Evolutionary Large-Scale Global Optimization

An Introduction: Part I

Mohammad Nabi Omidvar[1] Xiaodong Li[2] Daniel Molina[3]
Antonio LaTorre[4]

[1]School of Computer Science, University of Birmingham, UK

[2]School of Science, RMIT University, Australia,

[3]DASCI Andalusian Institute of Data Science, University of Granada, Spain,

[4]Universidad Politécnica de Madrid, Spain,

Outline

Optimization

$$\min \ f(\mathbf{x}), \ \mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n \quad (1)$$

$$s.t.: \ \mathbf{g}(\mathbf{x}) \leq 0 \quad (2)$$

$$\mathbf{h}(\mathbf{x}) = 0 \quad (3)$$

Can be converted to unconstrained optimization using:

- Penalty method;
- Lagrangian;
- Augmented Lagrangian.

Our focus is unconstrained optimization. We must learn how to walk before we can run.

Large Scale Global Optimization (LSGO)

How large is large?

- The notion of large-scale is not fixed.
- Changes over time.
- Differs from problem to problem.
- The dimension at which existing methods start to fail.

State-of-the-art (EC)

- Binary: $\approx$ 1 billion [*a*].
- Integer (linear): $\approx$ 1 billion [*b*], [*c*].
- Real: $\approx$ 1000-5000.

[*a*] Kumara Sastry, David E Goldberg, and Xavier Llora. "Towards billion-bit optimization via a parallel estimation of distribution algorithm". In: *Genetic and Evolutionary Computation Conference*. ACM. 2007, pp. 577–584.

[*b*] Kalyanmoy Deb and Christie Myburgh. "Breaking the Billion-Variable Barrier in Real-World Optimization Using a Customized Evolutionary Algorithm". In: *Genetic and Evolutionary Computation Conference*. ACM. 2016, pp. 653–660.

[*c*] Kalyanmoy Deb and Christie Myburgh. "A population-based fast algorithm for a billion-dimensional resource allocation problem with integer variables". In: *European Journal of Operational Research* 261.2 (2017), pp. 460–474.

Large Scale Global Optimization: Applications

Why large-scale optimization is important?

- Growing applications in various fields.
  - Target shape design optimization [*a*].
  - Satellite layout design [*b*].
  - Parameter estimation in large scale biological systems [*c*].
  - Seismic waveform inversion [*d*].
  - Parameter calibration of water distribution systems [*e*].
  - Vehicle routing [*f*].

[*a*] Zhenyu Yang et al. "Target shape design optimization by evolving B-splines with cooperative coevolution". In: *Applied Soft Computing* 48 (Nov. 2016), pp. 672–682.

[*b*] Hong-Fei Teng et al. "A dual-system variable-grain cooperative coevolutionary algorithm: satellite-module layout design". In: *IEEE transactions on evolutionary computation* 14.3 (Dec. 2010), pp. 438–455.

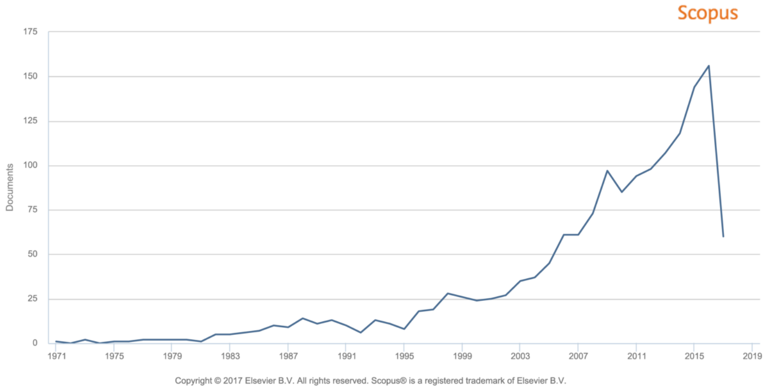
[*c*] Shuhei Kimura et al. "Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm". In: *Bioinformatics* 21.7 (Apr. 2005), pp. 1154–1163.

[*d*] Chao Wang and Jinghuai Gao. "High-dimensional waveform inversion with cooperative coevolutionary differential evolution algorithm". In: *IEEE Geoscience and Remote Sensing Letters* 9.2 (Mar. 2012), pp. 297–301.

[*e*] Yu Wang et al. "Two-stage based ensemble optimization framework for large-scale global optimization". In: *European Journal of Operational Research* 228.2 (2013), pp. 308–320.

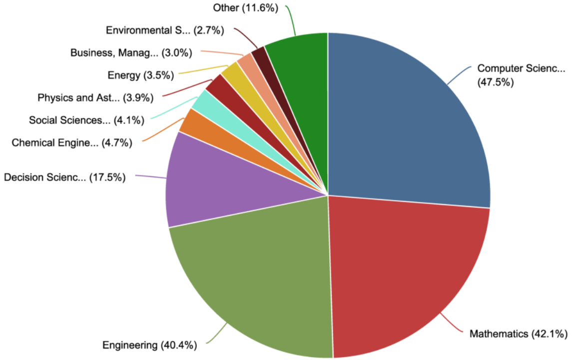
[*f*] Yi Mei, Xiaodong Li, and Xin Yao. "Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems". In: *IEEE Transactions on Evolutionary Computation* 18.3 (2014), pp. 435–449.

