Evolutionary computation for stochastic problems

Frank Neumann[1], Aneta Neumann[1] and Hemant Singh[2]

[1]Optimisation and Logistics,
School of Computer and Mathematical Sciences,
The University of Adelaide, Adelaide, SA 5005, Australia

[2]School of Engineering and Information Technology,
The University of New South Wales, Canberra, ACT 2600, Australia

Emails: frank.neumann@adelaide.edu.au,
aneta.neumann@adelaide.edu.au, h.singh@unsw.edu.au

Instructors

Frank Neumann is a Professor and the leader of the Optimisation and Logistics group at the University of Adelaide and an Honorary Professorial Fellow at the University of Melbourne. His current position is funded by the Australian Research Council through a Future Fellowship and focuses on AI-based optimisation methods for problems with stochastic constraints. Frank has been the general chair of the ACM GECCO 2016 and co-organised ACM FOGA 2013 in Adelaide. He is an Associate Editor of the journals "Evolutionary Computation" (MIT Press) and ACM Transactions on Evolutionary Learning and Optimization. In his work, he considers algorithmic approaches in particular for combinatorial and multi-objective optimization problems and focuses on theoretical aspects of evolutionary computation as well as high impact applications in the areas of cybersecurity, renewable energy, logistics, and mining.

Aneta Neumann is a researcher in the School of Computer and Mathematical Sciences at the University of Adelaide, Australia, and focuses on real world problems using evolutionary computation methods. She is also part of the Integrated Mining Consortium at the University of Adelaide. Aneta graduated in Computer Science from the Christian-Albrechts-University of Kiel, Germany, and received her PhD from the University of Adelaide, Australia. She served as the co-chair of the Real-World Applications track at GECCO 2021 and GECCO 2022 and is a co-chair of the Genetic Algorithms track at GECCO 2023 and GECCO 2024. Her main research interests are bio-inspired computation methods, with a particular focus on dynamic and stochastic multi-objective optimization for real-world problems that occur in the mining industry, defence, cybersecurity, creative industries, and public health.

Hemant Kumar Singh is an Associate Professor at the School of Engineering and Information Technology at the University of New South Wales (UNSW), Australia. He completed his PhD from UNSW in 2011 and B.Tech in Mechanical Engineering from Indian Institute of Technology (IIT) Kanpur in 2007. He worked with General Electric (GE) Aviation at John F. Welch Technology Centre as a Lead Engineer during 2011-13. His research interests include development of evolutionary computation methods to deal with various challenges such as multiple objectives, constraints, uncertainties, hierarchical (bi-level) objectives, and decision-making. He is an Associate Editor for IEEE Transactions on Evolutionary Computation and has been in the organizing team of several conferences, e.g. ACM GECCO (Competition chair 2024, RWACMO workshop 2018-21), IEEE CEC (Program chair 2021), SSCI (MCDM co-chair 2020-23).

Introduction

Introduction

- Many real-world optimization problems include uncertainties which effect the quality of solutions.

For example:

- Evaluating the performance of a machine does not give exactly the same result even if all parameters are the same.
- Traveling from A to B on a planned route takes different amounts of times and it's hard to predict the exact duration of the trip.

Introduction

- Noise can occur at different stages.
- Theoretical studies distinguish where noise is applied before or after functions evaluation.

Prior noise:

- effects the solution before the evaluations.
- Here a solution component might change before evaluation.

Posterior noise:

- adds noise to the functions value (dependent on a known/unknown distribution)

Introduction

- Many difficulties arise in stochastic optimization as the known realization is not known at the time of optimization.
- Knowledge about the ground truth is often gathered while implementing a solution.
- This new knowledge can potentially be used to reduce uncertainties for solution components that have not already been implemented.

- This tutorial will cover a selected set of topics for stochastic optimization using evolutionary algorithms.

Evolutionary algorithms (EAs)

- Evolutionary algorithms are general purpose algorithms.
- follow Darwin's principle (survival of the fittest).
- work with a set of solutions called population.
- parent population produces offspring population by variation operators (mutation, crossover).
- select individuals from the parents and children to create new parent population.
- Iterate the process until a "good solution" has been found.

EAs are adaptive and often yield good solutions for complex, dynamic and/or stochastic problems.

Outline

- Introduction
- Real-World Example from Mine Planning
- Knapsack Problem with Stochastic Profits
- Submodular Optimization Problems with Chance Constraints
- 3D Pareto Optimization for Stochastic Problems
- Many Objective Optimization and Robustness
- Conclusions

Real-World Example
Mine Planning

Significance of Uncertainty Modeling and Quantification

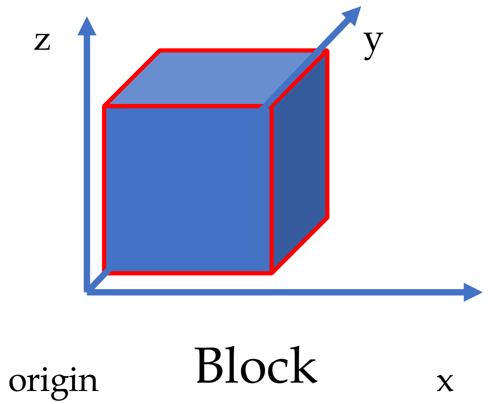
Metal Supply-Demand Conundrum

Increasing metal demand	Decreasing ore grades
Increasing global population	Increasing ore variability

- Optimising revenue taking into account uncertainties in the block model.
- Capture the maximum economic potential of mineral resources.
- Optimising a strategic long term plan based on geological estimation uncertainty in the mine scheduling process.

Background on Uncertainties in the Block Model

- Block model is a collection of spatially located blocks with set of attributes (rock lithological domain, ore grade, rock density).
- Example data based on the Neuronal Network prediction:



Predicted copper grade (%Cu): 0.41285940

0.38882205

0.24948858

0.24949805

0.24949934

The Profit-based Discounted Knapsack Problem

Motivation

- Often benefits of a given goal function can be impacted by uncertainties (e.g., profit obtainable from blocks in mining).
- Our goal is to maximize profit, but we would also like to guarantee that the profit only drops with a small probability below an optimized profit value.
- Consider the classical knapsack problem with stochastic profit.
- We aim to maximize the profit value P for which we can guarantee that it’s achieved by the solution presented by our algorithm with probability at least $1 - \alpha_p$.

Stochastic Knapsack Problem

In the classical problem, there are given *n* items *1, . . . , n* where each item has a profit p_i and a weight w_i, the goal is to maximize the profit $p(x) = \sum_{i=1}^{n} p_i x_i$ under the condition that $w(x) = \sum_{i=1}^{n} w_i x_i \leq B$ for a given weight bound B holds.

We consider the stochastic version, where the profits p_i are <u>stochastic</u>, and the weights are still deterministic.

Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profits

Goal: is to maximize the profit P among solutions x in $\{0,1\}^n$ for which we can guarantee that there is only a small probability α_p of dropping below P.

$$\max P \qquad s.t.\ Pr(p(x) < P) \leq \alpha_p$$
$$w(x) \leq B$$

We consider (a) the (1+1) EA, (b) the (1+1) EA with heavy tail mutation, (c) (μ + 1) EA with the heavy tail mutation and the Discounted Greedy Uniform Crossover.

Aneta Neumann, Xie Yue, Frank Neumann,
Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profit, PPSN (1) 2022

Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profits

Algorithm 3: $(\mu + 1)$ EA

Randomly generate μ initial solutions as the initial population P;
while *stopping criterion not meet* **do**
    Let x and y be two different individual from P chosen uniformly at random;
    if $rand([0,1]) \leq p_c$ **then**
        apply the discounted greedy uniform crossover operator to x and y to produce an offspring z.
    else
        Choose one individual x from P uniformly at random and let z be a copy of x.
    end if
    apply the heavy-tail mutation operator to z;
    if $f(z) \geq f(x)$ **then**
        $P \leftarrow (P \setminus \{x\}) \cup \{z\}$;
    else
        if $f(z) \geq f(y)$ **then**
            $P \leftarrow (P \setminus \{y\}) \cup \{z\}$;
        end if
    end if
end while

Setting for Stochastic Profit

We consider different chance constraint settings in terms of the uncertainty level δ_p, and the probability bound α_p.

Chebyshev's Inequality:

We assume that for a given solution only the expected value $\mu(x)$ and the variance $v(x)$ are known.

Hoeffding Bound:

We assume that each element i takes on a profit $p_i \in [\mu_i - \delta_p, \mu_i + \delta_p]$ independently of the other items.

Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profits

The fitness of a search point $x \in \{0, 1\}^n$ is given by

$$f(x) = (u(x), \hat{p}(x))$$

where $u(x) = \max\{w(x) - B, 0\}$ is the amount of constraint violation of the bound B by the weight that should be minimized and $\hat{p}(x)$ is the discounted profit of solution x that should be maximized.

$$\hat{p}_{Cheb}(x) = \mu(x) - \sqrt{(1-\alpha_p)/\alpha_p} \cdot \sqrt{v(x)}$$

$$\hat{p}_{Hoef}(x) = \mu(x) - \delta_p \cdot \sqrt{\ln(1/\alpha_p) \cdot 2|x|_1}$$

Comparison of Chebyshev and Hoeffding based fitness functions

$$\hat{p}_{Hoef}(x) \geq \hat{p}_{Cheb}(x) \iff \ln(1/\alpha_p) \cdot \alpha_p/(1-\alpha_p) \leq 1/6$$

Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profits

Settings: consider all combinations of α_p = 0.1, 0.01, 0.001, and δ_p = 25, 50, uncorrelated and bounded strong correlated ones, with n = 100, 300, 500 items.

Results for the Chebyshev based function p_{Cheb}.

				(1+1) EA			(1+1) EA-HT			(μ+1) EA		
	B	α_p	δ_p	$\hat{p}_{Cheb}$	std	stat	$\hat{p}_{Cheb}$	std	stat	$\hat{p}_{Cheb}$	std	stat
uncorr_100	2407	0.1	25	11073.5863	36.336192	$2^{(*)}, 3^{(*)}$	11069.0420	46.285605	$1^{(*)}, 3^{(*)}$	11057.4420	59.495722	$1^{(*)}, 2^{(*)}$
			50	10863.1496	85.210231	$2^{(*)}, 3^{(*)}$	10889.4840	37.175095	$1^{(*)}, 3^{(*)}$	10883.7163	53.635972	$1^{(*)}, 2^{(*)}$
		0.01	25	10641.9089	63.402329	$2^{(*)}, 3^{(*)}$	10664.5974	29.489838	$1^{(*)}, 3^{(*)}$	10655.7251	43.869265	$1^{(*)}, 2^{(*)}$
			50	10054.6427	49.184220	$2^{(*)}, 3^{(*)}$	10066.2854	36.689426	$1^{(*)}, 3^{(*)}$	10064.8734	39.556767	$1^{(*)}, 2^{(*)}$
		0.001	25	9368.33053	46.894877	$2^{(*)}, 3^{(*)}$	9368.2483	34.904933	$1^{(*)}, 3^{(*)}$	9365.5257	40.458098	$1^{(*)}, 2^{(*)}$
			50	7475.44948	50.681386	$2^{(*)}, 3^{(*)}$	7490.6387	27.819516	$1^{(*)}, 3^{(*)}$	7497.5054	14.098629	$1^{(*)}, 2^{(*)}$
strong_100	4187	0.1	25	8638.0428	68.740095	$2^{(-)}, 3^{(-)}$	8698.2592	64.435352	$1^{(+)}, 3^{(*)}$	8707.9271	49.633473	$1^{(+)}, 2^{(*)}$
			50	8441.9311	80.335771	$2^{(-)}, 3^{(-)}$	8483.1151	45.284814	$1^{(+)}, 3^{(*)}$	8481.0022	55.979520	$1^{(+)}, 2^{(*)}$
		0.01	25	8214.8029	56.705379	$2^{(-)}, 3^{(-)}$	8230.9642	42.084563	$1^{(+)}, 3^{(*)}$	8210.1448	55.148757	$1^{(+)}, 2^{(*)}$
			50	7512.3033	71.115520	$2^{(-)}, 3^{(-)}$	7563.5495	37.758812	$1^{(+)}, 3^{(*)}$	7554.7382	53.030592	$1^{(+)}, 2^{(*)}$
		0.001	25	6771.7849	58.314395	$2^{(-)}, 3^{(-)}$	6797.0376	42.944371	$1^{(+)}, 3^{(*)}$	6793.0387	43.492135	$1^{(+)}, 2^{(*)}$
			50	4832.2084	88.887119	$2^{(-)}, 3^{(-)}$	4929.1483	52.858392	$1^{(+)}, 3^{(*)}$	4902.0006	44.976733	$1^{(+)}, 2^{(*)}$

Main Results:

- introduced the knapsack problem with chance constrained profits.
- presented fitness functions for different stochastic settings that allow to maximize the profit value P such that the probability of obtaining a profit less than P is upper bounded by α_p.

Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profits

Results for the Hoeffding based function p_{Hoef}.

