From EHVI to Diffusion Models: Pareto Set Learning for Expensive Multi-Objective Bayesian Optimization

Dayeon Yoon

Uncertainty Quantification Lab
Seoul National University

June 15, 2026

Overview

1. Multi-Objective Bayesian Optimization (MOBO)
2. Hypervolume and Expected Hypervolume Improvement (EHVI)
3. EHVI-based Methods
4. Composite Diffusion Model for Pareto Set Learning (CDM-PSL)
5. Summary

Contents

Multi-Objective Optimization

We consider a black-box vector-valued objective:

$$\max_{x \in \Omega} f(x) = \left(f_1(x), f_2(x), \ldots, f_M(x)\right)^\top.$$

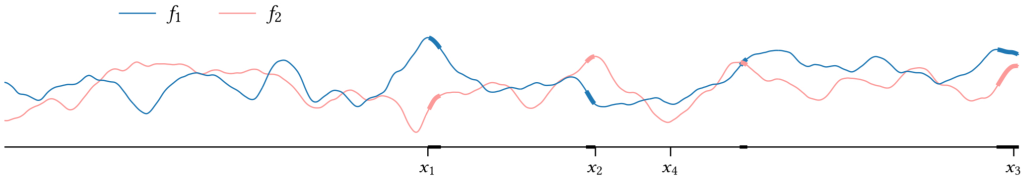


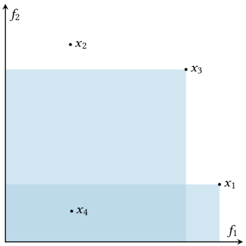
Figure: The first objective f_1 has its global maximum at x_1, which nearly coincides with the global minimum of the second objective f_2. Meanwhile, no rational agent would prefer x_4 to x_3 as the latter point achieves higher values for both objectives.

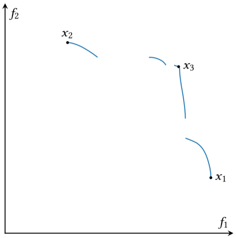
Pareto front

- We will say that a point x dominates y, denoted $y \prec x$, if

$$f_i(y) \leq f_i(x) \; \forall i, \qquad f_j(y) < f_j(x) \text{ for at least one } j.$$

- **Pareto optimal**: A point $x \in \mathcal{X}$ that is not dominated by any other point.
- **Pareto front**: The image of all Pareto optimal points.

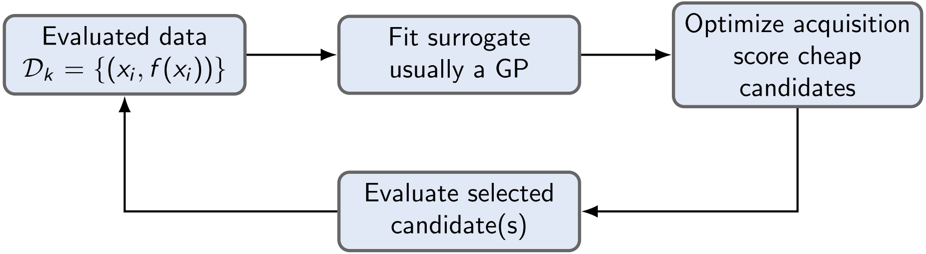




For minimization, we define $x \prec y$ if $f_i(x) \leq f_i(y) \; \forall i$, $f_j(x) < f_j(y)$ for at least one j.

Bayesian optimization loop

Bayesian optimization replaces repeated expensive evaluations with a cheaper probabilistic model.



For objective i, a GP surrogate gives a posterior mean and uncertainty:

$$f_i(x) \mid \mathcal{D}_k \sim \mathcal{N}\big(\mu_i(x), \sigma_i^2(x)\big).$$

The overall framework is called Multi-Objective Bayesian Optimization (MOBO).

