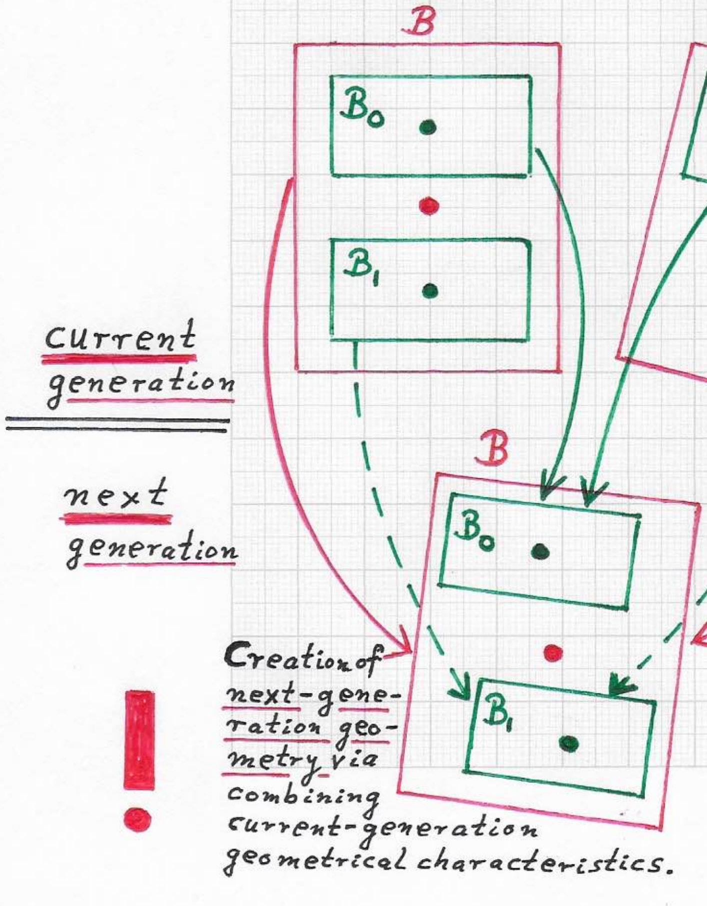
Example of constraints used to restrict the "replacement" of a box.

For example, one can interpret and design geometrical constraints in our box tree scenario as follows: Each box is defined by its center and D mutually orthogonal basis vectors. If a GA were designed and implemented in a way that were to "replace" a current-generation box $\mathbb{B}$ by a next-generation box $\overline{\mathbb{B}}$, then one could consider the incorporation of GA constraints that effectively limit the maximally allowable distance between current and next center points $\mathbb{c}$ and $\overline{\mathbb{c}}$, see left figure.

! (In this context, one must keep in mind that members of a population are entire box trees and not individual boxes. Thus, when combining two members, i.e., two box trees, to combine the geometrical characteristics of different tree nodes does not seem appropriate; in other words, it seems prudent to combine only data of the same node in two trees, e.g., the data defining box tree node $\mathbb{B}_{00}$ from two geometrically different trees.) The figure shows the restricted "replacement" of a box $\mathbb{B}$ by $\overline{\mathbb{B}}$, i.e., $\{\mathbb{c}, \mathbb{b}_1, \mathbb{b}_2\}$ by $\{\overline{\mathbb{c}}, \overline{\mathbb{b}}_1, \overline{\mathbb{b}}_2\}$. ...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Regarding constraints applied to the creation of the D basis vectors of a specific box tree node, basis vectors can change directions and lengths. One could employ constraints that limit changes in direction and length. A necessary constraint that must always be satisfied is orthogonality: It is expected that the created next-generation basis vectors $\mathbb{b}_1, \mathbb{b}_2, \ldots, \mathbb{b}_D$ are mutually orthogonal. Mathematically, one can express and enforce these constraintsvia the use of matrices that represent acceptable translation, rotation and scaling of a box. The figure (left) illustrates in an abstract manner how the geometrical characteristics of node B_0 of two box trees of the current generation, for example, could be combined to define next-generation geometrical characteristics of B_0.



Creation of next-generation geometry via combining current-generation geometrical characteristics.

The shown boxes and the center points capture the box-inherent coordinate systems (centers/origins and axis basis vectors) involved in in this specific (re-)combination. •••