	B	α_p	δ_p	(1+1) EA $\hat{p}_{Hoef}$	(1+1) EA std	(1+1) EA stat	(1+1) EA-HT $\hat{p}_{Hoef}$	(1+1) EA-HT std	(1+1) EA-HT stat	(μ+1) EA $\hat{p}_{Hoef}$	(μ+1) EA std	(μ+1) EA stat
uncorr_100	2407	0.1	25	10948.7292	90.633230	$2^{(-)}, 3^{(*)}$	11016.8190	49.768932	$1^{(+)}, 3^{(+)}$	10981.3880	37.569308	$1^{(*)}, 2^{(-)}$
			50	10707.1094	43.869094	$2^{(-)}, 3^{(*)}$	10793.1175	58.150646	$1^{(+)}, 3^{(+)}$	10708.6094	44.384035	$1^{(*)}, 2^{(-)}$
		0.01	25	10836.0906	91.332983	$2^{(-)}, 3^{(*)}$	10928.3054	45.464936	$1^{(+)}, 3^{(+)}$	10866.9831	45.408500	$1^{(*)}, 2^{(-)}$
			50	10482.6216	46.444510	$2^{(-)}, 3^{(*)}$	10611.1895	69.341044	$1^{(+)}, 3^{(+)}$	10477.2328	47.065426	$1^{(*)}, 2^{(-)}$
		0.001	25	10765.3289	68.565293	$2^{(-)}, 3^{(*)}$	10862.7124	49.091526	$1^{(+)}, 3^{(+)}$	10784.7286	38.187390	$1^{(*)}, 2^{(-)}$
			50	10263.9426	90.504901	$2^{(-)}, 3^{(*)}$	10487.5621	32.625499	$1^{(+)}, 3^{(+)}$	10309.8572	44.811326	$1^{(*)}, 2^{(-)}$
strong_100	4187	0.1	25	8553.1744	74.046187	$2^{(-)}, 3^{(*)}$	8640.05156	39.413105	$1^{(+)}, 3^{(+)}$	8588.4894	53.878268	$1^{(*)}, 2^{(-)}$
			50	8264.8129	63.309264	$2^{(-)}, 3^{(*)}$	8398.4354	46.013234	$1^{(+)}, 3^{(+)}$	8273.9670	41.403505	$1^{(*)}, 2^{(-)}$
		0.01	25	8422.9258	70.464985	$2^{(-)}, 3^{(*)}$	8540.2095	63.072560	$1^{(+)}, 3^{(+)}$	8447.8489	59.841707	$1^{(*)}, 2^{(-)}$
			50	7996.0193	65.822419	$2^{(-)}, 3^{(*)}$	8181.2980	45.667034	$1^{(+)}, 3^{(+)}$	8013.1724	56.445427	$1^{(*)}, 2^{(-)}$
		0.001	25	8338.5159	57.880350	$2^{(-)}, 3^{(*)}$	8460.7513	53.402755	$1^{(+)}, 3^{(+)}$	8362.9405	51.607219	$1^{(*)}, 2^{(-)}$
			50	7794.1245	80.411946	$2^{(-)}, 3^{(*)}$	8017.8843	53.266120	$1^{(+)}, 3^{(+)}$	7833.5575	37.293481	$1^{(*)}, 2^{(-)}$
uncorr_300	6853	0.1	25	33831.9693	181.485453	$2^{(-)}, 3^{(-)}$	34118.7631	200.095911	$1^{(+)}, 3^{(*)}$	34129.8891	172.788856	$1^{(+)}, 2^{(*)}$
			50	33380.4952	157.014552	$2^{(-)}, 3^{(-)}$	33715.2964	199.074378	$1^{(+)}, 3^{(*)}$	33662.2668	124.206823	$1^{(+)}, 2^{(*)}$
		0.01	25	33655.5737	234.136500	$2^{(-)}, 3^{(-)}$	34014.3456	200.488072	$1^{(+)}, 3^{(*)}$	33962.8643	161.560953	$1^{(+)}, 2^{(*)}$
			50	32933.5174	291.623690	$2^{(-)}, 3^{(-)}$	33327.8984	235.915481	$1^{(+)}, 3^{(*)}$	33277.4015	142.387738	$1^{(+)}, 2^{(*)}$
		0.001	25	33515.7445	219.707660	$2^{(-)}, 3^{(-)}$	33806.1572	184.532069	$1^{(+)}, 3^{(*)}$	33835.4528	149.327823	$1^{(+)}, 2^{(*)}$
			50	32706.4466	176.599463	$2^{(-)}, 3^{(-)}$	33112.7494	177.218747	$1^{(+)}, 3^{(*)}$	32940.4397	173.836538	$1^{(+)}, 2^{(*)}$
strong_300	13821	0.1	25	24602.1254	171.596469	$2^{(-)}, 3^{(-)}$	24848.3209	100.078545	$1^{(+)}, 3^{(+)}$	24734.7210	127.268428	$1^{(+)}, 2^{(-)}$
			50	24184.8938	125.755762	$2^{(-)}, 3^{(-)}$	24457.7279	118.679623	$1^{(+)}, 3^{(+)}$	24205.9660	116.049342	$1^{(+)}, 2^{(-)}$
		0.01	25	24476.1412	159.274566	$2^{(-)}, 3^{(-)}$	24638.0751	105.088783	$1^{(+)}, 3^{(+)}$	24538.4199	101.959196	$1^{(+)}, 2^{(-)}$
			50	23653.3561	225.087307	$2^{(-)}, 3^{(-)}$	24060.0806	87.242862	$1^{(+)}, 3^{(+)}$	23830.8655	85.829604	$1^{(+)}, 2^{(-)}$
		0.001	25	24256.4468	173.293324	$2^{(-)}, 3^{(-)}$	24558.9506	105.253206	$1^{(+)}, 3^{(+)}$	24345.4340	144.094192	$1^{(+)}, 2^{(-)}$
			50	23377.6774	143.350899	$2^{(-)}, 3^{(-)}$	23843.7258	114.231223	$1^{(+)}, 3^{(+)}$	23520.1166	112.403711	$1^{(+)}, 2^{(-)}$

Multi-Objective Approaches for chance-constrained submodular problems

Motivation

Many problems involve stochastic components and constraints that can only be violated with a small probability.

Example of such a constraint:

$$\Pr[W(S) > B] \leq \alpha.$$

Such constraints are known as chance constraints.

We investigate bio-inspired algorithms for submodular problems with chance constraints.

Submodular Optimization Problems

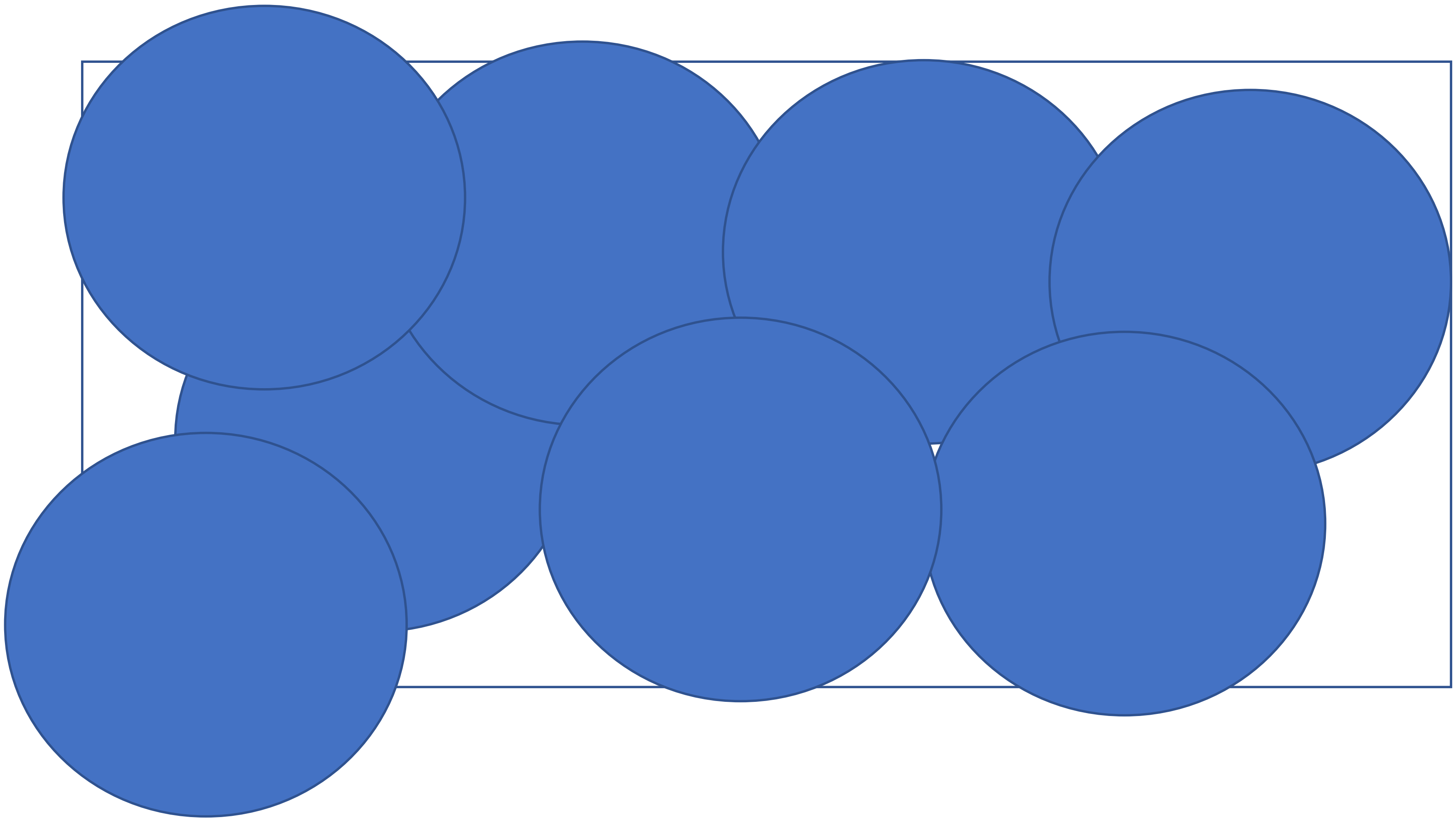
Submodular functions are functions that allow to model problems with diminishing returns.

They allow to model many real-world optimization problems.

Evolutionary multi-objective algorithms using Pareto optimization approaches have been shown to be very successful for these problems, both from a theoretical and empirical perspective.

Example: Sensor placement

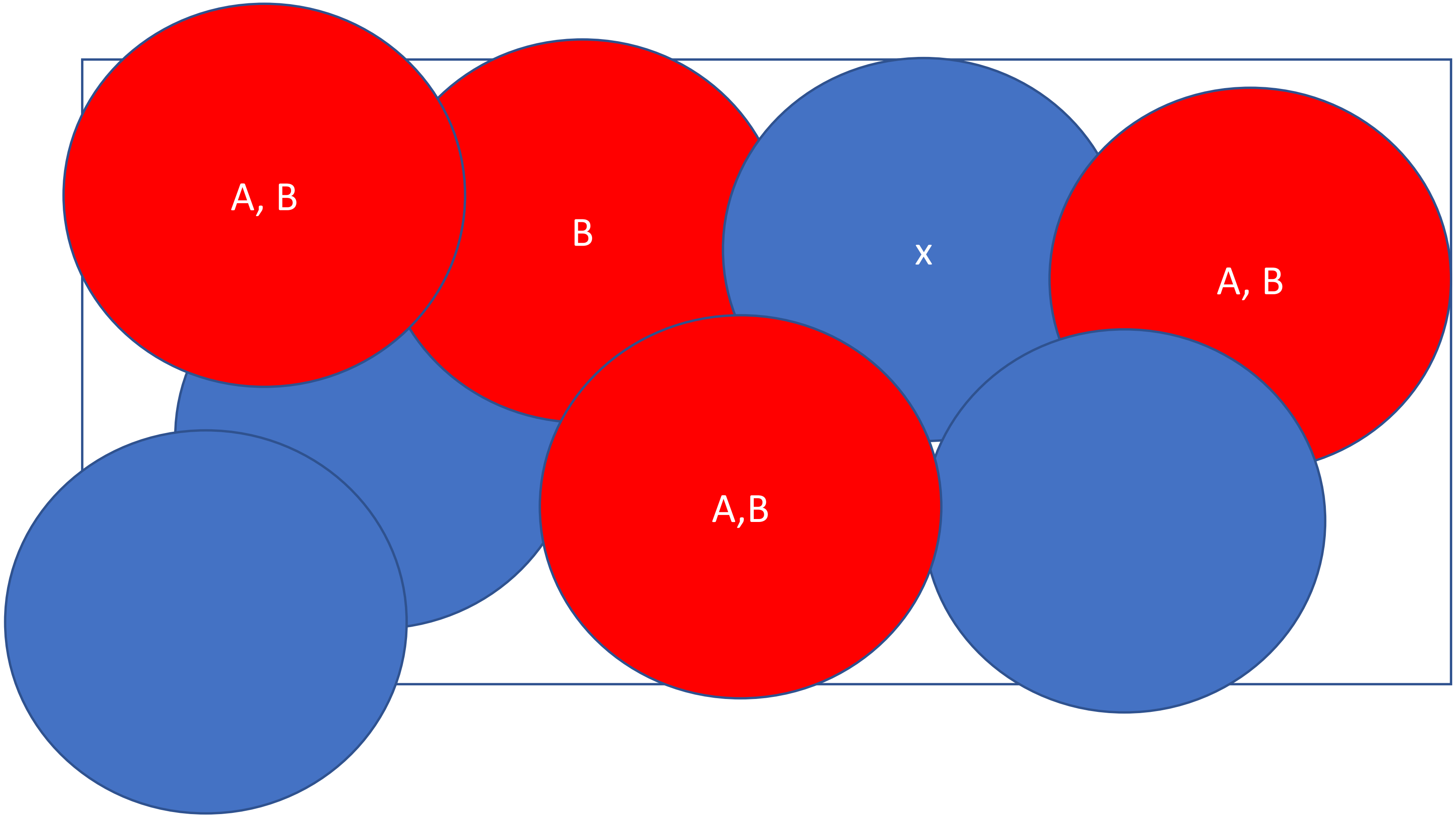
Cover the largest possible area by selecting k sensors:



Example: Sensor placement

Cover the largest possible area by selecting k sensors:

Submodular: $A \subseteq B \subseteq X$ and $x \in X \setminus B$, $f(B \cup \{x\}) - f(B) \leq f(A \cup \{x\}) - f(A)$.