Contents

Hypervolume

Let $P \subset \mathbb{R}^M$ be a non-dominated approximation set and let r be a reference point dominated by all relevant objective values [Zitzler and Thiele, 1999, Fleischer, 2002]. For minimization,

$$\mathrm{HV}(P; r) = \lambda_M \left(\bigcup_{y \in P} [y_1, r_1] \times \cdots \times [y_M, r_M] \right).$$

Given the non-dominated set P, the hypervolume improvement of a new objective value y is

$$\mathrm{HVI}(y \mid P, r) = \mathrm{HV}(P \cup \{y\}; r) - \mathrm{HV}(P; r).$$

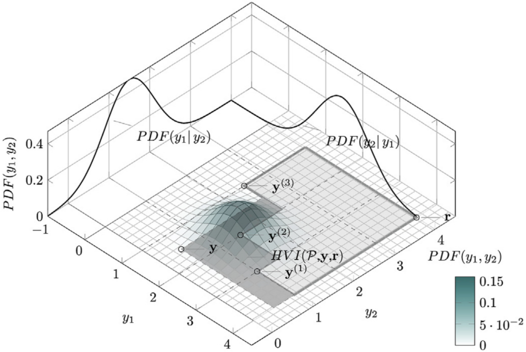
Expected Hypervolume Improvement (EHVI)

In MOBO, the value of a candidate x is uncertain under the surrogate posterior:

$$Y(x) \sim p\big(f(x) \mid \mathcal{D}_k\big).$$

EHVI averages hypervolume improvement under that posterior:

$$\mathrm{EHVI}(x) = \mathbb{E}_{Y(x)}\left[\mathrm{HVI}\big(Y(x) \mid P_k, r\big)\right].$$



Contents

Computation of EHVI

Efficient Computation of Expected Hypervolume Improvement Using Box Decomposition Algorithms. [Yang et al., 2019a]

1. **2-D case:** Partition the non-dominated space into $N_2 = n + 1$ vertical slices and sum closed-form EHVI contributions using Gaussian tail integrals and tail contributions.
2. **3-D case:** Use a sweep-line / AVL-tree box decomposition method, producing at most $N_3 = 2n + 1$ integration slices, with $2^{3-1} = 4$ closed-form terms per slice.
3. **Higher-dimensional case, $d > 3$:** Transform non-dominated-space partitioning into dominated-space partitioning using DKLV17 local lower bounds and LKF17 hyperbox decomposition, then compute EHVI.
4. **Computational costs:** For 2-D and 3-D, the cost is $O(n \log n)$ including sorting, or $\Theta(n)$ if P is already sorted; for $d > 3$, the cost is

$$O\left(2^{d-1} n^{\lfloor d/2 \rfloor}\right),$$

which is polynomial in n for fixed d but exponential in d.

Computation of qEHVI

Differentiable Expected Hypervolume Improvement for Parallel Multi-Objective Bayesian Optimization. [Daulton et al., 2020]

1. **qEHVI:** The batch version of EHVI; measures the expected hypervolume improvement from evaluating q candidate points jointly:

$$\alpha_{\mathrm{qEHVI}}(X_{\mathrm{cand}}) = \mathbb{E}\left[\mathrm{HVI}\left(f(X_{\mathrm{cand}})\right)\right].$$

2. **qEHVI cannot be calculated in closed form:** For $q > 1$, the joint HVI involves unions and intersections of improvement regions under the joint posterior, and no known analytical form exists for $q > 1$ or for correlated outcomes.

3. **Computation of qEHVI:** Use box decomposition and the inclusion–exclusion principle, then estimate the posterior expectation with Monte Carlo or quasi-Monte Carlo samples:

$$\hat{\alpha}^{N}_{\mathrm{qEHVI}}(X_{\mathrm{cand}}) = \frac{1}{N}\sum_{t=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{q}\sum_{X_j \in \mathcal{X}_j}(-1)^{j+1}\prod_{m=1}^{M}\left[z^{(m)}_{k,X_j,t} - l^{(m)}_{k}\right]_{+}.$$

   Then use auto-differentiation to compute exact gradients of this Monte Carlo estimator for gradient-based optimization.

