A Two-phase Framework with a Bézier Simplex-based Interpolation Method for Computationally Expensive Multi-objective Optimization

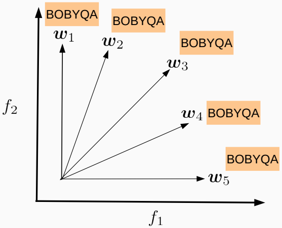
GECCO 2022 at Boston

Ryoji Tanabe (Yokohama National University),
Youhei Akimoto (University of Tsukuba),
Ken Kobayashi (Fujitsu Limited),
Hiroshi Umeki (Yokohama National University),
Shinichi Shirakawa (Yokohama National University),
Naoki Hamada (KLab Inc.)

Four-phased HMO-CMA-ES [Loshchilov 16] performs well for expensive opt.

1st phase runs BOBYQA on K scalar optimization problems

- BOBYQA [Powell 09]: SOTA mathematical derivative-free optimizer
  - It iteratively solves a trust region sub-problem using quadratic models



- 1st phase uses only the $10 \times N$ fevals *in total* (N = Num. variables)
- 2nd: SS-MO-CMA-ES → 3rd: MO-CMA-ES → 4th: CMA-ES
  - HMO-CMA-ES shows an excellent anytime performance ☺

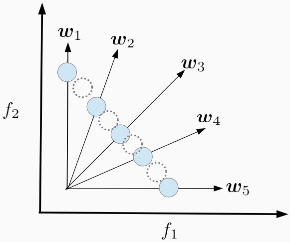
Ilya Loshchilov, Tobias Glasmachers: Anytime Bi-Objective Optimization with a Hybrid Multi-Objective CMA-ES (HMO-CMA-ES). GECCO (Companion) 2016: 1169-1176

M. J. D. Powell. 2009. The BOBYQA algorithm for bound constrained optimization without derivatives. Technical Report DAMTP 2009/NA06. University of Cambridge

One drawback of the 1st phase in HMO-CMA-ES

It can achieve only K sparsely distributed solutions

- Only a limited number of fevals are available
- K needs to be as small as possible ($K = 5$ in ↙)



Motivation

- Can this issue be addressed by a solution interpolation approach?

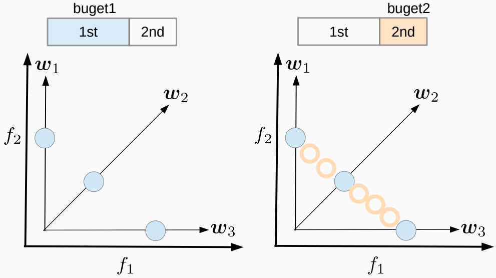
Proposed: A Two-Phase framework with a Bézier simplex-based interpolation method (TPB)

1st phase is similar to that in HMO-CMA-ES

- It runs BOBYQA on K scalar problems, but its details are different
  - K = Num. objectives $+1$ $(= 2 + 1 = 3$ in this study)
  - The normalization procedure, a budget allocation strategy, the order of scalar optimization, control parameters of BOBYQA, etc.

2nd phase interpolates the $K(= 3)$ solutions by the Bézier simplex

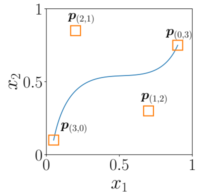
- It is theoretically well-founded and can fit the K solutions



Bézier simplex: A generalized version of the Bézier curve to higher dims

The Bézier curve

$(M = 2,\ N = 2,\ D = 3)$



The Bézier simplex

$(M = 3,\ N = 3,\ D = 3)$

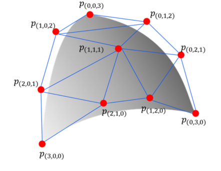


Figure 2: A Bézier simplex for $M = 3$, $D = 3$.