Examples of Submodular Functions

- *Linear functions:* All linear functions $f\colon 2^X \to \mathbb{R}$ with $f(A) = \sum_{i \in A} w_i$ for some weights $w\colon X \to \mathbb{R}$ are submodular. If $w_i \geq 0$ for all $i \in X$, then f is also monotone.
- *Cut:* Given a graph $G = (V, E)$ with nonnegative edge weights $w\colon E \to \mathbb{R}_{\geq 0}$. Let $\delta(S)$ be the set of all edges that contain both a vertex in S and $V \setminus S$. The cut function $w(\delta(S))$ is symmetric and submodular but not monotone.
- *Coverage:* Let the ground set be $X = \{1, 2, \ldots, n\}$. Given a universe U with n subsets $A_i \subseteq U$ for $i \in X$, and a non-negative weight function $w\colon U \to \mathbb{R}_{\geq 0}$. The coverage function $f\colon 2^X \to \mathbb{R}$ with $f(S) = |\bigcup_{i \in S} A_i|$ and the weighted coverage function f' with $f'(S) = w(\bigcup_{i \in S} A_i) = \sum_{u \in \bigcup_{i \in S} A_i} w(u)$ are monotone submodular.
- *Rank of a matroid:* The rank function $r(A) = \max\{|S|\colon S \subseteq A, S \in \mathcal{I}\}$ of a matroid $(X, \mathcal{I})$ is monotone submodular.

Chance Constraints

One of the difficulties lies in evaluating whether a given solution fulfills the chance constraint.

For independent Normally distributed random variables this can be done exactly.

In other cases, one way is to use sampling to estimate the probability of violating the constraint.

Another way is to use surrogate functions such as Chernoff bounds and Chebyshev's inequality to determine whether a solution is feasible.

These bounds don't allow for a precise calculation for the probability of a constraint violation.

However, the give an upper bound and a solution is accepted if its upper bound is at most α .

We establish conditions based on the expected cost, the variance, and constraint B to show that a given solution is feasible.

Setting for Random Weights

We consider two settings for random weights of a given set of items.

Both settings assume that the weights of the items are chosen independent of each other.

Uniform independent and identically distributed (IID) weights:

$$W(s) \in [a - \delta, a + \delta] \ (\delta \leq a).$$

Uniform Weights with same dispersion

$$W(s) \in [a(s) - \delta, a(s) + \delta].$$

Chance Constraint Conditions

Chernoff:

Lemma 1. *Let* $W(s) \in [a(s)-\delta, a(s)+\delta]$ *be independently chosen uniformly at random. If*

$$(B - E[W(X)]) \geq \sqrt{3\delta k \ln(1/\alpha)},$$

where $k = |X|$, *then* $\Pr[W(X) > B] \leq \alpha$.

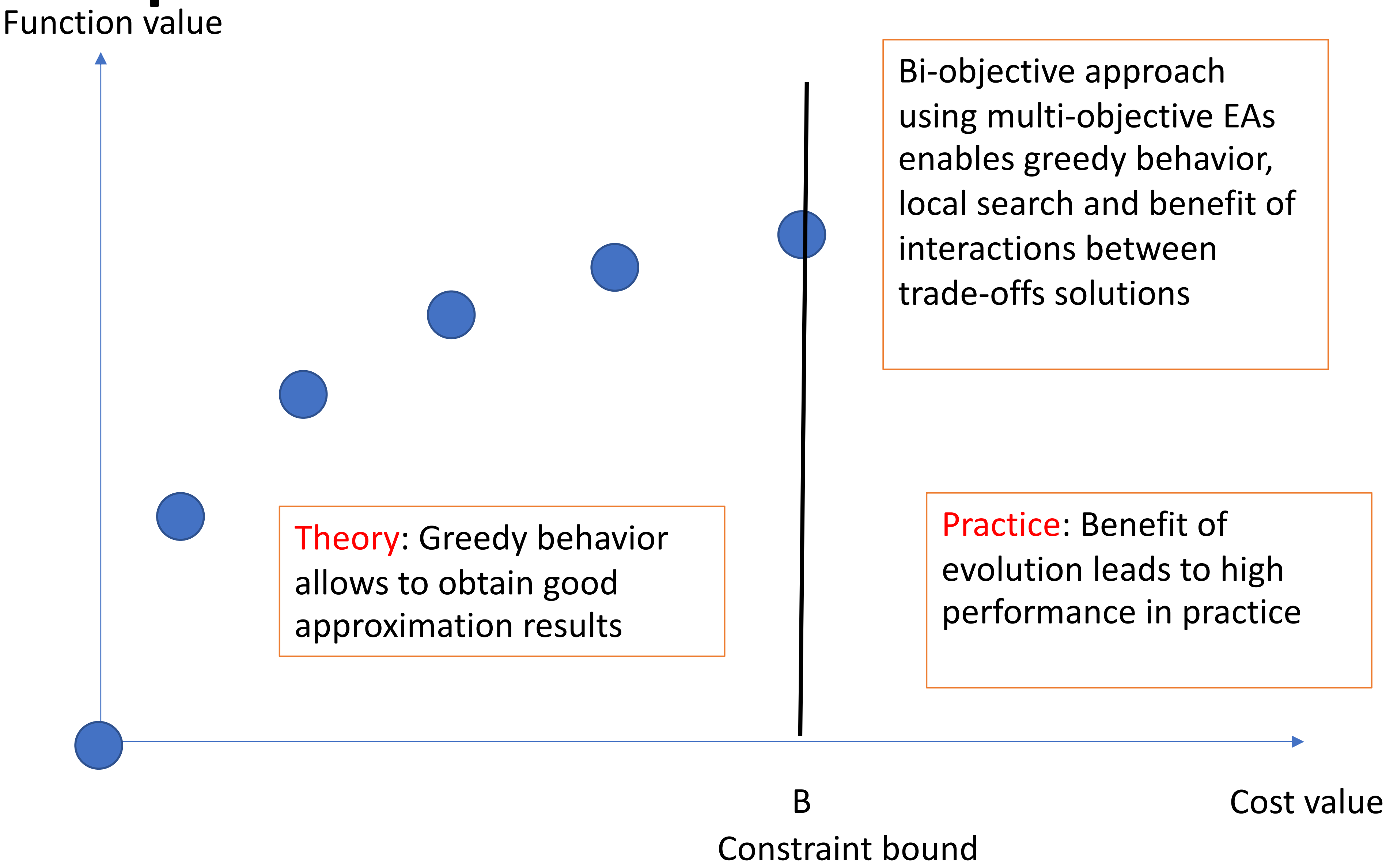
Chebyshev:

Lemma 2. *Let* X *be a solution with expected weight* $E[W(X)]$ *and variance* $\mathrm{Var}[W(X)]$. *If*

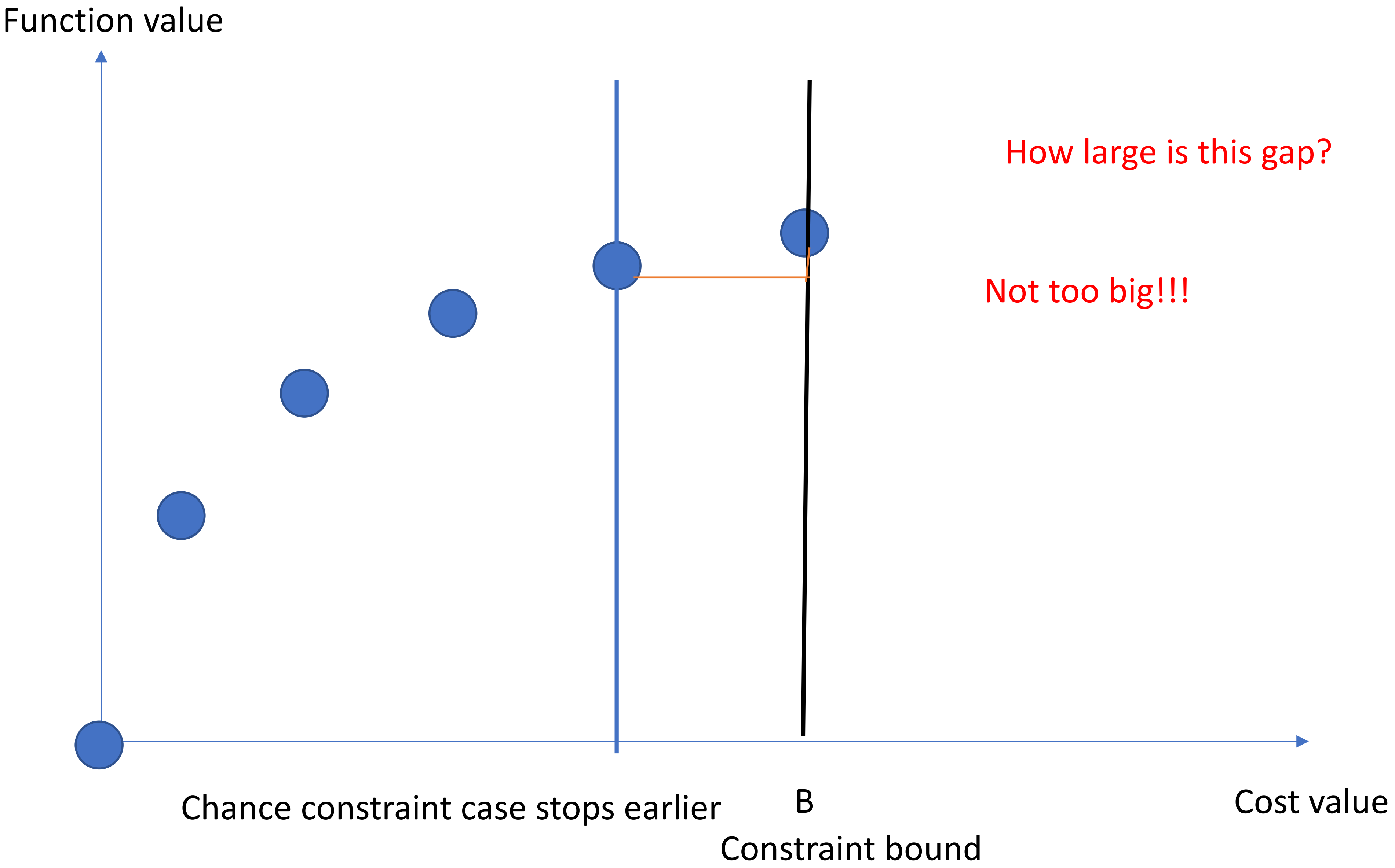
$$B - E[W(X)] \geq \sqrt{\frac{(1-\alpha)\,\mathrm{Var}[W(X)]}{\alpha}}$$

then $\Pr[W(X) > B] \leq \alpha$.

Bi-objective approach / Pareto Optimisation



Bi-objective approach / Pareto Optimisation



Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-Objective Algorithms

- We consider the performance of the Global Simple Evolutionary Multi-Objective Optimizer (GSEMO) and Non-dominated Sorting Genetic Algorithm (NSGA-II) for the optimisation of stochastic constrained submodular functions.

Goal: Bi-objective formulations of constrained submodular optimisation problems in terms of Pareto optimisation enable evolutionary algorithms to achieve:

- best theoretical performance guarantees and
- state-of-the-art practical results
- for a wide range of submodular optimisation problems.

Aneta Neumann, Frank Neumann: Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-objective Algorithms. PPSN (1) 2020

Submodular functions with Chance Constraints (stochastic settings)

We consider the optimization of a monotone submodular function f subject to a chance constraint where each element $s \in V$ takes on a random weight W(s).

We examine constraints of the type

$$\Pr[W(S) > C] \leq \alpha.$$

where $W(S) = \sum_{s \in S} w(s)$ is the sum of the random weights of the elements and C is the given constraint bound. The parameter α specifies the probability of exceeding the bound C that can be tolerated for a feasible solution S.

Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-Objective Algorithms

The multi-objective fitness function: $g(X) = (g_1(X), g_2(X))$

- g_1 measures the tightness in terms of the constraint
- g_2 measures the quality of X in terms of the given submodular function f.

constraint function

$$g_1(X) = \begin{cases} E_W(X) - C & \text{if} \quad (C - E_W(X))/(\delta \cdot |X|) \geq 1 \\ \hat{P}r(W(X) > C) & \text{if } (E_W(X) < C) \wedge (C - E_W(X))/(\delta|X| < 1) \\ 1 + (E_W(X) - C) & \text{if} \quad E_W(X) \geq C \end{cases}$$

always feasible

infeasible

objective function

$$g_2(X) = \begin{cases} f(X) \text{ if} & g_1(X) \leq \alpha \\ -1 \quad \text{if} & \text{otherwise} \end{cases}$$

Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-Objective Algorithms

- The case of uniform identically distributed (IID) weights.

Theorem 1. *Let $k = \min\{n+1, \lfloor C/a \rfloor\}$ and assume $\lfloor C/a \rfloor = \omega(1)$. Then the expected time until GSEMO has computed a $(1-o(1))(1-1/e)$-approximation for a given monotone submodular function under a chance constraint with uniform iid weights is $O(nk(k+\log n))$.*

Main Results:

- GSEMO using a multi-objective formulation of the problem based on tail inequalities is able to achieve the same approximation guarantee as recently studied greedy approaches.

Optimising Monotone Chance-Constrained Submodular Functions Using Evolutionary Multi-Objective Algorithms

Results for Influence Maximization and Maximum Coverage with uniform chance constraints, respectively.