EHVI as HVI, qEHVI as a Sum of HVI Terms

Expected Hypervolume Improvement Is a Particular Hypervolume Improvement
[Deng et al., 2025]

1. **Key idea:** By **coordinate transformation**, the authors show that EHVI can be written as a particular HVI, and qEHVI can be written as a sum of HVI terms.
2. **EHVI:** The paper defines

$$\tilde{r}_j = \mathrm{EI}(0, \mu_j, \Sigma_{jj}), \qquad \tilde{f}_j^{(i)} = \tilde{r}_j - \mathrm{EI}(f_j^{(i)}, \mu_j, \Sigma_{jj}),$$

   and then computes EHVI by applying a standard hypervolume improvement algorithm to

$$\mathrm{EHVI}(A; \mu, \Sigma) = \mathrm{HVI}(\{\tilde{r}\}, \tilde{A}).$$

3. **How the paper computes qEHVI:** The paper expresses qEHVI as a finite inclusion–exclusion sum of transformed hypervolume improvement terms:

$$\mathrm{qEHVI}(A; \mu, \Sigma) = \sum_{\emptyset \neq I \subseteq \{1,\dots,q\}} (-1)^{|I|+1} \mathrm{HVI}(\{\hat{r}(I)\}, \hat{A}(I)),$$

   where $\hat{r}(I)$ and $\hat{A}(I)$ are constructed using qMinEI.

EHVI as HVI, qEHVI as a Sum of HVI Terms

Algorithm	KMAC (Yang et al. 2019a)		DBB (Daulton, Balandat, and Bakshy 2020)		This paper	
Dimension	$m = 2, 3$	$m \geq 4$	$m = 2, 3$	$m \geq 4$	$m = 2, 3$	$m \geq 4$
Compute EHVI	$\Theta(n \log n)$ time $O(n)$ space	$O(n^{\lfloor \frac{m}{2} \rfloor})$ time $O(n^{\lfloor \frac{m}{2} \rfloor})$ space	The same method as KMAC		$\Theta(n \log n)$ time $O(n)$ space	$\boldsymbol{\tilde{O}(n^{\frac{m}{3}})}$ **time** $\boldsymbol{O(n)}$ **space**
Compute qEHVI ($q > 1$)	N/A		Monte-Carlo method with the same complexities of DBB for EHVI		**Exact method** with the same complexities of our method for EHVI	

Contents

CDM-PSL

Expensive Multi-Objective Bayesian Optimization Based on Diffusion Models [Li et al., 2025].

Problem statement

Existing Pareto set learning methods can be unstable when only a limited number of expensive evaluations are available.

Proposed idea

CDM-PSL uses a **composite diffusion model** for Pareto set learning:

- unconditional diffusion generation for diversity
- conditional guided denoising for convergence
- entropy-based objective weighting to balance multiple objectives
- a GP surrogate and batch selection inside an MOBO loop

DDPM

To sample from $x_0 \sim p_{data}$, Denoising Diffusion Probabilistic Models (DDPMs) involve two distinct stochastic processes:

- The forward process (fixed encoder):

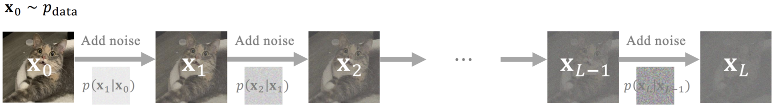


Figure: This process gradually corrupts data by injecting Gaussian noise over multiple steps.

- The reverse denoising process (learnable decoder):

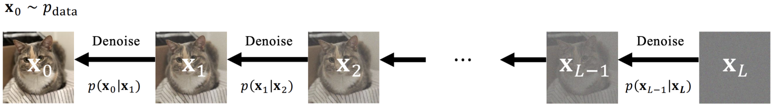


Figure: Starting from pure noise, this process iteratively denoises to generate realistic samples.

DDPM: Two Markov Chains

Denoising Diffusion Probabilistic Models (DDPMs) use two stochastic processes:

Forward process: fixed encoder

Gradually corrupts real data by adding Gaussian noise:

$$x_0 \sim p_{\text{data}}, \qquad x_T \approx \mathcal{N}(0, I).$$

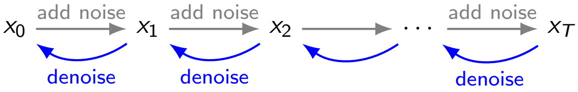
This process is fixed and non-trainable.

Reverse process: learned decoder

Starts from pure noise and denoises step-by-step:

$$x_T \sim p_{\text{prior}} = N(0, I), x_T \to x_{T-1} \to \cdots \to x_0.$$

The reverse transitions are learned by a neural network.