From [Kobayashi 19]

- M: Num. objectives, N: Num. variables, D: Degree of a model
  - D determines the number of control points $\boldsymbol{p}_{\boldsymbol{d}_1}, \ldots \in \mathbb{R}^N$
  - Control points define a Bézier simplex model
- It can describe the Pareto optimal solution set $\boldsymbol{X}^\star$ [Kobayashi 19]
  - When $\boldsymbol{X}^\star$ is homeomorphic to an $(M-1)$-dim. simplex [Hamada 20]
  - The theoretically well-founded nice property for the interpolation ☺

Ken Kobayashi, Naoki Hamada, Akiyoshi Sannai, Akinori Tanaka, Kenichi Bannai, Masashi Sugiyama: Bézier Simplex Fitting: Describing Pareto Fronts of Simplicial Problems with Small Samples in Multi-Objective Optimization. AAAI 2019: 2304-2313

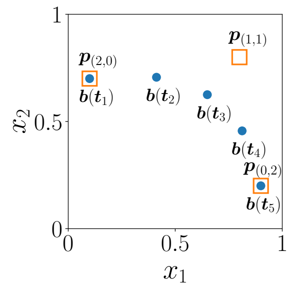
Naoki Hamada, Kenta Hayano, Shunsuke Ichiki, Yutaro Kabata, Hiroshi Teramoto: Topology of Pareto Sets of Strongly Convex Problems. SIAM J. Optim. 30(3): 2659-2686 (2020)

Example: Bézier simplex with $M = 2$ (N. obj), $N = 2$ (N. var), and $D = 2$

A Bézier simplex model $\boldsymbol{b} : \boldsymbol{t} \mapsto \boldsymbol{b}(\boldsymbol{t})$

$$\boldsymbol{b}(\boldsymbol{t}) = t_1^2 \boldsymbol{p}_{(2,0)} + 2 t_1 t_2 \boldsymbol{p}_{(1,1)} + t_2^2 \boldsymbol{p}_{(0,2)}$$

- Input $\boldsymbol{t} \in \mathbb{R}^M$: a parameter vector, and $\sum_{m=1}^{M} t_m = 1, t_m \geq 0$
- Output $\boldsymbol{b}(\boldsymbol{t}) \in \mathbb{R}^N$: a mapping of $\boldsymbol{t}$. It can be a solution $\boldsymbol{x} \in \mathbb{R}^N$
- Model parameters $\boldsymbol{p}_{\boldsymbol{d}_1}, \ldots \in \mathbb{R}^N$: Control points



- $\boldsymbol{p}_{(2,0)} = (0.1, 0.7)$
- $\boldsymbol{p}_{(1,1)} = (0.8, 0.8)$
- $\boldsymbol{p}_{(0,2)} = (0.9, 0.2)$
- $\boldsymbol{t}_1 = (1, 0) \mapsto \boldsymbol{b}(\boldsymbol{t}_1) = (0.1, 0.7)$
- $\boldsymbol{t}_2 = (0.75, 0.25) \mapsto \boldsymbol{b}(\boldsymbol{t}_2) \approx (0.41, 0.71)$
- $\boldsymbol{t}_3 = (0.5, 0.5) \mapsto \boldsymbol{b}(\boldsymbol{t}_3) \approx (0.65, 0.63)$
- $\boldsymbol{t}_4 = (0.25, 0.75) \mapsto \boldsymbol{b}(\boldsymbol{t}_4) \approx (0.81, 0.46)$
- $\boldsymbol{t}_5 = (0, 1) \mapsto \boldsymbol{b}(\boldsymbol{t}_5) = (0.9, 0.2)$

Bézier simplex fitting to approximate a solution set X [Kobayashi 19]

- Let $\boldsymbol{X} = \{\boldsymbol{x}_k \in \mathbb{R}^N\}_{k=1}^K$ be a solution set of size K
- Let $\boldsymbol{T} = \{\boldsymbol{t}_k \in \Delta^{M-1} \subseteq \mathbb{R}^M\}_{k=1}^K$ be a parameter vec. set of size K
  - $\boldsymbol{t}_k$ corresponds to $\boldsymbol{x}_k$
- We want a Bézier simplex model $\boldsymbol{b}$ that approximates $\boldsymbol{X}$
  - How do we set control points, e.g., $\boldsymbol{p}_{(2,0)}$, $\boldsymbol{p}_{(1,1)}$, and $\boldsymbol{p}_{(0,2)}$?
- The Bézier simplex fitting method adjusts the control points ($\boldsymbol{p}_{\boldsymbol{d}}$) by minimizing the ordinary least squares loss function:

$$\text{minimize} \quad \sum_{k=1}^{K} \|\boldsymbol{x}_k - \boldsymbol{b}(\boldsymbol{t}_k)\|^2$$