C	α	δ	GA (1)	GSEMO (2)					NSGA-II (3)				
				mean	min	max	std	stat	mean	min	max	std	stat
20	0.1	0.5	51.51	**55.75**	54.44	56.85	0.5571	$1^{(+)}$	55.66	54.06	56.47	0.5661	$1^{(+)}$
	0.1	1.0	46.80	**50.65**	49.53	51.68	0.5704	$1^{(+)}$	50.54	49.61	52.01	0.6494	$1^{(+)}$
50	0.1	0.5	90.55	**94.54**	93.41	95.61	0.5390	$1^{(+)}, 3^{(+)}$	92.90	90.75	94.82	1.0445	$1^{(+)}, 2^{(-)}$
	0.1	1.0	85.71	**88.63**	86.66	90.68	0.9010	$1^{(+)}, 3^{(+)}$	86.89	85.79	88.83	0.8479	$1^{(+)}, 2^{(-)}$
100	0.1	0.5	144.16	**147.28**	145.94	149.33	0.8830	$1^{(+)}, 3^{(+)}$	144.17	142.37	146.18	0.9902	$2^{(-)}$
	0.1	1.0	135.61	**140.02**	138.65	142.52	0.7362	$1^{(+)}, 3^{(+)}$	136.58	134.80	138.21	0.9813	$2^{(-)}$
20	0.001	0.5	48.19	**50.64**	49.10	51.74	0.6765	$1^{(+)}$	50.33	49.16	51.25	0.5762	$1^{(+)}$
	0.001	1.0	39.50	**44.53**	43.63	45.55	0.4687	$1^{(+)}$	44.06	42.18	45.39	0.7846	$1^{(+)}$
50	0.001	0.5	75.71	**80.65**	78.92	82.19	0.7731	$1^{(+)}$	80.58	79.29	81.63	0.6167	$1^{(+)}$
	0.001	1.0	64.49	69.79	68.89	71.74	0.6063	$1^{(+)}$	**69.96**	68.90	71.05	0.6192	$1^{(+)}$
100	0.001	0.5	116.05	**130.19**	128.59	131.51	0.7389	$1^{(+)}, 3^{(+)}$	127.50	125.38	129.74	0.9257	$1^{(+)}, 2^{(-)}$
	0.001	1.0	96.18	**108.95**	107.26	109.93	0.6466	$1^{(+)}, 3^{(+)}$	107.91	106.67	110.17	0.7928	$1^{(+)}, 2^{(-)}$

C	α	δ	GA (1)	GSEMO (2)					NSGA-II (3)				
				Mean	Min	Max	Std	Stat	Mean	Min	Max	Std	Stat
10	0.1	0.5	448.00	**458.80**	451.00	461.00	3.3156	$1^{(+)}$	457.97	449.00	461.00	4.1480	$1^{(+)}$
	0.1	1.0	376.00	**383.33**	379.00	384.00	1.7555	$1^{(+)}$	382.90	379.00	384.00	2.0060	$1^{(+)}$
15	0.1	0.5	559.00	**559.33**	555.00	562.00	2.0057	$3^{(+)}$	557.23	551.00	561.00	2.4309	$1^{(-)}, 2^{(-)}$
	0.1	1.0	503.00	**507.80**	503.00	509.00	1.1567	$1^{(+)}$	507.23	502.00	509.00	1.8323	$1^{(+)}$
20	0.1	0.5	587.00	**587.20**	585.00	589.00	1.2149	$3^{(+)}$	583.90	580.00	588.00	1.9360	$1^{(-)}, 2^{(-)}$
	0.1	1.0	569.00	**569.13**	566.00	572.00	1.4559	$3^{(+)}$	565.30	560.00	569.00	2.1520	$1^{(-)}, 2^{(-)}$
10	0.001	0.5	413.00	**423.67**	418.00	425.00	1.8815	$1^{(+)}$	422.27	416.00	425.00	2.6121	$1^{(+)}$
	0.001	1.0	376.00	**383.70**	379.00	384.00	1.1492	$1^{(+)}$	381.73	377.00	384.00	2.6514	$1^{(+)}$
15	0.001	0.5	526.00	**527.97**	525.00	532.00	2.1573	$1^{(+)}$	527.30	520.00	532.00	2.7436	
	0.001	1.0	448.00	**458.87**	453.00	461.00	2.9564	$1^{(+)}$	457.10	449.00	461.00	4.1469	$1^{(+)}$
20	0.001	0.5	568.00	**568.87**	565.00	572.00	1.5025	$3^{(+)}$	564.60	560.00	570.00	2.7618	$1^{(-)}, 2^{(-)}$
	0.001	1.0	526.00	**528.03**	525.00	530.00	1.8843	$1^{(+)}$	527.07	522.00	530.00	2.2427	

Main Results:

- Experimental results show that GSEMO computes significantly better solutions than the greedy approach and often outperforms NSGA-II.

References

- Aneta Neumann, Yue Xie, Frank Neumann: Evolutionary Algorithms for Limiting the Effect of Uncertainty for the Knapsack Problem with Stochastic Profits. In: Parallel Problem Solving from Nature, PPSN 2022, Proceedings, Part I, Lecture Notes in Computer Science, pp. 294—307.
- Aneta Neumann, Frank Neumann: Optimising Chance-Constrained Submodular Functions Using Evolutionary Multi-Objective Algorithms. In: Parallel Problem Solving from Nature, PPSN 2020, Proceedings, Part I, Lecture Notes in Computer Science, pp. 404—417.
- Benjamin Doerr, Carola Doerr, Aneta Neumann, Frank Neumann, Andrew M. Sutton: Optimization of Chance-Constrained Submodular Functions. In: AAAI Conference on Artificial Intelligence, AAAI 2020, pp. 1460—1467.
- Yue Xie, Oscar Harper, Hirad Assimi, Aneta Neumann, Frank Neumann: Evolutionary algorithms for the chance-constrained knapsack problem. In GECCO 2019, pp. *338-346.*
- Hirad Assimi, Oscar Harper, Yue Xie, Aneta Neumann, Frank Neumann: Evolutionary Bi-Objective Optimization for the Dynamic Chance-Constrained Knapsack Problem Based on Tail Bound Objectives. In. European Conference on Artificial Intelligence, ECAI 2020, pp. 307—314.
- Yue Xie, Aneta Neumann, Frank Neumann: Specific single- and multi-objective evolutionary algorithms for the chance-constrained knapsack problem. In GECCO 2020, pp. 271—279.

3D Pareto Optimisation for Problems with Chance Constraints

Chance constrained problem

Given n elements with stochastic weight where each weight w_i
is chosen independently of the others according to a Normal distribution

$N(\mu_i, \sigma_i^2)$, $1 \leq i \leq n$, where $\mu_i \geq 1$ and $\sigma_i \geq 1$, $1 \leq i \leq n$.

Our goal is to

$$\min W \quad \text{subject to} \quad (Pr(w(x) \leq W) \geq \alpha) \wedge (|x|_1 \geq k) \tag{1}$$

where $w(x) = \sum_{i=1}^{n} w_i x_i$, $x \in \{0,1\}^n$, and $\alpha \in [1/2, 1[$.

Problem is equivalent to minimizing

$$\hat{w}(x) = \mu(x) + K_\alpha \sqrt{v(x)}, \quad \text{subject to} \quad (|x|_1 \geq k)$$

K_α denotes the α-fractional point of the standard Normal distribution

2- and 3-objective Models

min ↓ ↓ min ↓ ↓ max ↑

$$f_{2D}(x) = (\hat{\mu}(x), \hat{v}(x))$$

where

$$\hat{\mu}(x) = \begin{cases} \sum_{i=1}^{n} \mu_i x_i & |x|_1 \geq k \\ (k - |x|_1) \cdot (1 + \sum_{i=1}^{n} \mu_i) & |x|_1 < k \end{cases}$$

$$\hat{v}(x) = \begin{cases} \sum_{i=1}^{n} \sigma_i^2 x_i & |x|_1 \geq k \\ (k - |x|_1) \cdot (1 + \sum_{i=1}^{n} \sigma_i^2) & |x|_1 < k \end{cases}$$

Neumann/Witt IJCAI 2022

$$f_{3D}(x) = (\mu(x), v(x), c(x))$$

Pareto optimization turns constraint c(x) into third objective. Expected weight and variance no longer need penalty terms.

Consequence:
Additional search direction, but also many more trade-off objective vectors.

Using the 3-objective model, we can also solve:

$$\max c(x) \quad \text{subject to} \quad Pr(w(x) \leq B) \geq \alpha. \tag{3}$$

by returning for any given B and α the solution of maximal c(x)-value in the final population that meets the constraint.

GSEMO

Algorithm 1: GSEMO

Choose $x \in \{0,1\}^n$ uniformly at random;
$P \leftarrow \{x\}$;
repeat
  Choose $x \in P$ uniformly at random;
  Create y by flipping each bit x_i of x with probability $\frac{1}{n}$;
  if $\nexists\, w \in P : w \prec y$ **then**
    $P \leftarrow (P \setminus \{z \in P \mid y \preceq z\}) \cup \{y\}$;
until $stop$;

Apply GSEMO to 2-objective and 3-objective model

SEMO differs from GSEMO flipping in each iteration exactly one or two bits (2D case) or 1-bit (3D case).

Theoretical Investigations

Analyze GSEMO with respect to the time until it has produced for each k and $\alpha \geq \frac{1}{2}$ an optimal solution for the case $c(x) = |x|_1$.

Runtime Analysis: Measure time by the (expected) number of fitness evaluations to reach the goal.

Theoretical Investigations

Key argument: When considering $\boldsymbol{c(x) = |x|_1}$. Then there exists for each α ≥ $\frac{\mathbf{1}}{\mathbf{2}}$ a convex combination of $\boldsymbol{\mu(x)}$ and $\boldsymbol{v(x)}$, i.e.

$$\boldsymbol{f_{\lambda}(x) = \lambda\mu(x) + (1-\lambda)v(x), \lambda \in [0,1]}$$

such that if $\boldsymbol{x}^*$ is minimal for $\boldsymbol{f_{\lambda}(x)}$ under the constraint $\boldsymbol{c(x) \geq k}$ then $\boldsymbol{x}^*$ is minimal for $\hat{w}(x) = \mu(x) + K_\alpha\sqrt{v(x)}$, and $\boldsymbol{c(x) \geq k}$.

Implies: Set of desired solutions can be obtained by computing the extremal corner points of the Pareto front.

3D Analysis based on 2-bit flips

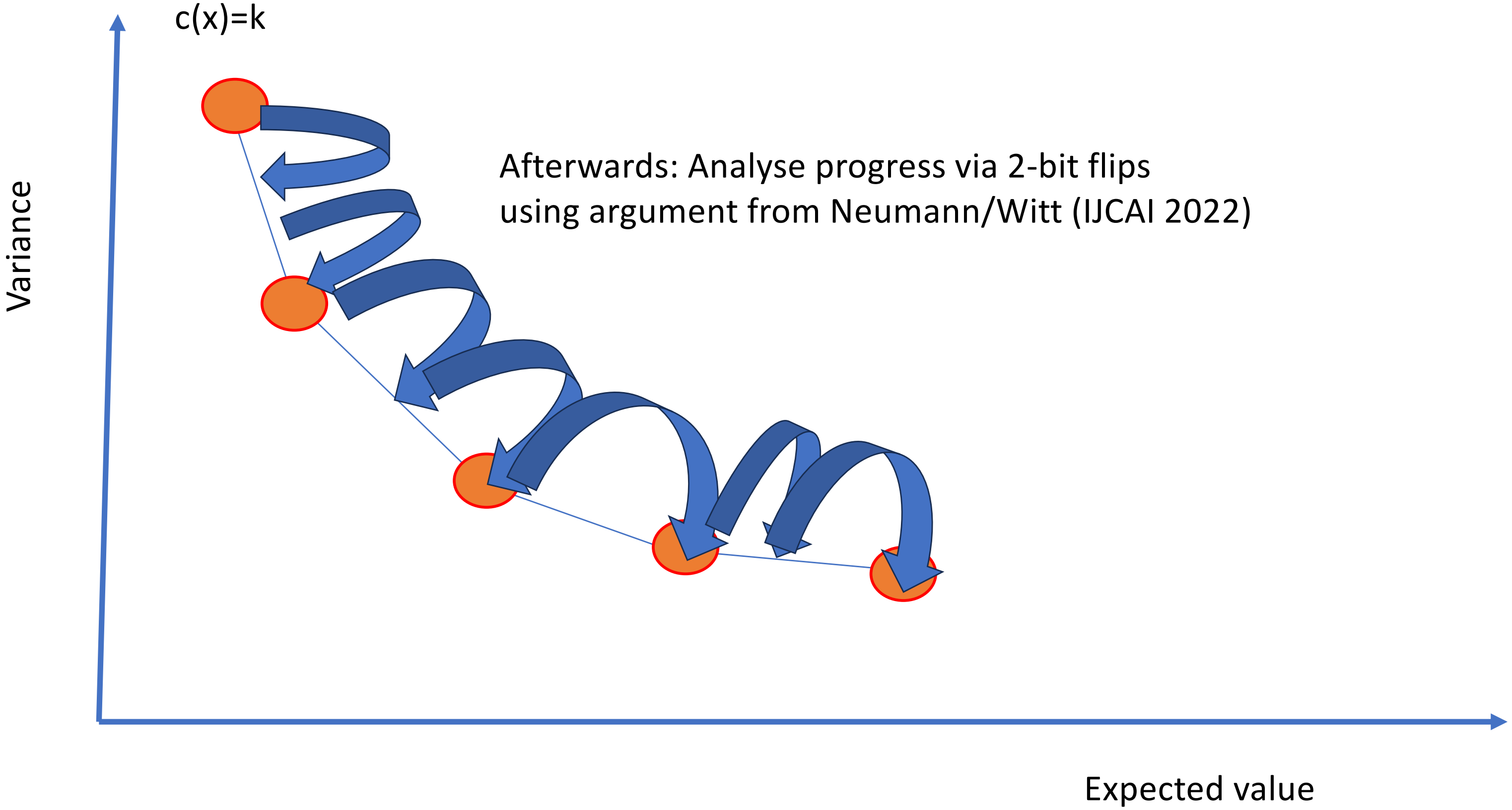
Theorem 2. *The expected time until GSEMO has computed a population which includes an optimal solution for the problem given in Equation 3 with $c(x) = |x|_1$ for any possible choice of B and $\alpha \in [1/2, 1[$ is $O(P_{\max} n^4 (\log n + \log v_{\max}))$,*