Large Scale Global Optimization: Research



Large Scale Global Optimization: Research

Scopus



The Challenge of Large Scale Optimization

Why is it difficult?

- Exponential growth in the size of search space (curse of dimensionality).

Research Goal

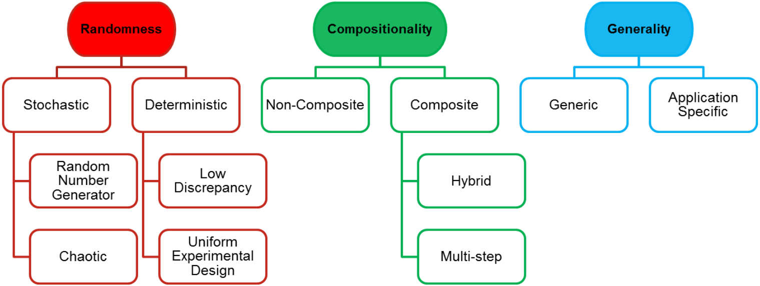
- Improving search quality (get to the optimal point).
- Improving search efficiency (get there fast).

Large Scale Global Optimization: Evolutionary Approaches

1. Initialization
2. Sampling and Variation Operators
3. Approximation and Surrogate Modeling
4. Local Search and Memetic Algorithms
5. Decomposition and Divide-and-Conquer
6. Parallelization (GPU, CPU)
7. Hybridization

Initialization Methods

- Study the importance of initialization methods [1] in large-scale optimization.



[1] Borhan Kazimipour, Xiaodong Li, and A Kai Qin. "A review of population initialization techniques for evolutionary algorithms". In: *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE. 2014, pp. 2585–2592.

Initialization Methods

- Inconclusive evidence for or against initialization methods:
  - Uniform design works worse than RNG, while good-lattice point and opposition-based methods perform better [1].
  - Another study showed that population size has a more significant effect than the initialization [2].
  - Achieving uniformity is difficult in high-dimensional spaces [3].
  - Yet another study suggest comparing average performances may not reveal the effect of initialization [4].
- Shortcomings:
  - It is difficult to isolate the effect of initialization.
  - Different effect on different algorithms (mostly tested on DE).
  - Numerous parameters to study.

---

[1] Borhan Kazimipour, Xiaodong Li, and A Kai Qin. "Initialization methods for large scale global optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2013, pp. 2750–2757.

[2] Borhan Kazimipour, Xiaodong Li, and A Kai Qin. "Effects of population initialization on differential evolution for large scale optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2014, pp. 2404–2411.

[3] Borhan Kazimipour, Xiaodong Li, and A Kai Qin. "Why advanced population initialization techniques perform poorly in high dimension?" In: *SEAL*. 2014, pp. 479–490.

[4] Eduardo Segredo et al. "On the comparison of initialisation strategies in differential evolution for large scale optimisation". In: *Optimization Letters* (2017), pp. 1–14.

Sampling and Variation Operators

- Opposition-based sampling [1]
- Center-based sampling [2].
- Quantum-behaved particle swarm [3].
- Competitive Swarm Optimizer [4].
- Social learning PSO [5].
- Mutation operators [6], [7].

[1] Hui Wang, Zhijian Wu, and Shahryar Rahnamayan. "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems". In: *Soft Computing* 15.11 (2011), pp. 2127–2140.

[2] Sedigheh Mahdavi, Shahryar Rahnamayan, and Kalyanmoy Deb. "Center-based initialization of cooperative co-evolutionary algorithm for large-scale optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 3557–3565.

[3] Deyu Tang et al. "A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous non-linear large scale problems". In: *Information Sciences* 289 (2014), pp. 162–189.

[4] Ran Cheng and Yaochu Jin. "A competitive swarm optimizer for large scale optimization". In: *IEEE Transactions on Cybernetics* 45.2 (2015), pp. 191–204.

[5] Ran Cheng and Yaochu Jin. "A social learning particle swarm optimization algorithm for scalable optimization". In: *Information Sciences* 291 (2015), pp. 43–60.

[6] Hongwei Ge et al. "Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation". In: *Applied Soft Computing* 36 (2015), pp. 300–314.

[7] Ali Wagdy Mohamed and Abdulaziz S Almazyad. "Differential Evolution with Novel Mutation and Adaptive Crossover Strategies for Solving Large Scale Global Optimization Problems". In: *Applied Computational Intelligence and Soft Computing* 2017 (2017).

Approximation Methods and Surrogate Modeling

- High-Dimensional Model Representation (HDMR) [1].
- Radial Basis Functions [2].
- Kriging and Gradient-Enhanced Kriging Metamodels [3].
- Piecewise Polynomial (Spline) [4].
- Turning large-scale problems into expensive optimization problems [5].

---

[1] Enying Li, Hu Wang, and Fan Ye. "Two-level Multi-surrogate Assisted Optimization method for high dimensional nonlinear problems". In: *Applied Soft Computing* 46 (2016), pp. 26–36.

[2] Rommel G Regis. "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions". In: *IEEE Transactions on Evolutionary Computation* 18.3 (2014), pp. 326–347.

[3] Selvakumar Ulaganathan et al. "A hybrid sequential sampling based metamodelling approach for high dimensional problems". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 1917–1923.