- The loss function is a convex quadratic function with respect to $\boldsymbol{p}_{\boldsymbol{d}}$
- Its minimizer can be found by solving a normal equation

The PyTorch implementation is available at

`https://github.com/rafcc/pytorch-bsf`

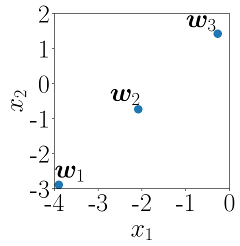
Ken Kobayashi, Naoki Hamada, Akiyoshi Sannai, Akinori Tanaka, Kenichi Bannai, Masashi Sugiyama: Bézier Simplex Fitting: Describing Pareto Fronts of Simplicial Problems with Small Samples in Multi-Objective Optimization. AAAI 2019: 2304-2313

1st phase applies BOBYQA to K scalar optimization problems $\{g_{\boldsymbol{w}_k}\}_{k=1}^{K}$

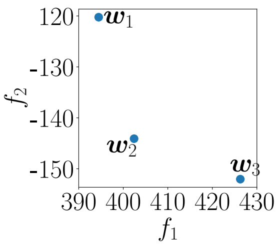
- We set $K = 2 + 1 = 3$, $\boldsymbol{w}_1 = (0, 1)$, $\boldsymbol{w}_2 = (0.5, 0.5)$, $\boldsymbol{w}_1 = (1, 0)$
  - K should be $K \geq M + 1$ to handle the nonlinear PS set
- We use the weighted sum function as in HMO-CMA-ES
  - TPB can use any g, e.g., the weighted Tchebycheff function
- We set a budget ratio $r^{\text{1st}} = 0.9$
  - e.g., $\texttt{budget}^{\text{1st}} = 0.9 \times 40 = 36$ fevals when $\texttt{budget} = 40$ fevals
  - Each run of BOBYQA can use $\texttt{budget}^{\text{1st}}/K = 36/3 = 12$ fevals

Results on f_1 (Sphere/Sphere) with $N = 2$ in `bbob-biobj`

Solution space



Objective space



2nd phase interpolates the K sol. by a Bézier simplex model-based method

2.1 The fitting phase

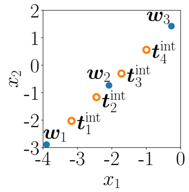
- The 1st phase obtained $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$ using $\boldsymbol{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3\}$
- TPB treats $\boldsymbol{w}_k$ as $\boldsymbol{t}_k^{\text{fit}}$, and $\boldsymbol{T}^{\text{fit}} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3\}$
- TPB fits a Bézier simplex model $\boldsymbol{b}$ to $\boldsymbol{X}$ with $\boldsymbol{T}^{\text{fit}} = \boldsymbol{W}$

2.2 The solution generation phase

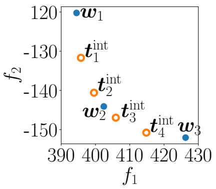
- $\texttt{budget}^{\text{2nd}} = \texttt{budget} - \texttt{budget}^{\text{1st}} = 40 - 36 = 4$ in this example
- 4 solutions are generated by giving $\boldsymbol{t}_1^{\text{int}}, \boldsymbol{t}_2^{\text{int}}, \boldsymbol{t}_3^{\text{int}}, \boldsymbol{t}_4^{\text{int}}$ to $\boldsymbol{b}$
  - We equally generate parameters on Δ^{M-1}, removing $(0, 1)$ and $(1, 0)$