Theorem 1. *GSEMO computes a population ... $k \in \{0, \ldots\}$ an optimal ...*

$$x_\alpha^k = \arg\min_{x \in X^k} \left\{ \mu(x) + K_\alpha \sqrt{v(x)} \right\} \quad (4)$$

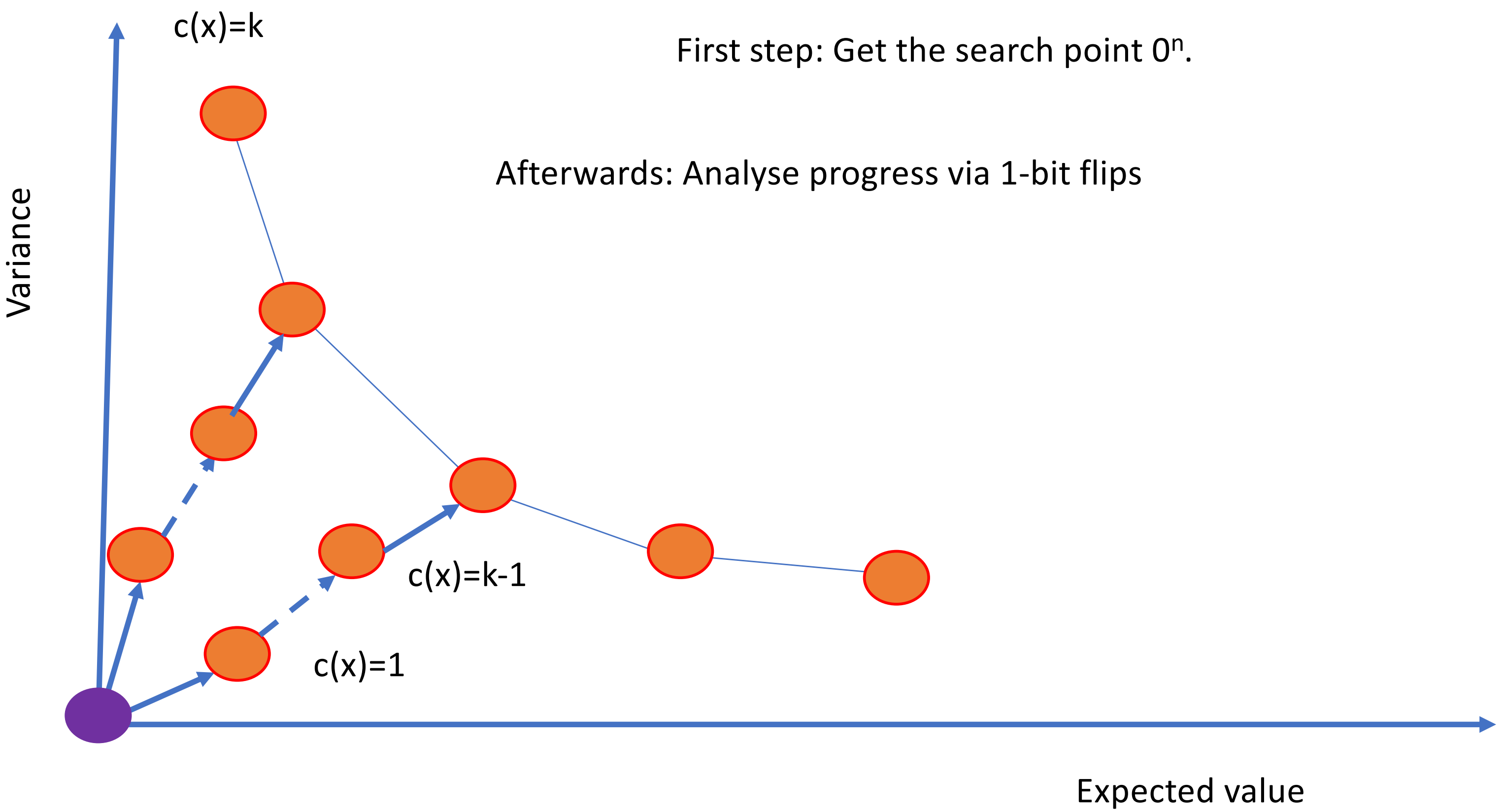
in expected time $O(P_{\max} n^4 (\log n + \log v_{\max}))$.

First step: Analyse time to get for each k
a solution with k elements and minimal expected value



3D Analysis based on 1-bit flips

Theorem 3. *The expected time until SEMO3D and GSEMO have computed a population which includes an optimal solution for the problems given in Equation 1 (for any choice of k and α) and Equation 3 (with $c(x) = |x|_1$ for any choice of B and α) is $O(P_{\max} n^2)$ and it is at most $2eP_{\max} n^2$ with probability $1 - e^{-\Omega(n)}$.*



Experiments

We consider a stochastic version of the minimum dominating set problem in a given undirected graph

G=(V, E) with stochastic weights on the nodes.

A node dominates itself and all of its neighbors.

c(x) denotes the number of nodes dominated in the given search point x. Constraint c(x)=n.

Weight of each node $\boldsymbol{v_i}$ is chosen independently of the others according to a Normal distribution $N\left(\mu_i, \sigma_i^2\right)$.

10M fitness evaluations, 30 independent runs of each instance.

Uniform, Uniform-fixed, degree based weights.

Different values of $\boldsymbol{\alpha = 1 - \beta}$.

Maximum Population Size during runs

Graph	weight gype	SEMO2D		SEMO3D		GSEMO2D		GSEMO3D	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
cfat200-1	uniform	57	19	2921	964	56	16	2923	929
cfat200-2	uniform	29	11	348	128	23	10	361	128
ca-netscience	uniform	69	22	5531	997	40	11	4631	678
ca-GrQc	uniform	4	3	7519	488	3	1	3921	262
Erdos992	uniform	2	1	4476	338	1	1	2173	153
cfat200-1	uniform-fixed	1	0	66	12	1	0	67	11
cfat200-2	uniform-fixed	1	0	18	4	1	0	18	4
ca-netscience	uniform-fixed	1	0	401	43	1	0	403	40
ca-GrQc	uniform-fixed	1	0	3565	251	1	0	1942	133
Erdos992	uniform-fixed	1	0	2217	102	1	0	1313	61
cfat200-1	degree	2	1	335	36	2	1	340	35
cfat200-2	degree	1	1	42	6	1	0	42	6
ca-netscience	degree	22	8	3293	764	18	7	2981	585
ca-GrQc	degree	3	2	6112	371	3	2	3240	252
Erdos992	degree	2	1	3128	166	2	1	1725	84

Table 1: Maximum population size for stochastic minimum weight dominating set.

Experimental Results (Uniform random)

Graph	weight gype	β	SEMO2D Mean	SEMO2D Std	SEMO3D Mean	SEMO3D Std	p-value	GSEMO2D Mean	GSEMO2D Std	GSEMO3D Mean	GSEMO3D Std	p-value
cfat200-1	uniform	0.2	3618	76	**3599**	82	0.308	3615	91	**3599**	79	0.544
		0.1	3994	82	**3970**	82	0.268	3989	96	**3972**	80	0.544
		0.01	4877	101	**4842**	86	0.169	4866	109	**4845**	86	0.535
		1.0E-4	6030	123	**5985**	94	0.128	6015	126	**5991**	98	0.455
		1.0E-6	6870	139	**6824**	103	0.201	6855	138	**6832**	108	0.605
		1.0E-8	7562	152	**7519**	113	0.326	7546	147	**7525**	118	0.641
		1.0E-10	8163	163	**8122**	123	0.469	8145	154	**8125**	125	0.751
		1.0E-12	8700	172	**8660**	130	0.535	8680	159	**8660**	130	0.859
		1.0E-14	9190	180	**9150**	136	0.657	9169	164	**9148**	133	0.842
		1.0E-16	9633	188	**9593**	142	0.636	9611	168	**9589**	137	0.865
cfat200-2	uniform	0.2	1797	72	**1788**	53	0.923	1791	49	**1767**	32	0.049
		0.1	2049	78	**2035**	55	0.865	2040	54	**2016**	37	0.074
		0.01	2634	92	**2617**	69	0.739	2621	72	**2593**	51	0.162
		1.0E-4	3394	111	**3369**	85	0.535	3381	97	**3336**	65	0.070
		1.0E-6	3948	125	**3918**	95	0.511	3937	113	**3880**	71	0.044
		1.0E-8	4403	134	**4372**	106	0.496	4394	124	**4329**	77	0.032
		1.0E-10	4799	143	**4768**	117	0.549	4793	132	**4720**	82	0.028
		1.0E-12	5153	150	**5123**	125	0.559	5149	139	**5071**	85	0.024
		1.0E-14	5476	157	**5447**	133	0.589	5475	145	**5391**	88	0.020
		1.0E-16	5769	164	**5740**	141	0.559	5769	150	**5681**	91	0.021
ca-netscience	uniform	0.2	32922	1308	**32608**	904	0.506	33042	1289	**33007**	1023	0.712
		0.1	34456	1323	**34115**	907	0.544	34568	1302	**34514**	1028	0.745
		0.01	38097	1361	**37694**	919	0.408	38189	1334	**38089**	1040	0.848
		1.0E-4	42938	1414	**42461**	938	0.274	43012	1380	**42846**	1054	1.000
		1.0E-6	46527	1457	**45995**	951	0.255	46591	1415	**46377**	1065	0.824
		1.0E-8	49500	1493	**48923**	960	0.198	49557	1442	**49303**	1076	0.712
		1.0E-10	52091	1526	**51478**	970	0.165	52145	1465	**51857**	1087	0.615
		1.0E-12	54416	1554	**53773**	979	0.147	54467	1487	**54150**	1096	0.564
		1.0E-14	56542	1581	**55873**	987	0.132	56592	1507	**56249**	1105	0.487
		1.0E-16	58469	1605	**57776**	996	0.117	58517	1526	**58151**	1114	0.478

Experimental Results (Uniform fixed variance)

Graph	weight gype	β	SEMO2D		SEMO3D			GSEMO2D		GSEMO3D		
			Mean	Std	Mean	Std	p-value	Mean	Std	Mean	Std	p-value
cfat200-1	uniform-fixed	0.2	3891	183	**3851**	129	0.530	3813	125	**3721**	59	0.006
		0.1	4353	195	**4306**	135	0.464	4269	133	**4169**	59	0.006
		0.01	5450	224	**5385**	149	0.333	5352	151	**5235**	59	0.006
		1.0E-4	6913	264	**6823**	169	0.258	6795	177	**6655**	59	0.006
		1.0E-6	7999	293	**7890**	183	0.234	7868	196	**7710**	59	0.006
		1.0E-8	8901	317	**8776**	195	0.223	8757	212	**8586**	59	0.006
		1.0E-10	9688	339	**9549**	206	0.217	9534	226	**9350**	59	0.006
		1.0E-12	10395	358	**10243**	216	0.206	10232	239	**10036**	59	0.006
		1.0E-14	11043	376	**10878**	225	0.191	10871	250	**10665**	59	0.006
		1.0E-16	11630	392	**11455**	234	0.186	11450	261	**11235**	59	0.006
cfat200-2	random-fixed	0.2	1989	116	**1980**	112	0.690	1937	104	**1866**	50	0.011
		0.1	2307	128	**2297**	123	0.679	2249	115	**2171**	54	0.011
		0.01	3064	157	**3048**	149	0.554	2990	141	**2897**	63	0.011
		1.0E-4	4073	196	**4049**	185	0.554	3978	176	**3864**	76	0.011
		1.0E-6	4822	225	**4792**	213	0.554	4712	203	**4583**	86	0.011
		1.0E-8	5444	249	**5410**	236	0.554	5321	225	**5179**	94	0.011
		1.0E-10	5986	269	**5948**	257	0.554	5853	244	**5700**	102	0.011
		1.0E-12	6474	288	**6432**	275	0.554	6330	261	**6168**	108	0.011
		1.0E-14	6920	305	**6875**	292	0.554	6768	277	**6596**	114	0.011
		1.0E-16	7325	321	**7276**	308	0.554	7164	291	**6984**	120	0.011
ca-netscience	uniform-fixed	0.2	35378	1891	**32956**	844	0.000	34936	1747	**32926**	816	0.000
		0.1	37239	1934	**34718**	844	0.000	36779	1785	**34687**	819	0.000
		0.01	41659	2038	**38901**	844	0.000	41156	1878	**38869**	825	0.000
		1.0E-4	47551	2178	**44475**	844	0.000	46991	2004	**44439**	831	0.000
		1.0E-6	51927	2282	**48614**	843	0.000	51325	2098	**48576**	835	0.000
		1.0E-8	55559	2369	**52049**	842	0.000	54922	2177	**52009**	838	0.000
		1.0E-10	58729	2445	**55048**	842	0.000	58061	2246	**55006**	841	0.000
		1.0E-12	61577	2513	**57741**	843	0.000	60881	2309	**57698**	844	0.000
		1.0E-14	64184	2576	**60207**	843	0.000	63463	2366	**60162**	847	0.000
		1.0E-16	66548	2633	**62443**	844	0.000	65805	2418	**62397**	850	0.000

Experimental Results (degree-based)