[4] Zhenyu Yang et al. "Target shape design optimization by evolving B-splines with cooperative coevolution". In: *Applied Soft Computing* 48 (Nov. 2016), pp. 672–682.

[5] Peng Yang, Ke Tang, and Xin Yao. "Turning high-dimensional optimization into computationally expensive optimization". In: *IEEE Transactions on Evolutionary Computation* 22.1 (2018), pp. 143–156.

Local Search and Memetic Algorithms

- Multiple Trajectory Search (MTS) [1].
- Memetic algorithm with local search chaining [2].
  - MA-SW-Chains [3].
  - MA-SSW-Chains [4].
- Multiple offspring sampling (MOS) [5], [6].

[1] Lin-Yu Tseng and Chun Chen. "Multiple trajectory search for large scale global optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2008, pp. 3052–3059.

[2] Daniel Molina, Manuel Lozano, and Francisco Herrera. "Memetic algorithm with local search chaining for large scale continuous optimization problems". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2009, pp. 830–837.

[3] Daniel Molina, Manuel Lozano, and Francisco Herrera. "MA-SW-Chains: Memetic algorithm based on local search chains for large scale continuous global optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.

[4] Daniel Molina et al. "Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains". In: *Soft Computing* 15.11 (2011), pp. 2201–2220.

[5] Antonio LaTorre, Santiago Muelas, and José-María Peña. "Multiple offspring sampling in large scale global optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2012, pp. 1–8.

[6] Antonio LaTorre, Santiago Muelas, and José-María Peña. "A MOS-based dynamic memetic differential evolution algorithm for continuous optimization: a scalability test". In: *Soft Computing* 15.11 (2011), pp. 2187–2199.

Parallelization

- Algorithms capable of parallelization [1], [2].
- GPU [3], [4].
- CPU/OpenMP [5].

[1] Jing Tang, Meng Hiot Lim, and Yew Soon Ong. "Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems". In: *Soft Computing* 11.9 (2007), pp. 873–888.

[2] Hui Wang, Shahryar Rahnamayan, and Zhijian Wu. "Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems". In: *Journal of Parallel and Distributed Computing* 73.1 (2013), pp. 62–73.

[3] Kumara Sastry, David E Goldberg, and Xavier Llora. "Towards billion-bit optimization via a parallel estimation of distribution algorithm". In: *Genetic and Evolutionary Computation Conference*. ACM. 2007, pp. 577–584.

[4] Alberto Cano and Carlos García-Martínez. "100 Million dimensions large-scale global optimization using distributed GPU computing". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 3566–3573.

[5] AJ Umbarkar. "OpenMP Genetic Algorithm for Continuous Nonlinear Large-Scale Optimization Problems". In: *International Conference on Soft Computing for Problem Solving*. Springer. 2016, pp. 203–214.

Hybridization (The best of both worlds)

- Rationale: benefiting from unique features of different optimizers.
  - EDA+DE: [1].
  - PSO+ABC: [2].
  - Different DE variants: JADE+SaNSDE [3].
  - PSO+ACO [4].
  - Minimum Population Search+CMA-ES [5].

---

[1] Yu Wang, Bin Li, and Thomas Weise. "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems". In: *Information Sciences* 180.12 (2010), pp. 2405–2420.

[2] LN Vitorino, SF Ribeiro, and Carmelo JA Bastos-Filho. "A hybrid swarm intelligence optimizer based on particles and artificial bees for high-dimensional search spaces". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2012, pp. 1–6.

[3] Sishi Ye et al. "A hybrid adaptive coevolutionary differential evolution algorithm for large-scale optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2014, pp. 1277–1284.

[4] Wu Deng et al. "A novel two-stage hybrid swarm intelligence optimization algorithm and application". In: *Soft Computing* 16.10 (2012), pp. 1707–1722.

[5] Antonio Bolufé-Röhler, Sonia Fiol-González, and Stephen Chen. "A minimum population search hybrid for large scale global optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2015, pp. 1958–1965.

Decomposition Methods

- Divide-and-conquer

Variable Interaction, Linkage, Epistasis

What is variable interaction?

- Genetics: two genes are said to interact with each other if they collectively represent a feature at the phenotype level.
- The extent to which the fitness of one gene can be suppressed by another gene.
- The extent to which the value taken by one gene activates or deactivates the effect of another gene.

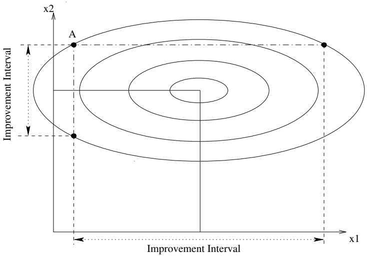
Why variable interaction?

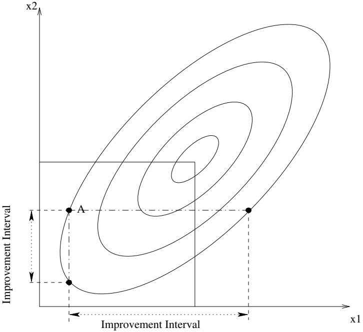
- The effectiveness of optimization algorithms is affected by how much they take variable interaction into account.
- Also applies to classic mathematical programming methods.