Results on f_1 (D-Sphere) with $N = 2$ in `bbob-biobj`

Solution space



Objective space



- $\boldsymbol{t}_1^{\text{int}} = (0.2, 0.8)$
- $\boldsymbol{t}_2^{\text{int}} = (0.4, 0.6)$
- $\boldsymbol{t}_3^{\text{int}} = (0.6, 0.4)$
- $\boldsymbol{t}_4^{\text{int}} = (0.8, 0.2)$

Advantages and disadvantages of TPB

Advantages ☺

1. TPB can use any SOTA single-objective black-box optimizer
2. TPB can exploit the structure of the PS set
3. TPB is faster than model-based optimizers, e.g., ParEGO [Knowles 06]

Disadvantages ☹

1. The poor anytime performance as in most two-phase approaches
   - They can obtain only a poor-quality solution set when they stop before reaching the maximum budget [Dubois-Lacoste 11]
   - But, this is true for most model-based optimizers (due to the LHS)
2. It performs poorly when the PS topology cannot be a simplex
   - The $15/55$ unimodal `bbob-biobj` problems would be OK
   - The $40/55$ multimodal `bbob-biobj` problems would be NG

Joshua D. Knowles: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. 10(1): 50-66 (2006)

Jérémie Dubois-Lacoste, Manuel López-Ibáñez, Thomas Stützle: Improving the anytime behavior of two-phase local search. Ann. Math. Artif. Intell. 61(2): 125-154 (2011)

Experimental setup

- All experiments were conducted using COCO [Hansen 21]
    - The 55 `bbob-biobj` problems [Brockhoff 22] with $N \in \{2, 3, 5, 10, 20\}$
    - I_{COCO}: The uncrowded HV based on the unbounded external arch.
        - When none of objective points dominates the reference point $\boldsymbol{z}$, I_{COCO} is based on the smallest distance to the ROI defined by $\boldsymbol{z}$
- We compare TPB with HMO-CMA-ES [Loshchilov 16] and ...
    - ParEGO [Knowles 06], MOTPE [Ozaki 20], K-RVEA [Chugh 18], KTA2 [Song 21], and EDN-ARMOEA [Guo 22]
    - Maximum fevals: $20 \times N$, $30 \times N$, and $40 \times N$
- Parameter setting for TPB:
    - The number of weight vectors K in the 1st phase: 3
    - The budget ratio in the 1st phase r^{1st}: 0.9
    - The degree of a Bézier simplex D: 2

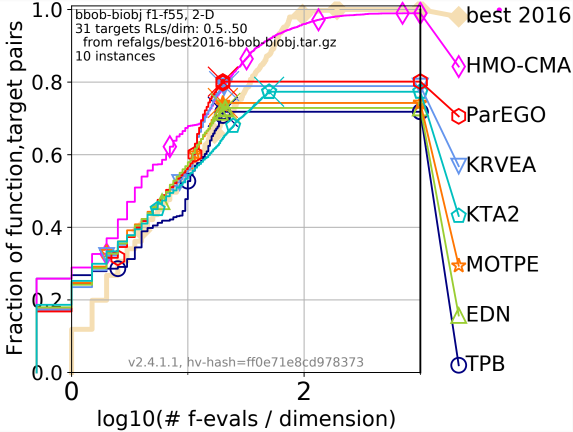
Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, Dimo Brockhoff: COCO: a platform for comparing continuous optimizers in a black-box setting. Optim. Methods Softw. 36(1): 114-144 (2021)

Dimo Brockhoff, Anne Auger, Nikolaus Hansen, Tea Tušar: Using Well-Understood Single-Objective Functions in Multiobjective Black-Box Optimization Test Suites. Evol. Comput. 30(2): 165-193 (2022)

Results on the 55 bbob-biobj problems with $N = 2$ (max. fevals $= 40$)