Graph	weight gype	β	SEMO2D		SEMO3D			GSEMO2D		GSEMO3D		
			Mean	Std	Mean	Std	p-value	Mean	Std	Mean	Std	p-value
cfat200-1	degree	0.2	4495	143	**4392**	10	0.002	4444	115	**4387**	6	0.001
		0.1	4835	148	**4727**	14	0.002	4781	119	**4721**	9	0.003
		0.01	5642	158	**5523**	25	0.001	5582	129	**5512**	16	0.004
		1.0E-4	6718	172	**6584**	39	0.001	6650	143	**6566**	26	0.003
		1.0E-6	7517	184	**7372**	50	0.001	7443	154	**7349**	34	0.003
		1.0E-8	8180	193	**8025**	59	0.001	8101	163	**7999**	40	0.003
		1.0E-10	8758	202	**8596**	67	0.001	8675	171	**8567**	45	0.003
		1.0E-12	9278	210	**9108**	74	0.001	9191	178	**9076**	50	0.003
		1.0E-14	9754	217	**9578**	81	0.001	9663	185	**9542**	55	0.003
		1.0E-16	10185	223	**10003**	87	0.001	10091	191	**9965**	59	0.003
cfat200-2	degree	0.2	3218	227	**3029**	154	0.033	3041	172	**2963**	4	0.027
		0.1	3448	235	**3256**	160	0.033	3267	178	**3185**	6	0.027
		0.01	3996	255	**3795**	173	0.033	3803	194	**3713**	11	0.027
		1.0E-4	4726	280	**4514**	193	0.033	4518	216	**4416**	17	0.027
		1.0E-6	5268	300	**5048**	209	0.033	5049	232	**4938**	22	0.027
		1.0E-8	5718	316	**5491**	223	0.033	5490	245	**5371**	26	0.027
		1.0E-10	6110	329	**5878**	235	0.033	5875	257	**5749**	30	0.027
		1.0E-12	6463	342	**6225**	247	0.033	6220	267	**6089**	33	0.027
		1.0E-14	6786	354	**6543**	257	0.033	6537	277	**6400**	36	0.027
		1.0E-16	7079	364	**6832**	267	0.033	6823	286	**6682**	38	0.027
ca-netscience	degree	0.2	28587	1535	**26148**	201	0.000	28164	1002	**26169**	196	0.000
		0.1	30122	1580	**27636**	207	0.000	29689	1029	**27657**	200	0.000
		0.01	33758	1686	**31158**	228	0.000	33300	1098	**31183**	216	0.000
		1.0E-4	38593	1828	**35840**	269	0.000	38103	1192	**35874**	251	0.000
		1.0E-6	42180	1936	**39313**	306	0.000	41665	1265	**39355**	285	0.000
		1.0E-8	45155	2026	**42192**	338	0.000	44620	1327	**42243**	317	0.000
		1.0E-10	47751	2104	**44705**	367	0.000	47198	1381	**44763**	347	0.000
		1.0E-12	50082	2175	**46962**	394	0.000	49514	1429	**47026**	374	0.000
		1.0E-14	52216	2239	**49027**	420	0.000	51633	1474	**49098**	400	0.000
		1.0E-16	54151	2297	**50900**	444	0.000	53555	1515	**50977**	423	0.000

“Many”-objective optimization + robustness

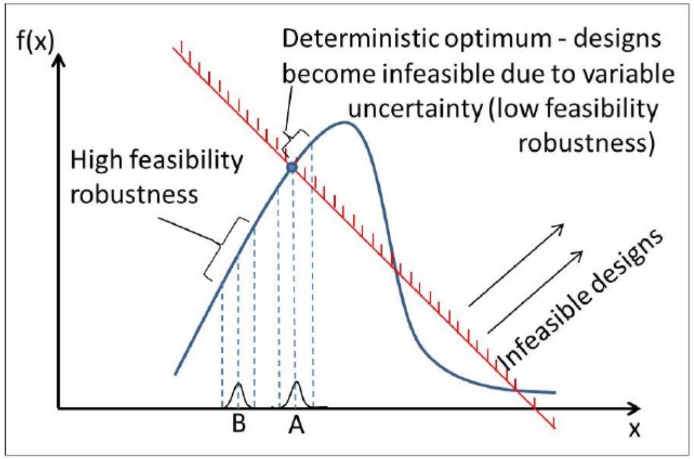
Introduction

- Colloquially, “many-objective” optimization problems (MaOP) is a term used in evolutionary computation community to denote a subset of multi-objective problems (MOP) with more than 3 objectives
- The reason for this differentiation is that increasing number of objectives, especially beyond 3, bring additional challenges in search, visualization and decision-making
- The research in MaOPs has proliferated in the last decade, but limited for the most part to the deterministic optimization scenarios
- In this part of the tutorial, we look into some works that use MaOP formulations while considering stochasticity

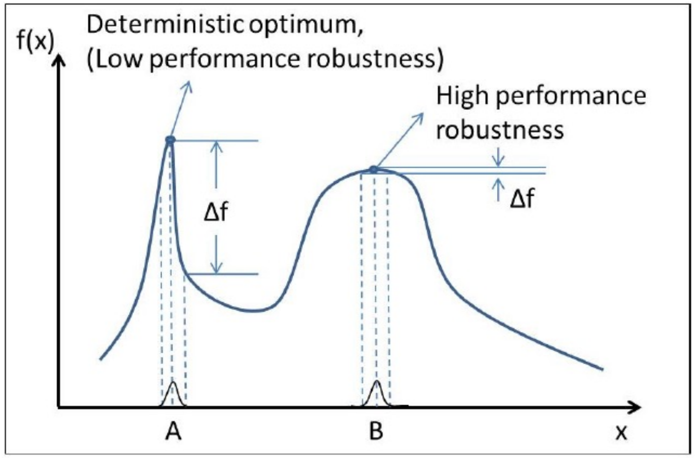
Robustness – different types

Feasibility robustness: Robustness with respect to constraint violation. (also referred to as Reliability). *It measures how likely a design is to violate certain constraints.*

Performance robustness: Robustness with respect to the given objective value. *It measures how likely a design is to deliver its performance.*



(a) Feasibility robustness



(b) Performance robustness

Types of robustness. Design 'B' is more robust in both cases, while 'A' has better objective value (f(x) is being maximized)

Robustness – different types

- Robustness is primarily incorporated in the problem through re-formulation by:
  - Additional or modified the objectives
  - Additional or modified constraints

- If robustness is specified as an objective, there is an opportunity to view trade-offs between the design performance and robustness

- If *both* feasibility and performance robustness are considered as additional objectives, the problem tends to become MaOP. Solving such a formulation might provide more options for studying trade-offs, but is more challenging than solving MOPs.

- Therefore many of the studies are inclined towards limiting the objectives, and using constraints in the re-formulation.

- Given the advancements in the field of deterministic MaOPs in the last decade, it is worthwhile considering formulations that use additional objectives for robust optimization

Common (re-)formulations

Reference work	Robust formulation	Type of robustness considered	Robustness as an additional objective?	Quantification of robustness	Ability to generate robust solutions with varying levels of robustness in a single run?	Optimization method	Type of problems studied
Deb and Gupta [35]	Min.$[f_1, f_2 \dots, f_M]$ Subject to $\|f^p(x) - f(x)\|/\|f(x)\| \leq \eta$ Or Min.$[f_1^{\text{eff}}, f_2^{\text{eff}} \dots, f_M^{\text{eff}}]$	Performance robustness (as additional constraint)	No	Expected measure	No	Nondominated sorting algorithm-II(NSGA-II)	(MO, C)
Jin and Sendhoff [8]	Min.$[f_1 = f, f_2 = \sigma_f]$ Subject to $g_j \leq 0$	Performance robustness (as additional objective)	Yes	Variance measure	Yes, with respect to performance	ES	(SO, C)
Chen et al. [3]	Min.$[(\mu_f, \sigma_f^2)]$ Subject to $E[g_j(x, z)] + n\sigma_{g_j} \leq 0$	Performance and feasibility robustness	Yes	Expected and variance measures	Yes, with respect to performance	Adaptive linear programing	(MO, C)
G. Sun et al. [4]	Min.$[(\mu_{f_1}, \sigma_{f_1}), (\mu_{f_2}, \sigma_{f_2}), \dots,$ $(\mu_{f_M}, \sigma_{f_M})]$ Subject to $\mu_{g_j} + n\sigma_{g_j} \leq 0$	Performance and feasibility robustness	Yes	Sigma level based measure	Yes, with respect to performance	PSO	(MO, C)
Sundaresan et al. [5]	Min.$E\,[f]$ Subject to $E[g_j] \leq 0, E[h_j] = 0$	Feasibility robustness	No	Expected measure	No	Statistical optimization technique/design of experiments	(SO, C)
Wang et al. [11]	Min.$[\mu_f + k\sigma_f]$ Subject to $\mu_{g_j} + n\sigma_{g_j} \leq 0$	Performance and feasibility robustness	No	Aggregation of expected and variance measure	No	Sequential optimization and reliability assessment	(SO, C)
Mourelatos and Liang [65]	Min.$[(\mu_f, \Delta R_\sigma = \sigma_{R2} - \sigma_{R1})]$ Subject to Prob $\{g_j \leq 0\} \geq \alpha_i, i = 1, 2, \dots, M$	Performance and feasibility robustness	Yes	Expected and variance measures	Yes, with respect to performance	Single-loop reliability based design optimization RBDO	(MO, C)
Youn et al. [89]	Min.$[(\mu_H - h_t/\mu_{H_0} - h_t)^2 + (\sigma_H/\sigma_{H_0})^2]or[(\text{sgn}(\mu_H))(\mu_H/\mu_{H_0})^2 + (\sigma_H/\sigma_{H_0})^2]or[(\text{sgn}(\mu_H))(\mu_1/H/\mu_1/H_0)^2 + (\sigma_1/H/\sigma_2/H_0)^2]$ Subject to Prob $\{g_j \leq 0\} \geq \alpha_i, i = 1, 2, \dots, M$	Performance and feasibility robustness	No	Expected and variance measures	No	Enhanced hybrid mean value (HMV+)	(MO, C)
Du et al. [26]	Min.$[(\mu_{g_{\text{obj}}}, \Delta g^{\alpha_2}_{\text{obj}_\alpha 1})]$ Subject to $[g_i^\alpha \leq 0, i = 1, 2, \dots m]$	Performance and feasibility robustness	No	Percentile performance	No	MPP of inverse reliability (MPPIR) search method	(SO, C)
Gunawan and Azarm [66]	Min.$\mathbf{f}(\mathbf{x}, \mathbf{p}) = [f_1, f_2, \dots, f_M]$ Subject to $g_j(\mathbf{x}, \mathbf{p}) \leq 0, j = 1, 2, \dots J 1 - (\eta_f, \eta_g) \leq 0$	Performance and feasibility robustness	No	Worst Case sensitivity region (WCSR)	No	NSGA	(MO, C)
Asafuddoula et. al. [81]	Four different formulations, Form-1 to Form-4; incorporating FR alone or both FPR	Performance and feasibility robustness	Yes	Sigma-level	Yes, with respect to feasibility and performance	DBEA-r optimization DBEA-r	(MO/MaO, C)

Note: SO: single objective; MO: multi-objective; MaO: many-objective ($\geq$ 4 objectives); C: constrained. For complete details on the robust formulations, please refer to the cited publications.

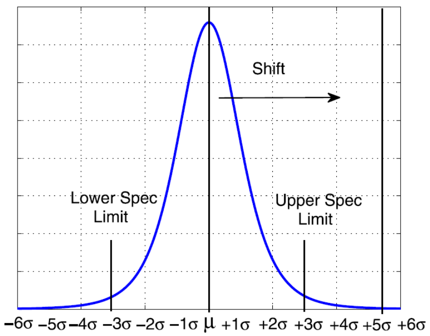
Some existing formulations used for robust optimization [Ray et al, JMD 2015]

Proposed formulation(s)

Robust form	Formulation type	Robust formulation	Robust measure	ATGFSSR[a]
FR	Optimization of expected objective function(s) with FR	$\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}}\ \mu_{f_i(\mathbf{d},\mathbf{x})}, i = 1, 2, \ldots\ldots M$ $\underset{(\mathbf{d},\mathbf{x})}{\text{Maximize}} f_{M+1}(\mathbf{d},\mathbf{x}) = \text{Min}(\text{Sigma}_g, R_c)$ Subject to: $\text{Sigma}_g \equiv \text{Min}(\mu_{g_j(\mathbf{d},\mathbf{x})}/\sigma_{g_j(\mathbf{d},\mathbf{x})}) \geq 0$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)} \quad (5)$	Sigma level based measure (Sigma_g)	Yes
FPR	Optimization of expected objective function(s) with FPR	$\underset{(\mathbf{d},\mathbf{x})}{\text{Minimize}}\ \mu_{f_i(\mathbf{d},\mathbf{x})}, i = 1, 2, \ldots\ldots M$ $\underset{(\mathbf{d},\mathbf{x})}{\text{Maximize}} f_{M+1}(\mathbf{d},\mathbf{x}) = \text{Min}(\text{Sigma}_g, R_c)$ $\underset{(\mathbf{d},\mathbf{x})}{\text{Maximize}} f_{M+2}(\mathbf{d},\mathbf{x}) = \text{Min}(\text{Sigma}_f, R_f)$ Subject to $\text{Sigma}_g \equiv \text{Min}(\mu_{g_j(\mathbf{d},\mathbf{x})}/\sigma_{g_j(\mathbf{d},\mathbf{x})}) \geq 0$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)}$ where $\text{Sigma}_f \equiv \text{Min}(\sigma_{f_{0,i}(\mathbf{d},\mathbf{x})}/\sigma_{f_i(\mathbf{d},\mathbf{x})})$ $\mathbf{x}^{(L)} \leq \mathbf{x} \leq \mathbf{x}^{(U)}, \mathbf{d}^{(L)} \leq \mathbf{d} \leq \mathbf{d}^{(U)} \quad (6)$	Sigma level based measure (Sigma_g, Sigma_f)	Yes

[a]ATGFSSR: Ability to generate tradeoff solutions with various levels of robustness in a single run. The values of R_c and R_f are considered 6 to meet the six sigma quality.