Variable Interaction, Linkage, Epistasis

Illustrative Example

- $f(x, y) = x^2 + \lambda_1 y^2$
- $g(x, y) = x^2 + \lambda_1 y^2 + \lambda_2 xy$





Definitions

Variable Interaction

A variable x_i is separable or does not interact with any other variable iff:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) = \Big(\arg\min_{x_i} f(\mathbf{x}), \arg\min_{\forall x_j, j \neq i} f(\mathbf{x})\Big),$$

where $\mathbf{x} = (x_1, \ldots, x_n)^\top$ is a decision vector of n dimensions.

Partial Separability

A function $f(\mathbf{x})$ is partially separable with m independent subcomponents iff:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) = \Big(\arg\min_{\mathbf{x}_1} f(\mathbf{x}_1, \ldots), \ldots, \arg\min_{\mathbf{x}_m} f(\ldots, \mathbf{x}_m)\Big),$$

$\mathbf{x}_1, \ldots, \mathbf{x}_m$ are disjoint sub-vectors of $\mathbf{x}$, and $2 \leq m \leq n$.

Note: a function is fully separable if sub-vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are 1-dimensional (i.e., $m = n$).

Definitions

Full Nonseparability

A function $f(\mathbf{x})$ is fully non-separable if every pair of its decision variables interact with each other.

Additive Separability

A function is *partially additively separable* if it has the following general form:

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_i) \ ,$$

where $\mathbf{x}_i$ are mutually exclusive decision vectors of f_i, $\mathbf{x} = (x_1, \dots, x_n)^\top$ is a global decision vector of n dimensions, and m is the number of independent subcomponents.

Effect of Variable Interaction (1)

Sampling and Variation Operators:

- GAs: inversion operator to promote tight linkage [1].
  - Increasing the likelihood of placing linked genes close to each other to avoid disruption by crossover.
  - Rotation of the landscape has a detrimental effect on GA [2].
- The need for rotational invariance:
  - Model Building Methods:
    - ★ Estimation of Distribution Algorithms and Evolutionary Strategies: Covariance Matrix Adaptation.
    - ★ Bayesian Optimization: Bayesian Networks.
  - DE's crossover is not rotationally invariant.
  - PSO is also affected by rotation [3].

---

[1] David E Goldberg, Robert Lingle, et al. "Alleles, loci, and the traveling salesman problem". In: *International Conference on Genetic Algorithms and Their Applications*. Vol. 154. 1985, pp. 154–159.

[2] Ralf Salomon. "Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms". In: *BioSystems* 39.3 (1996), pp. 263–278.

[3] Daniel N Wilke, Schalk Kok, and Albert A Groenwold. "Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance". In: *International journal for numerical methods in engineering* 70.8 (2007), pp. 985–1008.

Effect of Variable Interaction (2)

1. Approximation and Surrogate Modelling:
   - Should be able to capture variable interaction.
   - Second order terms of HDMR.
2. Local Search and Memetic Algorithms:
   - What subset of variables should be optimized in each iteration of local search?
   - Coordinate-wise search may not be effective. Memetics perform well on separable functions! A coincidence?!
3. Decomposition and Divide-and-Conquer:
   - Interacting variables should be placed in the same component.

Linkage Learning and Exploiting Modularity

- Implicit Methods:
  - In EC:
    - Estimation of Distribution Algorithms
    - Bayesian Optimization: BOA, hBOA, Linkage Trees
    - Adaptive Encoding, CMA-ES
  - Classic Optimization:
    - Quasi-Newton Methods: Approximation of the Hessian.
    - Adaptive Coordinate Descent
- Explicit Methods:
  - In EC:
    - Random Grouping
    - Statistical Correlation-Based Methods
    - Delta Grouping
    - Meta Modelling
    - Monotonicity Checking
    - Differential Grouping
  - Classic Optimization
    - Block Coordinate Descent

Implicit Methods

- Scaling Up EDAs:
  - Model Complexity Control [1].
  - Random Matrix Projection [2].
  - Use of mutual information [3].
  - Cauchy-EDA [4].

[1] Weishan Dong et al. "Scaling up estimation of distribution algorithms for continuous optimization". In: *IEEE Transactions on Evolutionary Computation* 17.6 (2013), pp. 797–822.

[2] Ata Kabán, Jakramate Bootkrajang, and Robert John Durrant. "Toward large-scale continuous EDA: A random matrix theory perspective". In: *Evolutionary Computation* 24.2 (2016), pp. 255–291.

[3] Qi Xu, Momodou L Sanyang, and Ata Kabán. "Large scale continuous EDA using mutual information". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 3718–3725.

[4] Momodou L Sanyang, Robert J Durrant, and Ata Kabán. "How effective is Cauchy-EDA in high dimensions?" In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 3409–3416.

Implicit Methods

- Scaling up CMA-ES:
  - CC-CMA-ES [1].
  - sep-CMA-ES [2]
  - Reducing space complexity:
    - L-CMA-ES [3].
    - LM-CMA [4].

[1] Jinpeng Liu and Ke Tang. "Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2013, pp. 350–357.

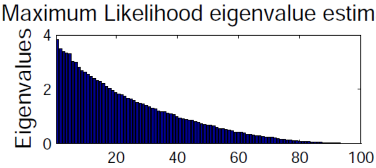
[2] Raymond Ros and Nikolaus Hansen. "A simple modification in CMA-ES achieving linear time and space complexity". In: *Parallel Problem Solving from Nature*. Springer. 2008, pp. 296–305.