HMO-CMA-ES shows the best anytime performance

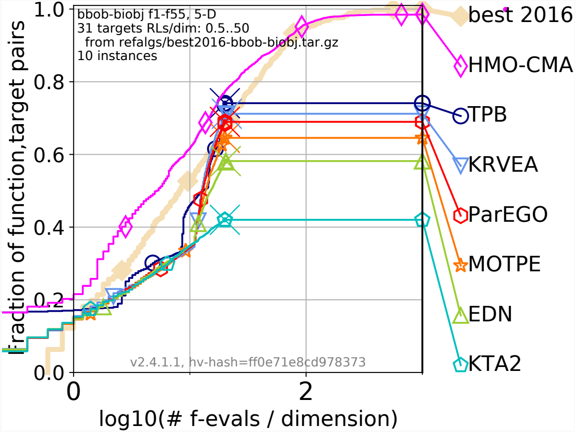
- ParEGO is the best optimizer at $20 \times N$ fevals
- TPB is the second-worst optimizer



Results on the 55 bbob-biobj problems with $N = 5$ (max. fevals $= 100$)

TPB performs better than the five model-based optimizers

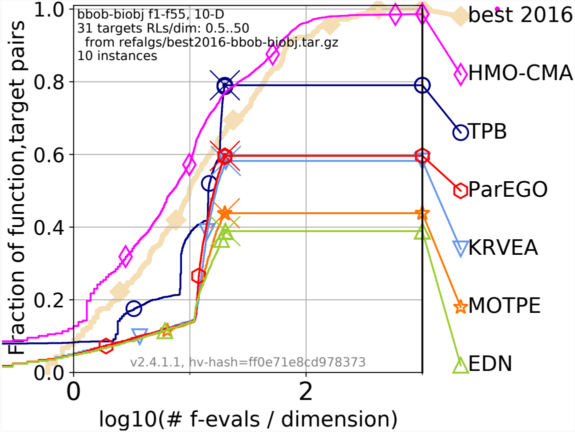
- HMO-CMA-ES performs the best



Results on the 55 `bbob-biobj` problems with $N = 10$ (max. fevals $= 200$)

TPB performs better than HMO-CMA-ES at $20 \times N$ fevals

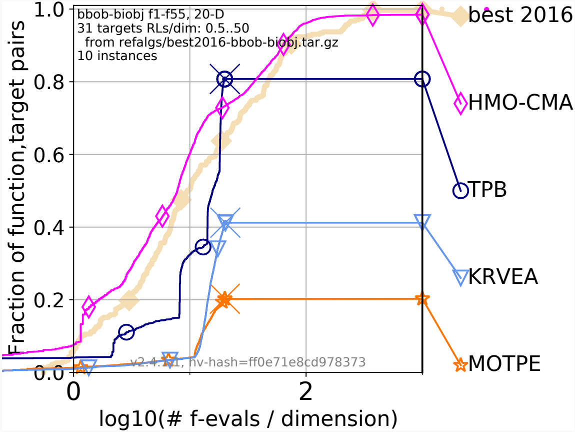
- TPB outperforms the model-based optimizers at anytime
- Because KTA2 was too time-consuming, it was removed



Results on the 55 `bbob-biobj` problems with $N = 20$ (max. fevals $= 400$)

TPB performs better than HMO-CMA-ES at $20 \times N$ fevals

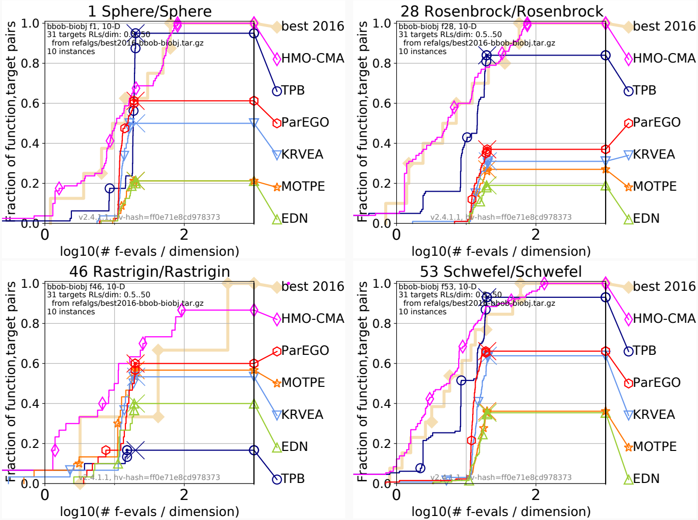
- Because ParEGO and EDN were time-consuming, they were removed