Forward Process: Fixed Gaussian Noise

One-step transition

For a pre-defined noise schedule $\{\beta_t\}_{t=1}^{T}$, define

$$\alpha_t := \sqrt{1 - \beta_t^2}.$$

Then

$$p(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t;\, \alpha_t x_{t-1}, \beta_t^2 I\right).$$

Equivalently,

$$x_t = \alpha_t x_{t-1} + \beta_t \epsilon_t, \qquad \epsilon_t \sim N(0, I).$$

Forward Process: Fixed Gaussian Noise

Direct sampling at any noise level

Let
$$\bar{\alpha}_t := \prod_{k=1}^{t} \alpha_k.$$

Then the noisy sample at step t can be sampled directly from x_0:

$$p_t(x_t \mid x_0) = \mathcal{N}\left(x_t;\ \bar{\alpha}_t x_0, (1 - \bar{\alpha}_t^2) I\right),$$

or equivalently,
$$x_t = \bar{\alpha}_t x_0 + \sqrt{1 - \bar{\alpha}_t^2}\,\epsilon, \qquad \epsilon \sim N(0, I).$$

As t becomes large, x_t approaches pure Gaussian noise:

$$x_T \approx p_{\text{prior}} = N(0, I).$$

Reverse Process: Learned Denoising

Learned reverse transition

DDPM approximates the unknown reverse transition by a Gaussian model:

$$p_\phi(x_{t-1} \mid x_t) = \mathcal{N}\left(x_{t-1};\, \mu_\phi(x_t, t), \sigma_t^2 I\right),$$

where μ_ϕ is predicted by a neural network.

Conditioning trick

Instead of modeling the intractable marginal reverse kernel directly, DDPM uses the tractable conditional posterior:

$$p(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(x_{t-1};\, \mu(x_t, x_0, t), \sigma_t^2 I\right).$$

Thus training reduces to matching Gaussian means:

$$\mathcal{L}_{\mathrm{DDPM}}(\phi) = \sum_{t=2}^{L} \frac{1}{2\sigma_t^2} \mathbb{E}_{x_0} \mathbb{E}_{p(x_t \mid x_0)} \left[\|\mu_\phi(x_t, t) - \mu(x_t, x_0, t)\|_2^2 \right].$$

Reverse Process: Learned Denoising

Practical noise-prediction form

$$x_t = \bar{\alpha}_t x_0 + \sqrt{1 - \bar{\alpha}_t^2}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

DDPM commonly trains a network $\epsilon_\phi(x_t, t)$ to predict the added noise:

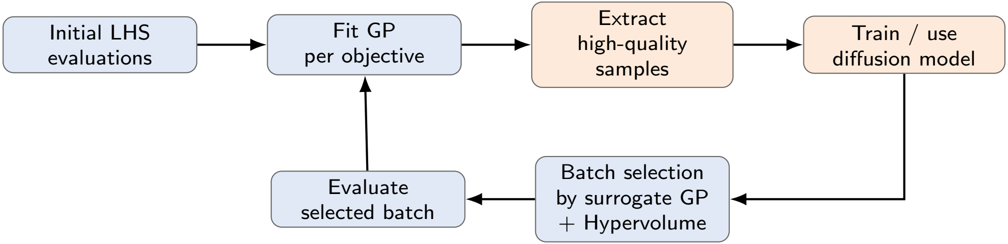
$$\mathcal{L}_{\text{simple}}(\phi) = \mathbb{E}_{t, x_0, \epsilon}\left[\|\epsilon_\phi(x_t, t) - \epsilon\|_2^2\right].$$

Sampling: Reverse Denoising After Training

The model recursively samples x_{t-1} using the learned reverse transition $p_\phi(x_{t-1} \mid x_t)$:

$$x_{t-1} \leftarrow \underbrace{\frac{1}{\alpha_t}\left(x_t - \frac{1 - \alpha_t^2}{\sqrt{1 - \bar{\alpha}_t^2}}\epsilon_\phi(x_t, t)\right)}_{\mu_\phi(x_t, t)} + \sigma_t \epsilon_t, \qquad epsilon_t \sim \mathcal{N}(0, I).$$

Algorithmic overview of CDM-PSL



Three main components

1. **Data extraction**: choose promising samples from evaluated data.
2. **Diffusion model training**: learn the distribution of good decision vectors.
3. **Composite generation**: combine guided conditional sampling with unconditional sampling.