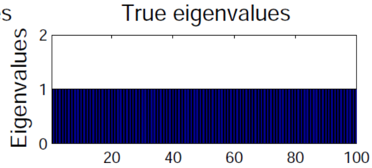
[3] James N Knight and Monte Lunacek. "Reducing the space-time complexity of the CMA-ES". In: *Genetic and Evolutionary Computation Conference*. ACM. 2007, pp. 658–665.

[4] Ilya Loshchilov. "LM-CMA: An Alternative to L-BFGS for Large-Scale Black Box Optimization". In: *Evolutionary Computation* (2015).

Scalability issues of EDAs

- Accurate estimation requires a large sample size which grows exponentially with the dimensionality of the problem [1].
- A small sample results in poor estimation of the eigenvalues [2].
- The cost of sampling from a multi-dimensional Gaussian distribution increases cubically with the problem size [3].





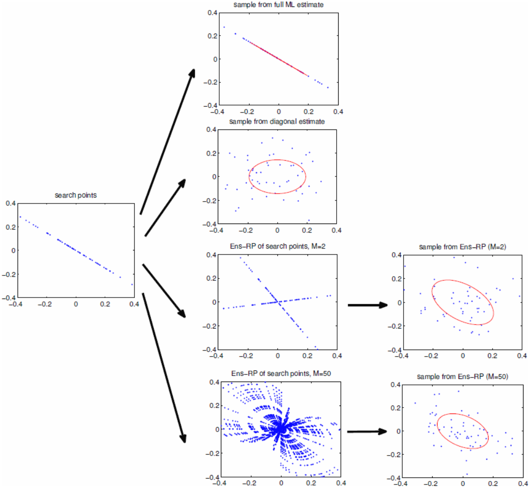
[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin, 2001.

[2] Roman Vershynin. "Introduction to the non-asymptotic analysis of random matrices". In: *arXiv preprint arXiv:1011.3027* (2010).

[3] Weishan Dong and Xin Yao. "Unified eigen analysis on multivariate Gaussian based estimation of distribution algorithms". In: *Information Sciences* 178.15 (2008), pp. 3000–3023.

Random Projection EDA [1]

.



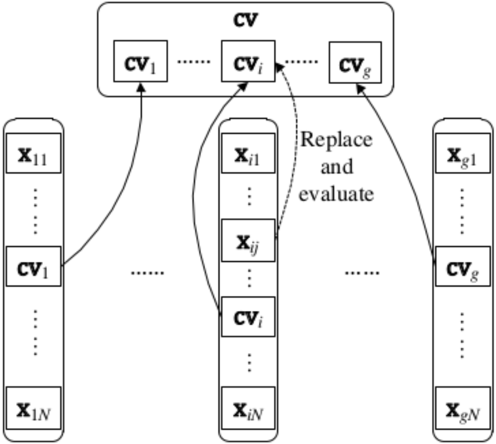
[1] Ata Kabán, Jakramate Bootkrajang, and Robert John Durrant. "Toward large-scale continuous EDA: A random matrix theory perspective". In: *Evolutionary Computation* 24.2 (2016), pp. 255–291.

Explicit Methods

- A large problem can be subdivided into smaller and simpler problems.
- Dates back to René Descartes (*Discourse on Method*).
- Has been widely used in many areas:
  - Computer Science: Sorting algorithms (quick sort, merge sort)
  - Optimization: Large-scale linear programs (Dantzig)
  - Politics: Divide and rule (In *Perpetual Peace* by Immanuel Kant: *Divide et impera* is the third political maxims.)

Acknowledgement: the above image is obtained from: http://draininbrain.blogspot.com.au/

Decomposition in EAs: Cooperative Co-evolution [1]



[1] Mitchell A. Potter and Kenneth A. De Jong. "A cooperative coevolutionary approach to function optimization". In: *Proc. Int. Conf. Parallel Problem Solving from Nature.* Vol. 2. 1994, pp. 249–257.

CC is a Framework

CC as a scalability agent:

- CC is not an optimizer.
- Requires a component optimizer.
- CC coordinates how the component optimizer is applied to components.
- A scalability agent.

Challenges of CC

Main Questions

1. How to decompose the problem?
2. How to allocated resources?
3. How to coordinate?

The Decomposition Challenge

How to decompose?

- There are many possibilities.
- Which decomposition is the best?

Optimal decomposition

- It is governed by the interaction structure of decision variables.
- An optimal decomposition is the one that minimizes the interaction between components.

Survey of Decomposition Methods

- Uninformed Decomposition [1]
  - n 1-dimensional components (the original CC)
  - k s-dimensional components.
- Random Grouping [2]
- Statistical Correlation-Based Methods
- Delta Grouping [3]
- Meta Modelling [4]
- Monotonicity Checking [5]
- Differential Grouping [6]

[1] F. van den Bergh and Andries P Engelbrecht. "A cooperative approach to particle swarm optimization". In: *IEEE Transactions on Evolutionary Computation* 2.3 (June 2004), pp. 225–239.

[2] Zhenyu Yang, Ke Tang, and Xin Yao. "Large scale evolutionary optimization using cooperative coevolution". In: *Information Sciences* 178.15 (2008), pp. 2985–2999.