Results on f_1, f_{28}, f_{46}, f_{53} with $N = 10$ (max. fevals = 200)

We did not expect the results on f_{53} with no simplex structure

- The Bézier simplex can represent only a standard simplex



Conclusion

Proposed: A Two-Phase framework with a Bézier simplex-based interpolation method (TPB)

- The 1st phase runs BOBYQA on K scalar optimization problems
- The 2nd phase interpolates the K solutions by the Bézier simplex
  - It can describe the PS set under under certain conditions
- The performance of TPB was investigates on `bbob-biobj`
  - TPB performs better than HMO-CMA-ES for $N \geq 10$ at max. fevals
  - TPB performs better than the five model-based optimizers for $N \geq 5$
  - TPB is computatoinally cheaper than the five optimizers

TPB can give a new perspective for expensive MO optimization

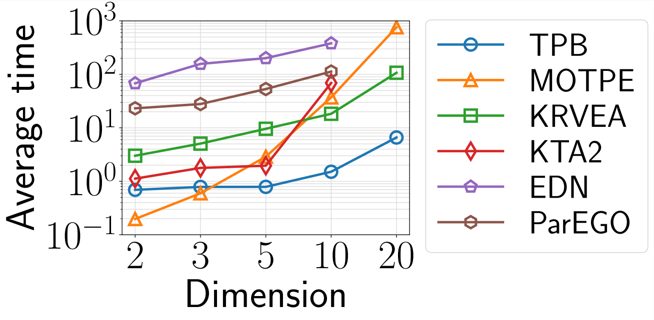
- A non-Bayesian optimization approach
- The use of the Bézier simplex for MO optimization

Future work

- An extension of TPB to optimization with more than 2 objectives
- An extension of TPB to initialize the population in EMO

Average computation time of the optimizers over the 15 instances of f_1

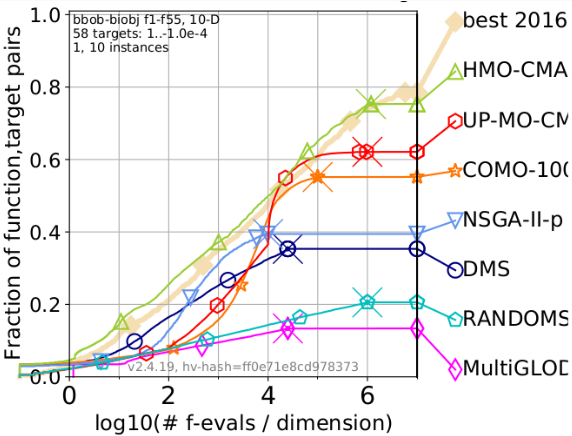
- TPB is the fastest for $N \geq 5$
- Meta-model-based optimizers are generally time-consuming
  - Especially for a larger N



Comparison of 7 optimizers on `bbob-biobj` for N. var $N = 10$ [Brockhoff 21]

HMO-CMA-ES performs the best for $10^2 \times N$ (= $1\,000$) fevals

- HMO-CMA-ES performs well for computationally expensive opt.



Dimo Brockhoff, Baptiste Plaquevent-Jourdain, Anne Auger, Nikolaus Hansen: DMS and MultiGLODS: black-box optimization benchmarking of two direct search methods on the bbob-biobj test suite. GECCO Companion 2021: 1251-1258

Definition of the Bézier simplex (M: Num. obj, N: Num. var, D: Degree)

The standard $(M-1)$-simplex

$$\Delta^{M-1} = \left\{ \boldsymbol{t} = (t_1, \dots, t_M) \in \mathbb{R}^M \middle| \sum_{m=1}^{M} t_m = 1,\ t_m \geq 0 \right\}$$

- E.g., $\boldsymbol{t} = (0.2, 0.8)$ and $\boldsymbol{t} = (1, 0)$ for $M = 2$

A set of non-negative integers

$$\mathbb{N}_D^M := \left\{ \boldsymbol{d} = (d_1, \dots, d_M) \in \mathbb{N}^M \middle| \sum_{m=1}^{M} d_m = D \right\}$$