Component 1: data extraction

The diffusion model is not trained on all evaluated points; it is trained on selected high-quality samples. The paper uses shift-based density estimation to score evaluated samples:

$$\mathrm{Fitness}(p) = \min_{q \in X^k \setminus \{p\}} \sqrt{\sum_{i=1}^{M} \left(\max\{0, f_i(q) - f_i(p)\}\right)^2}.$$

Intuition

- Keep samples that are promising for convergence and diversity.
- Use those samples as empirical evidence of the Pareto-set distribution.

Component 2: diffusion model training

Given selected samples X_k^*, the forward process gradually adds Gaussian noise:

$$X_{k,t}^* = \sqrt{1-\beta_t}\, X_{k,t-1}^* + \sqrt{\beta_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I).$$

The network learns to predict the added noise:

$$\mathcal{L}(\theta) = \frac{1}{H} \sum_{i=1}^{H} \left\| \epsilon - \epsilon_\theta(x_{k,i,t}^*, t) \right\|^2.$$

Component 3: conditional generation by guided denoising

To improve sample quality, CDM-PSL guides the reverse process using surrogate-gradient information:

$$X_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(X_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(X_t, t)\right) + \sigma_t^2 \hat{g} + \sigma_t z.$$

$$\hat{g} = \sum_{j=1}^{M} W_j \nabla a_j(x), \qquad a_j(x) : \text{GP-LCB surrogate for the } j\text{-th objective}$$

Denoising term

Moves noisy samples back toward the learned distribution of high-quality samples.

Guidance term $\hat{g}$

Uses GP surrogate gradients to bias the reverse trajectory toward improved objective values.

Component 3: entropy-based objective weighting

For each objective j, normalize objective values, form probabilities, and compute entropy:

$$\tilde{y}_{ij} = \frac{y_{ij} - \min(y_j)}{\max(y_j) - \min(y_j)}, \qquad P_{ij} = \frac{\tilde{y}_{ij}}{\sum_{k=1}^{N} \tilde{y}_{kj}},$$

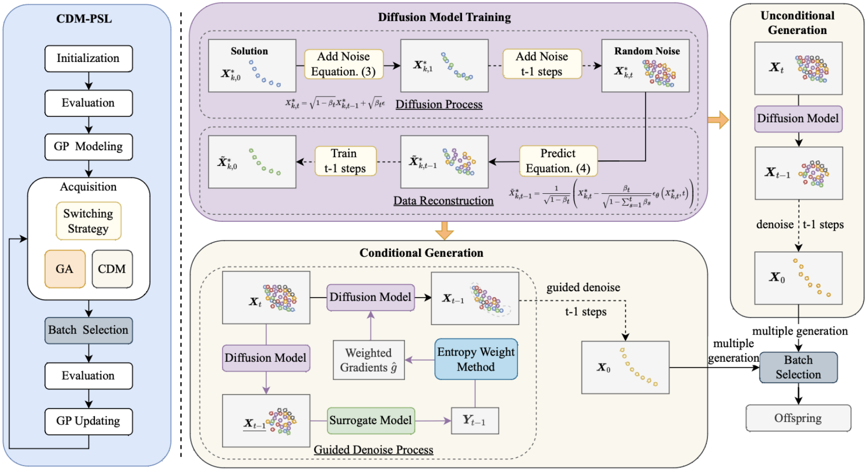
$$E_j = -\frac{1}{\ln N} \sum_{i=1}^{N} P_{ij} \ln(P_{ij} + \eta), \qquad W_j = \frac{1 - E_j}{\sum_{k=1}^{M} (1 - E_k)}.$$

Here, $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$, where N is the total number of samples and M is the number of objectives.

Intuition

E_j: minimized at 0 when $P_{1j} = 1, P_{ij} = 0 \forall i \neq 1$. W_j: maximized when $E_j = 0$. Therefore, this gives more weight to objectives with greater variation among individuals.