[3] Mohammad Nabi Omidvar, Xiaodong Li, and Xin Yao. "Cooperative co-evolution with delta grouping for large scale non-separable function optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.

[4] Sedigheh Mahdavi, Mohammad Ebrahim Shiri, and Shahryar Rahnamayan. "Cooperative co-evolution with a new decomposition method for large-scale optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2014, pp. 1285–1292.

[5] Wenxiang Chen et al. "Large-scale global optimization using cooperative coevolution with variable interaction learning". In: *Parallel Problem Solving from Nature*. Springer. 2010, pp. 300–309.

[6] Mohammad Nabi Omidvar et al. "Cooperative co-evolution with differential grouping for large scale optimization". In: *IEEE Transactions on Evolutionary Computation* 18.3 (2014), pp. 378–393.

Illustrative Example (Canonical CC)

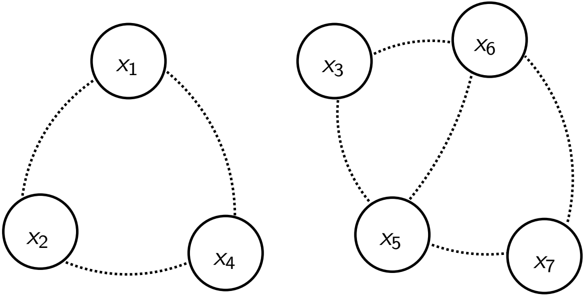


Figure: Variable interaction of a hypothetical function.

- n 1-dimensional components:
  - C_1: $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}$
  - C_2: $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}$
  - ...
  - C_c: $\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}$

Illustrative Example (fixed k s-dimensional)

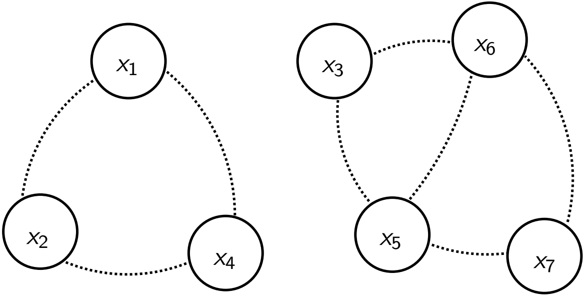


Figure: Variable interaction of a hypothetical function.

- k s-dimensional ($k = 2, s = 4$):
  - C_1: $\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}$
  - C_2: $\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}$
  - ...
  - C_c: $\{x_1, x_2, x_3, x_4\}, \{x_5, x_6, x_7\}$

Illustrative Example (Random Grouping)

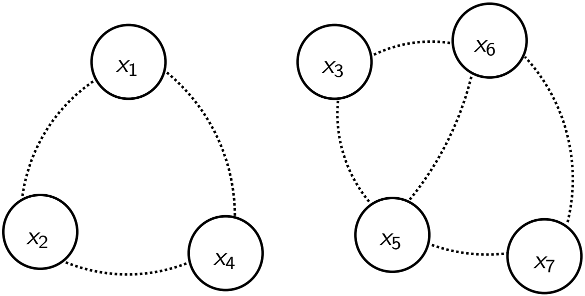


Figure: Variable interaction of a hypothetical function.

- Random Grouping ($k = 2, s = 4$):
  - C_1: $\{x_2, x_3, x_6, x_5\}, \{x_7, x_1, x_4\}$
  - C_2: $\{x_3, x_4, x_1, x_2\}, \{x_6, x_7, x_5\}$
  - ...
  - C_c: $\{x_1, x_5, x_6, x_7\}, \{x_2, x_4, x_3\}$

Random Grouping

Theorem

Given N cycles, the probability of assigning v interacting variables $x_1, x_2, ..., x_v$ into one subcomponent for at least k cycles is:

$$P(X \geq k) = \sum_{r=k}^{N} \binom{N}{r} \left(\frac{1}{m^{v-1}}\right)^r \left(1 - \frac{1}{m^{v-1}}\right)^{N-r} \quad (4)$$

where N is the number of cycles, v is the total number of interacting variables, m is the number of subcomponents, and the random variable X is the number of times that v interacting variables are grouped in one subcomponent.

Random Grouping

Example

Given $n = 1000$, $m = 10$, $N = 50$ and $v = 4$, we have:

$$P(X \geq 1) = 1 - P(X = 0) = 1 - \left(1 - \frac{1}{10^3}\right)^{50} = 0.0488$$

which means that over 50 cycles, the probability of assigning 4 interacting variables into one subcomponent for at least 1 cycle is only 0.0488. As we can see this probability is very small, and it will be even less if there are more interacting variables.

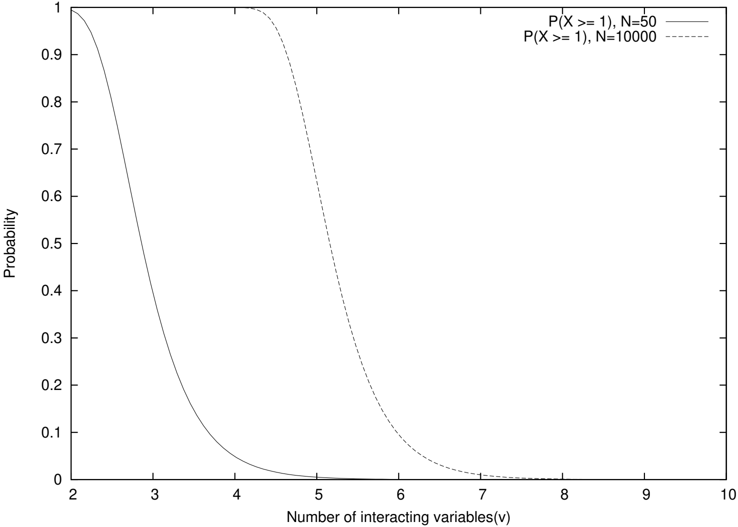


Figure: Increasing v, the number of interacting variables will significantly decrease the probability of grouping them in one subcomponent, given $n = 1000$ and $m = 10$.