- E.g., $\mathbb{N}_2^2 = \{\boldsymbol{d} = (d_1, d_2) \in \mathbb{N}^2 | d_1 + d_2 = 2\} = \{(2,0), (1,1), (0,2)\}$

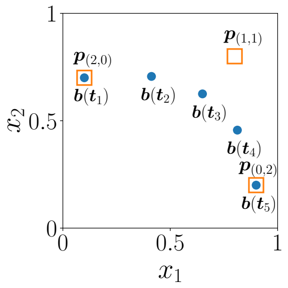
A Bézier simplex, $\boldsymbol{b} : \Delta^{M-1} \to \mathbb{R}^N$, $\boldsymbol{b} : \boldsymbol{t} \mapsto \boldsymbol{b}(\boldsymbol{t})$

$$\boldsymbol{b}(\boldsymbol{t}) = \sum_{\boldsymbol{d} \in \mathbb{N}_D^M} \binom{D}{\boldsymbol{d}} \boldsymbol{t}^{\boldsymbol{d}} \boldsymbol{p}_{\boldsymbol{d}}$$

- $\boldsymbol{p}_{\boldsymbol{d}} \in \mathbb{R}^N$ is a control point, e.g., $\boldsymbol{p}_{(2,0)} = (4.1, -3.2)$, $\boldsymbol{p}_{(1,1)} = (2.6, 0)$
- $\binom{D}{\boldsymbol{d}} := \frac{D!}{d_1! \dots d_M!}$ is a multinomial coefficient, e.g., $\binom{2}{(2,0)}$
- $\boldsymbol{t}^{\boldsymbol{d}} := t_1^{d_1} \dots t_M^{d_M}$ is a monomial for each $\boldsymbol{t}$ and $\boldsymbol{d}$, e.g., $\boldsymbol{t}^{(2,0)} = (t_1^2, t_2^0)$

Example: Bézier simplex with $M = 2$ (N. obj), $N = 2$ (N. var), and $D = 2$

$$
\begin{aligned}
\underline{\boldsymbol{b}(\boldsymbol{t})} &= \sum_{\boldsymbol{d} \in \mathbb{N}_D^M} \binom{D}{\boldsymbol{d}} \boldsymbol{t}^{\boldsymbol{d}} \boldsymbol{p}_{\boldsymbol{d}}, \\
&= \binom{2}{(2,0)} t_1^2 t_2^0 \boldsymbol{p}_{(2,0)} + \binom{2}{(1,1)} t_1^1 t_2^1 \boldsymbol{p}_{(1,1)} + \binom{2}{(0,2)} t_1^0 t_2^2 \boldsymbol{p}_{(0,2)} \\
&= \underline{t_1^2 \boldsymbol{p}_{(2,0)} + 2 t_1 t_2 \boldsymbol{p}_{(1,1)} + t_2^2 \boldsymbol{p}_{(0,2)}}
\end{aligned}
$$



- $\boldsymbol{p}_{(2,0)} = (0.1, 0.7)$
- $\boldsymbol{p}_{(1,1)} = (0.8, 0.8)$
- $\boldsymbol{p}_{(0,2)} = (0.9, 0.2)$
- $\boldsymbol{t}_1 = (1, 0), \boldsymbol{b}(\boldsymbol{t}_1) = (0.1, 0.7)$
- $\boldsymbol{t}_2 = (0.75, 0.25), \boldsymbol{b}(\boldsymbol{t}_2) \approx (0.41, 0.71)$
- $\boldsymbol{t}_3 = (0.5, 0.5), \boldsymbol{b}(\boldsymbol{t}_3) \approx (0.65, 0.63)$
- $\boldsymbol{t}_4 = (0.25, 0.75), \boldsymbol{b}(\boldsymbol{t}_4) \approx (0.81, 0.46)$
- $\boldsymbol{t}_5 = (0, 1), \boldsymbol{b}(\boldsymbol{t}_5) = (0.9, 0.2)$