Diffusion model framework



Conditional Generation : Unconditional Generation $= 1 : 10$

Selection strategy after generation

Batch selection

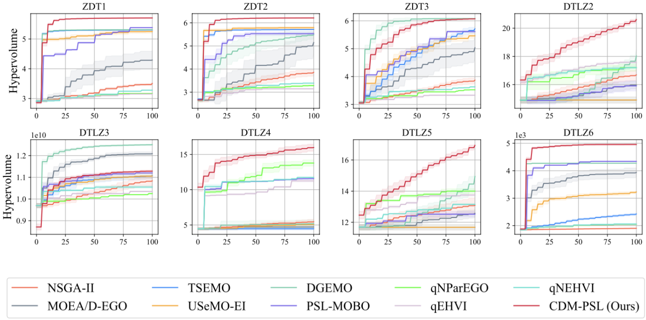
- Use the GP surrogate to estimate Hypervolume (HV).
- Select a small subset $X_k^B = \{x_b \mid b = 1, \ldots, B\}$ with high HV values.
- Evaluate only the selected batch with the expensive black-box function.

Operator switching

- Monitor hypervolume growth over an iteration window.
- If growth stagnates, switch the offspring generator to a genetic algorithm.
- This is intended to reduce local stagnation caused by a single operator.

Experimental Study

The paper evaluates CDM-PSL on synthetic benchmarks and real-world problems, comparing against classical and recent MOBO / EMO baselines.



Parameter settings

The DM is designed with two linear layers, each containing 128 hidden units, and is trained for 4000 epochs with a batch size of 1024. The hyperparameter step t is 25, and the noise level is defined in the range from 1e-5 to 0.5e-1.

Contents

Summary

Main problem

Multi-Objective Bayesian Optimization (MOBO) aims to optimize several expensive black-box objectives at the same time. Since objectives can conflict, the goal is not a single optimum but a set of Pareto-optimal solutions.

	EHVI as HVI [Deng et al., 2025]	**CDM-PSL** [Li et al., 2025]
Core idea	EHVI can be converted into a hypervolume-improvement HVI problem.	Uses a composite diffusion model to learn the distribution of Pareto-set samples.
Treatment of q batch optimization	qEHVI is expressed as a finite sum of particular HVI terms, giving an exact analytic expression for $q > 1$.	Supports batch optimization by generating many offspring candidates through CDM-PSL.
Candidate generation	Provides a more efficient exact acquisition-function computation that can be used inside an MOBO candidate-selection loop.	Generates candidates directly using the composite diffusion model: conditional and unconditional generation.

References I

Daulton, S., Balandat, M., and Bakshy, E. (2020).
Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization.
In *Advances in Neural Information Processing Systems*, volume 33, pages 9851–9864.

Deng, J., Sun, J., Zhang, Q., and Li, H. (2025).
Expected hypervolume improvement is a particular hypervolume improvement.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16217–16225.

Fleischer, M. (2002).
The measure of pareto optima: Applications to multi-objective metaheuristics.
Technical Report TR 2002-32, University of Maryland, Institute for Systems Research.

Garnett, R. (2023).
Bayesian Optimization.
Cambridge University Press.

References II

Ho, J., Jain, A., and Abbeel, P. (2020).
Denoising diffusion probabilistic models.
In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851.

Lai, C.-H., Song, Y., Kim, D., Mitsufuji, Y., and Ermon, S. (2025).
The principles of diffusion models.
arXiv preprint arXiv:2510.21890.

Li, B., Di, Z., Lu, Y., Qian, H., Wang, F., Yang, P., Tang, K., and Zhou, A. (2025).
Expensive multi-objective bayesian optimization based on diffusion models.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27063–27071.

Lin, X., Yang, Z., Zhang, X., and Zhang, Q. (2022).
Pareto set learning for expensive multi-objective optimization.
In *Advances in Neural Information Processing Systems*, volume 35, pages 19231–19247.

References III

Yang, K., Emmerich, M., Deutz, A., and Bäck, T. (2019a).
Efficient computation of expected hypervolume improvement using box decomposition algorithms.
Journal of Global Optimization, 75:3–34.

Yang, K., Emmerich, M., Deutz, A., and Bäck, T. (2019b).
Multi-objective bayesian global optimization using expected hypervolume improvement gradient.
Swarm and Evolutionary Computation, 44:945–956.

Zitzler, E. and Thiele, L. (1999).
Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach.
IEEE Transactions on Evolutionary Computation, 3(4):257–271.