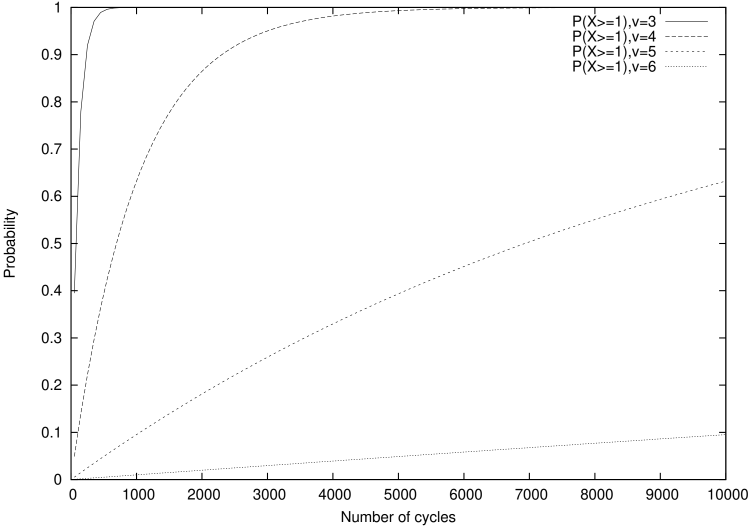


Figure: Increasing N, the number of cycle increases the probability of grouping v number of interacting variables in one subcomponent.

Illustrative Example (Informed with Fixed Groups)

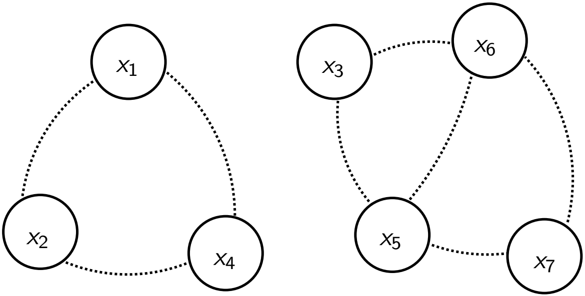
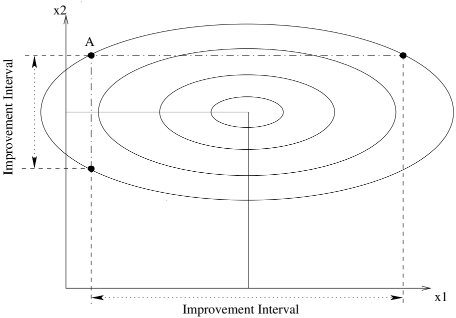
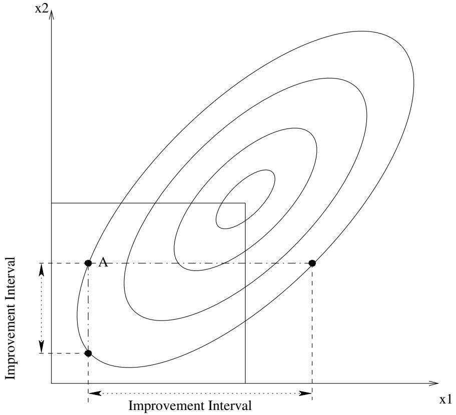


Figure: Variable interaction of a hypothetical function.

- Delta Grouping ($k = 2, s = 4$):
  - C_1: $\{x_1, x_5, x_2, x_4\}, \{x_3, x_6, x_7\}$
  - C_2: $\{x_3, x_5, x_6, x_7\}, \{x_1, x_2, x_4\}$
  - ...
  - C_c: $\{x_3, x_6, x_1, x_4\}, \{x_2, x_5, x_7\}$

Delta Grouping





Informed Decompositions with Fixed Groups

- Adaptive Variable Partitioning [1].
- Delta Grouping [2].
- Min/Max-Variance Decomposition (MiVD/MaVD) [3].
  - Sorts the dimensions based on the diagonal elements of the covariance matrix in CMA-ES.
- Fitness Difference Partitioning [4], [5], [6].

---

[1] Tapabrata Ray and Xin Yao. "A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2009, pp. 983–989.

[2] Mohammad Nabi Omidvar, Xiaodong Li, and Xin Yao. "Cooperative co-evolution with delta grouping for large scale non-separable function optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.

[3] Jinpeng Liu and Ke Tang. "Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution". In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer. 2013, pp. 350–357.

[4] Eman Sayed, Daryl Essam, and Ruhul Sarker. "Dependency identification technique for large scale optimization problems". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2012, pp. 1–8.