$Sigma_g$ and $Sigma_f$ are robustness measures denoting the number of standard deviations between the mean and spec limit



Sigma Level	Confidence Interval (CI)	Defects per million
$\pm 1\sigma$	68.26	691462
$\pm 2\sigma$	95.46	308538
$\pm 3\sigma$	99.73	66807
$\pm 4\sigma$	99.9937	6210
$\pm 5\sigma$	99.999943	233
$\pm 6\sigma$	99.9999998	2

[Asafuddoula et al, IEEE TEVC 2015]

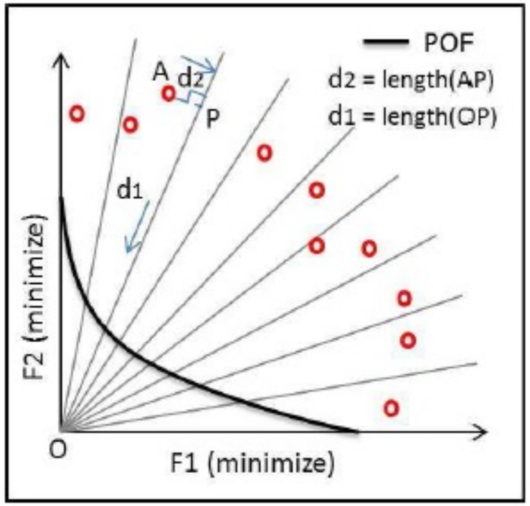
Search method

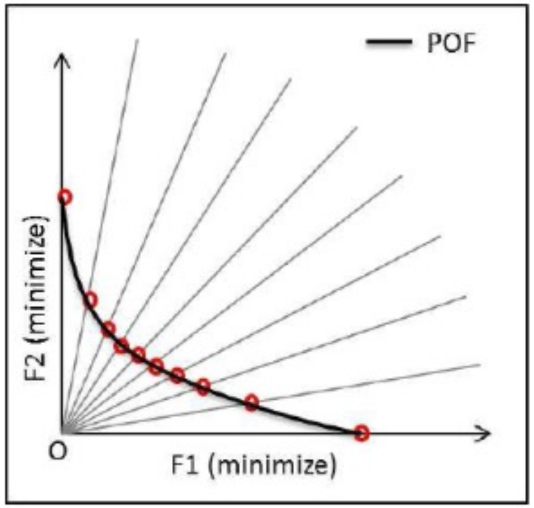
Decomposition based Evolutionary Algorithm (DBEA) for robust optimization

Algorithm 1 DBEA-rg

Input: Gen_{max} (maximum number of generations), W (number of reference points), p_c (probability of crossover), p_m (probability of mutation), η_c (distribution index for crossover), η_m (distribution index for mutation)

$gen = 1$; $CS = \emptyset$;
Generate the reference points using Normal Boundary Intersection (NBI) method.
Initialize the population **P** consisting of W individuals. Randomly assign each individual of **P** to an unique reference direction.
Assign a random binary vector BV of size n to each solution, where n denotes the number of variables of the problem.
Repair the individuals of the population based on its BV and the base design.
Evaluate the initial population using prescribed robust formulation.
Compute the ideal point and the extreme point.
Normalize the individuals of the population
Use *corner-sort* to identify $2M$ corner solutions and assign them to Corner Set (CS).
while ($gen \leq Gen_{max}$) **do**
  Select $I_1 = 1 : W$ as the base parents
  I_2=Generate a shuffled list of individuals in the population
  Create offspring solutions **C** via recombination of I_1 and I_2
  Create offspring BV's via recombination of I_1 and I_2
  Repair the offspring solutions using their BV's and the base design.
  Evaluate the offspring solutions **C**
  Update the corner set CS, ideal and extreme points
  Normalize the individuals of **P** and **C**
  Compute the distances (d_1 and d_2) for all members of **P** in their respective reference directions.
  Compute the distances (d_1 and d_2) for all members of **C** in all reference directions.
  Update the parent solutions in the shuffled order of W with C_l using *single-first encounter strategy* satisfying replacement condition.
  $gen = gen + 1$
end while





(Other methods suitable for MaOPs can also be used)

[Ray et al, JMD 2015]

Results

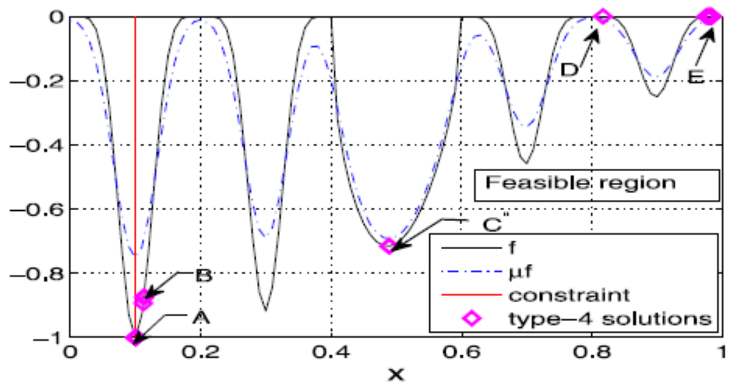
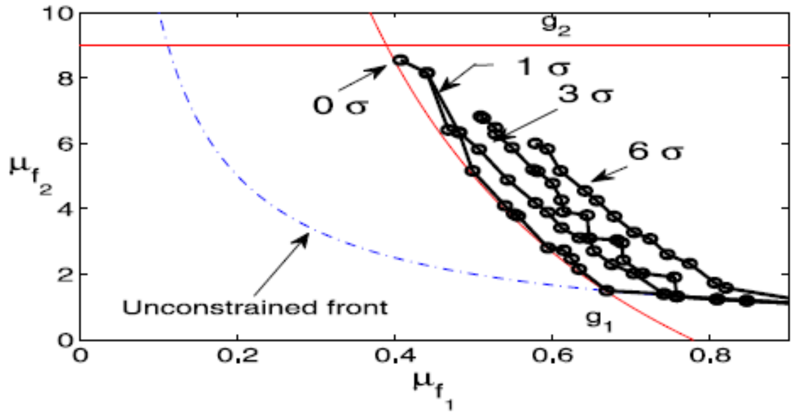


Fig. 6. Solutions obtains using Form-4 robust formulation. The solutions are labeled as (x, f, sigma$_g$, sigma$_f$) i.e., A = (0.1000, −1.000, 0.0012, 0.3952), B = (0.1136, −0.8715, 0.3948, 0.3937), C'' = (0.4892, −0.7151, 6, 3.3624), D = (0.8169, −1.057e-4, 6, 5.4359), and E = (0.9832, −5.89e-5, 6, 6).

Test problem: single-objective



Test problem: bi-objective

Applications studied:

- Welded beam design, Coil compression spring design, Car side impact design problem, Aircraft design, Water resource management, etc.
- The problems span one to ten objectives.

[Asafuddoula et al, IEEE TEVC 2015; Ray et al, JMD 2015]

Robust *re*-design

- Re-Design for Robustness (RDR) represents a practical class of problems, where a limited set of components of *an existing product* are re-designed to improve the overall robustness of the product.
- This avoids the need to design the product from scratch and enables the use of existing inventory of some of the components.
- The central question is given an existing baseline design, which components can be changed to improve the robustness of the product?
- The number of changed components relative to the existing design can be added as an objective while solving the previous (FPR) formulation
- The resulting formulation is referred to as FPRR

$$\underset{(\mathbf{x})}{\text{Minimize}}\ \mu_{f_i(\mathbf{x})}, i = 1, 2, \ldots\ldots\ldots\ldots M$$

$$\underset{(\mathbf{x})}{\text{Maximize}}\, f_{M+1}(\mathrm{x}) = \mathrm{Min}(sigma_g, R_c)$$

$$\underset{(\mathbf{x})}{\text{Maximize}}\, f_{M+2}(\mathrm{x}) = \mathrm{Min}(sigma_f, R_f)$$

$$\underset{(\mathbf{x})}{\text{Minimize}}\, f_{M+3} = F_{nc}(\mathrm{x}, \mathrm{x}^b)$$

subject to

$$sigma_g \equiv \mathrm{Min}(\mu_{g_j(\mathbf{x})}/\sigma_{g_j(\mathbf{x})}) \geq 0$$
$$\mathrm{x}^{(L)} \leq \mathrm{x} \leq \mathrm{x}^{(U)}$$

where

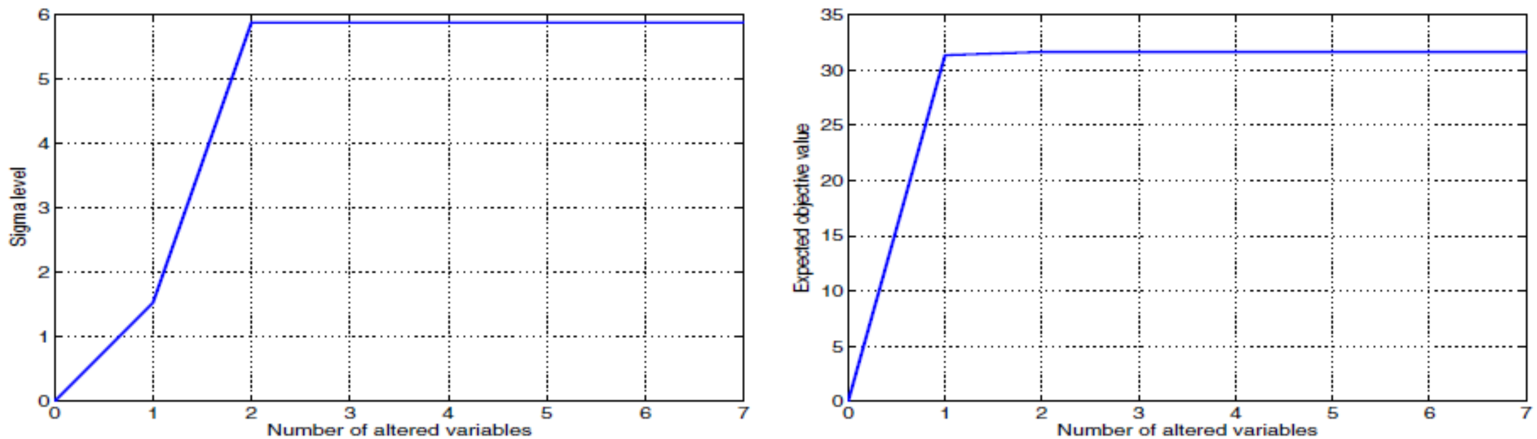
$$sigma_f \equiv \mathrm{Min}(\sigma_{f_{0,i}(\mathbf{x})}/\sigma_{f_i(\mathbf{x})})$$
$$\mathrm{x}^{(L)} \leq \mathrm{x} \leq \mathrm{x}^{(U)}$$

[Singh et al, EMO 2015]

Robust re-design example

Example: Car side impact design problem

- 1 objective, 7 Variables, 10 constraints
- Baseline design available is infeasible (hence sigma level 0)
- The re-design enables it to get to nearly 6-sigma design by changing only 2 of the variables
- Expected performance is very similar to the case of enabling all variables to change



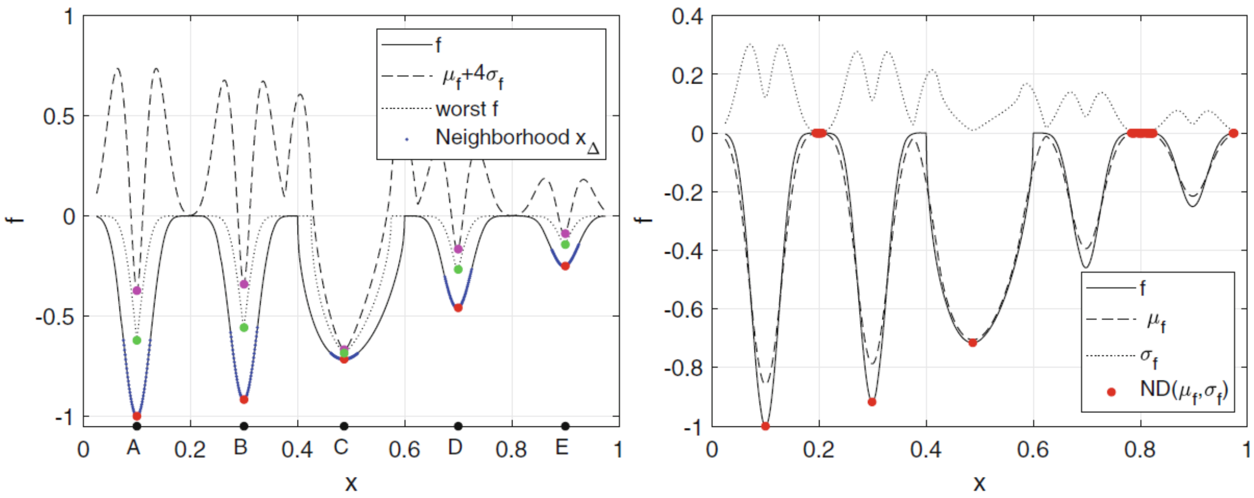
(a) Sigma level of re-designed solutions (b) Expected performance of the re-designed solutions

[Singh et al, EMO 2015]

An alternate many-objective formulation

Consider a single-objective optimization problem with stochasticity

Some basic reformulations: optimize $\mu_f + w\sigma_f$, worst f, or multi-objective (μ_f, σ_f)



- The limitations of the existing optimization formulations include, e.g.
  - $p(f(A) \leq f(C)) = 88.23\%$; yet $\mu_f + 4\sigma_f$ or worst f metrics will (mis-)identify C as the better design among the two.
  - worst f has many indistinguishable regions and needs bilevel formulation.
  - considering all (μ_f, σ_f)-ND solutions yields several designs with poor performance and low uncertainty.

[Singh and Branke, PPSN 2022]

An alternate many-objective formulation

We propose robustness based on the quantile function (QF) of the objective computed within x_Δ

- This function defines, for each possible probability $p \in [0, 1]$ the fitness value that is obtained at least with that probability. More formally:

$$QF(x, p) = \inf\{y \in R : p \leq G(f(x))\}$$

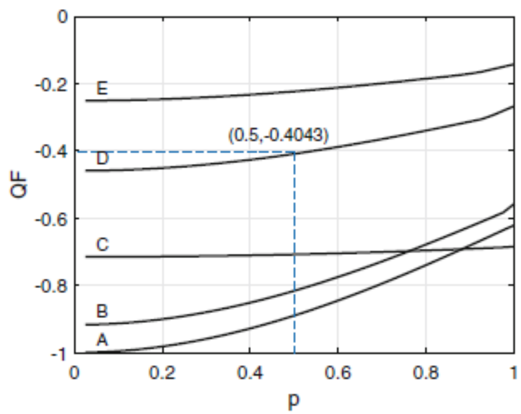
- To identify all first-order stochastically non-dominated solutions, we solve:

$$\min QF(x, p) \quad \forall p \quad s.t. \quad x_i^L \leq x\_i \leq x_i^U, i = 1, \ldots n_x.$$

- In this definition, a solution x_A is considered better than another solution x_B if $QF(x_A)$ yields a lower or equal value than $QF(x_B)$ (for minimization) for all values of $p \in [0, 1]$. This is equivalent to x_A first-order stochastically dominating x_B.

- In this figure,
  - Solution A stochastically dominates B, D, E
  - {A,C} and {B,C} are non-dominated
  - A, B, C, D all dominate E
  - (and so on)

We propose robustness based on objective computed within x_Δ

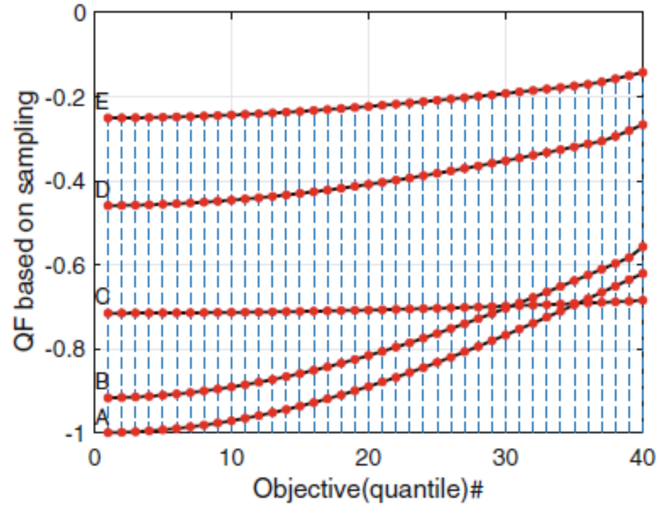


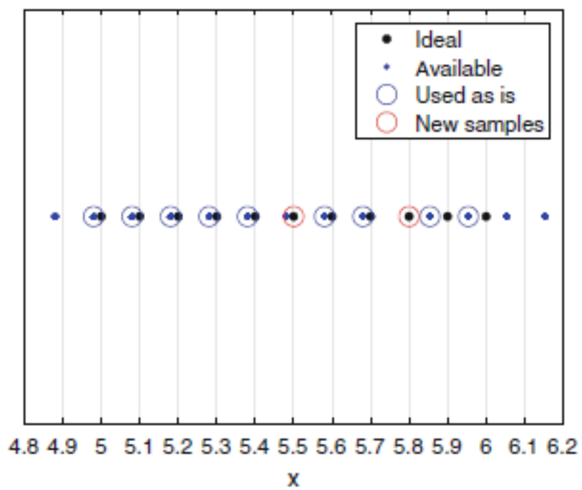
[Singh and Branke, PPSN 2022]

Search

Customized EA for finding stochastically non-dominated solutions:

- **Discretization:** The quantile function is discretized using M uniformly sampled values of $p \in [0\ 1]$; leading to M−objective problem. N_s samples are used to compute the objective values.
- **Environmental selection** using first-order stochastic non-domination ranking:
  - Compute $d_{ij} = \max(\max_q f_q(j) - f_q(i)), 0)$
  - Compute $dMin(i) = \min_{j \in S} d_{ij}$
  - Remove solution with lowest $dMin$; recompute d_{ij}
  - Repeat until all solutions are eliminated; reverse the sequence to get the final ranks
- Strategies for **reducing function evaluations**:
  - reuse the neighbor solutions' samples for QF calculation
  - Use Kriging models to approximate the QF values in lieu of true evaluations





[Singh and Branke, PPSN 2022]

Results

Our experiments thus include four variants:

- V1 = Baseline; neighboring samples OFF, surrogates OFF
- V2 = reuse neighboring samples OFF, surrogates ON
- V3 = reuse neighboring samples ON, surrogates OFF
- V4 = reuse neighboring samples ON, surrogates ON.

Table 1: Median IGD and number of evaluations (FE). The numbers in parenthesis denote the ratio relative to the baseline (V*/V1 for IGD, V1/V* for FE). ↑ and ↓ denote higher or lower than baseline, respectively.

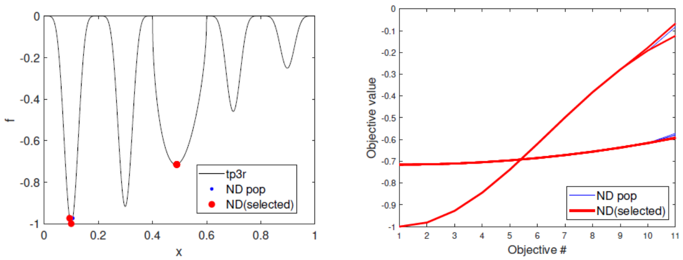
	IGD				FE			
Problem	V1	V2	V3	V4	V1	V2	V3	V4
TP1	0.0002	0.0006 (2.45× ↑)	0.0021 (9.23× ↑)	0.0054 (23.59× ↑)	1.01e+05	13020 (7.76× ↓)	3721 (27.14× ↓)	2824 (35.76× ↓)
TP2	0.0002	0.0004 (2.82× ↑)	0.0006 (3.93× ↑)	0.0006 (3.54× ↑)	1.01e+05	13020 (7.76× ↓)	3958 (25.52× ↓)	2845 (35.5× ↓)
TP3	0.0012	0.0012 (1.05× ↑)	0.0019 (1.62× ↑)	0.0012 (1.05× ↑)	1.01e+05	13020 (7.76× ↓)	4000 (25.25× ↓)	2845 (35.5× ↓)
TP4	0.0004	0.0004 (1.21× ↑)	0.0007 (1.90× ↑)	0.0004 (1.16× ↑)	1.01e+05	13020 (7.76× ↓)	3241 (31.16× ↓)	2341 (43.14× ↓)
TP5	0.0012	0.0022 (1.78× ↑)	0.0019 (1.61× ↑)	0.0023 (1.89× ↑)	1.01e+05	13020 (7.76× ↓)	3984 (25.35× ↓)	2892 (34.92× ↓)
TP6	0.0072	0.0065 (0.91× ↓)	0.0083 (1.15× ↑)	0.0061 (0.85× ↓)	1.01e+05	13020 (7.76× ↓)	4869 (20.74× ↓)	3180 (31.76× ↓)
TP7	0.1421	0.5141 (3.62× ↑)	1.3860 (9.75× ↑)	6.6902 (47.09× ↑)	1.01e+05	13020 (7.76× ↓)	3742 (26.99× ↓)	2776 (36.38× ↓)
TP8	0.0384	0.0395 (1.03× ↑)	0.0450 (1.17× ↑)	0.0436 (1.13× ↑)	1.01e+05	13020 (7.76× ↓)	5315 (19.00× ↓)	3217 (31.4× ↓)
TP9	0.0051	0.0051 (1.01× ↑)	0.0088 (1.73× ↑)	0.0055 (1.08× ↑)	1.01e+05	13020 (7.76× ↓)	3927 (25.72× ↓)	2949 (34.25× ↓)
TP10	0.0117	0.0124 (1.06× ↑)	0.0137 (1.17× ↑)	0.0124 (1.06× ↑)	1.001e+06	31020 (32.27× ↓)	30849 (32.45× ↓)	21031 (47.6× ↓)

- All variants achieve a low median IGD. Thus, the proposed EA is competent in identifying stochastically ND solutions.
- The IGD of V2-V4 is generally 1-4 times that obtained by V1. At the same time, the typical reduction in FE is 8-fold for V2 and 20 to 40-fold for V3-V4 relative to V1. Thus, large savings in FE are achieved with little compromise on solution quality.

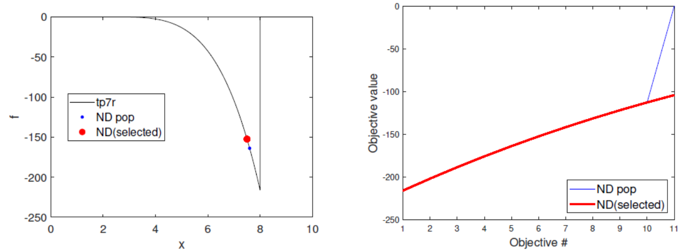
[Singh and Branke, PPSN 2022]

Results

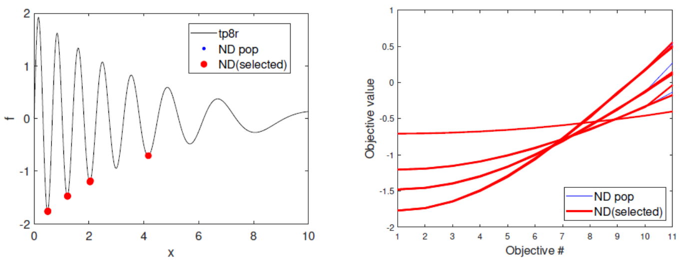
A visualization of the typical results obtained (similar quality for all variants)



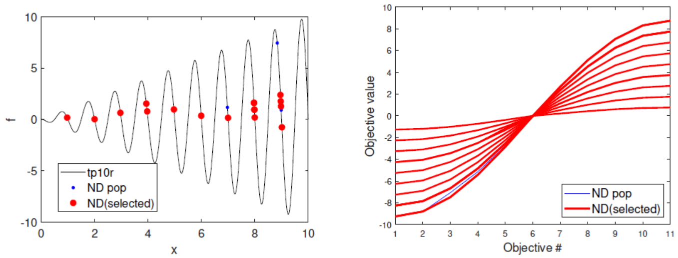
(a) TP3



(b) TP7



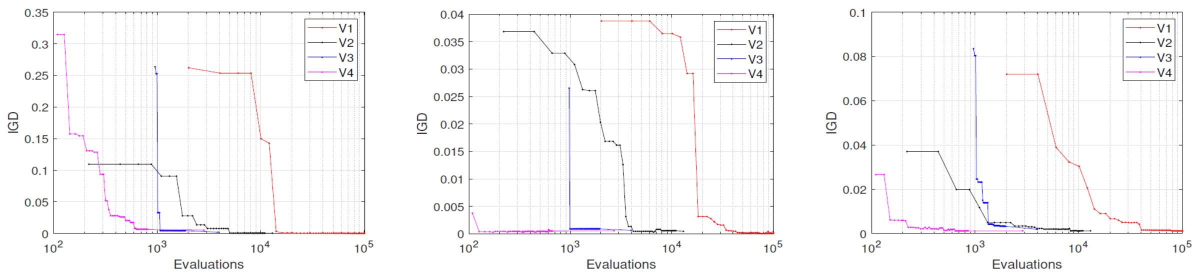
(c) TP8



(d) TP10

For each problem, left subfigure shows solutions in f space, while the right figure shows the solutions in the QF space

The median convergence plots indicate drastic speed-up



Convergence plots for selected problems (TP1, TP2, TP3)

[Singh and Branke, PPSN 2022]

References

The contents for this (many-objective) part of the tutorial are taken from the following:

- Singh, H.K. and Branke, J., “Identifying stochastically non-dominated solutions using evolutionary computation,” in Proceedings of Parallel Problem Solving from Nature, (Dortmund, Germany), pp. 193-206, 2022.
- Asafuddoula, M., Singh, H.K., and Ray, T., “Six Sigma Robust Design Optimization using a Many-objective Decomposition Based Evolutionary Algorithm.” IEEE Transactions on Evolutionary Computation, vol. 19, issue 4, pp. 490-507, 2015
- Ray, T., Asafuddoula, M., Singh, H.K., and Alam, K.,“ An Approach to Identify Six Sigma Robust Solutions of Multi/Many-objective Engineering Design Optimization Problems.” ASME Journal of Mechanical Design, vol. 137, issue 5, pp. 051404-051404, 2015.
- Singh, H.K., Asafuddoula, M., Ray, T. and Alam, K., "Re-design for robustness using many-objective decomposition based evolutionary optimization," in Proceedings of International Conference on Evolutionary Multi-Criteria Optimization (Guimarães, Portugal), Lecture Notes in Computer Science, vol. 9019, pp. 343–357, Springer, 2015.
- Singh, H.K., and Ray, T., “Many-objective optimization in engineering design: Case studies using a decomposition based evolutionary algorithm,” in Advances in Structural and Multidisciplinary Optimization, Proceedings of the Eleventh World Congress of Structural and Multidisciplinary Optimization,(Sydney, Australia), pp. 106-111, 2015.