[5] Eman Sayed et al. "Decomposition-based evolutionary algorithm for large scale constrained problems". In: *Information Sciences* 316 (2015), pp. 457–486.

[6] Adan E Aguilar-Justo and Efrén Mezura-Montes. "Towards an improvement of variable interaction identification for large-scale constrained problems". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 4167–4174.

Informed Decompositions with Variable Groups

- Multilevel Grouping: MLCC [1], MLSoft [2].
- Adaptive Variable Partitioning 2 [3].
- 4CDE [4].
- Fuzzy Clustering [5].

[1] Zhenyu Yang, Ke Tang, and Xin Yao. "Multilevel cooperative coevolution for large scale optimization". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2008, pp. 1663–1670.

[2] Mohammad Nabi Omidvar, Yi Mei, and Xiaodong Li. "Effective decomposition of large-scale separable continuous functions for cooperative co-evolutionary algorithms". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2014, pp. 1305–1312.

[3] Hemant Kumar Singh and Tapabrata Ray. "Divide and conquer in coevolution: A difficult balancing act". In: *Agent-Based Evolutionary Search*. Springer, 2010, pp. 117–138.

[4] Yazmin Rojas and Ricardo Landa. "Towards the use of statistical information and differential evolution for large scale global optimization". In: *International Conference on Electrical Engineering Computing Science and Automatic Control*. IEEE. 2011, pp. 1–6.

[5] Jianchao Fan, Jun Wang, and Min Han. "Cooperative coevolution for large-scale optimization based on kernel fuzzy clustering and variable trust region methods". In: *IEEE Transactions on Fuzzy Systems* 22.4 (2014), pp. 829–839.

Illustrative Example (Exact Methods)

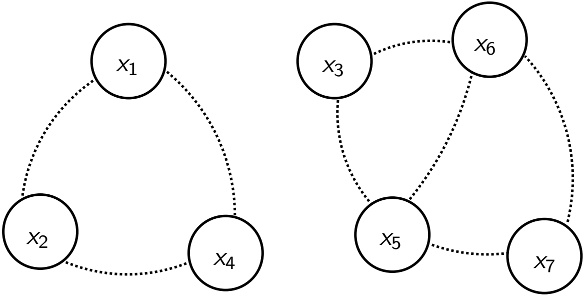
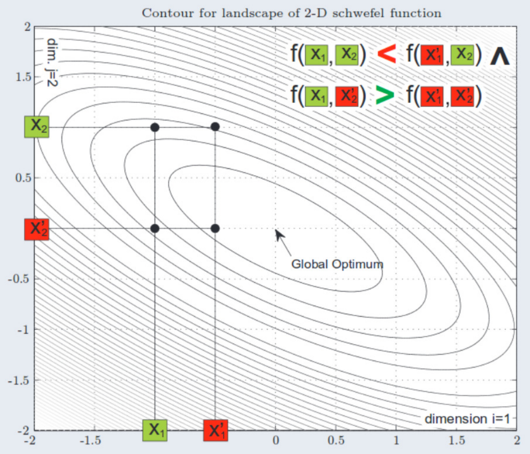


Figure: Variable interaction of a hypothetical function.

- Differential Grouping and Variable Interaction Learning:
  - C_1: $\{x_1, x_2, x_4\}, \{x_3, x_5, x_6, x_7\}$
  - C_2: $\{x_1, x_2, x_4\}, \{x_3, x_5, x_6, x_7\}$
  - ...
  - C_c: $\{x_1, x_2, x_4\}, \{x_3, x_5, x_6, x_7\}$

Monotonicity Check

$$\exists \mathbf{x}, x_i', x_j' : f(x_1, ..., x_i, ..., x_j, ..., x_n) < f(x_1, ..., x_i', ..., x_j, ..., x_n) \wedge$$
$$f(x_1, ..., x_i, ..., x_j', ..., x_n) > f(x_1, ..., x_i', ..., x_j', ..., x_n)$$



Monotonicity Check (Algorithms)

- Linkage Identification by Non-Monotonicity Detection [1]
- Adaptive Coevolutionary Learning [2]
- Variable Interaction Learning [3]
- Variable Interdependence Learning [4]
- Fast Variable Interdependence [5]

[1] Masaharu Munetomo and David E Goldberg. "Linkage identification by non-monotonicity detection for overlapping functions". In: *Evolutionary Computation* 7.4 (1999), pp. 377–398.

[2] Karsten Weicker and Nicole Weicker. "On the improvement of coevolutionary optimizers by learning variable interdependencies". In: *IEEE Congress on Evolutionary Computation*. Vol. 3. IEEE. 1999, pp. 1627–1632.

[3] Wenxiang Chen et al. "Large-scale global optimization using cooperative coevolution with variable interaction learning". In: *Parallel Problem Solving from Nature*. Springer. 2010, pp. 300–309.

[4] Liang Sun et al. "A cooperative particle swarm optimizer with statistical variable interdependence learning". In: *Information Sciences* 186.1 (2012), pp. 20–39.

[5] Hongwei Ge et al. "Cooperative differential evolution with fast variable interdependence learning and cross-cluster mutation". In: *Applied Soft Computing* 36 (2015), pp. 300–314.

Differential Grouping [1]

Theorem

Let $f(\mathbf{x})$ *be an additively separable function.* $\forall a, b_1 \neq b_2, \delta \in \mathbb{R}, \delta \neq 0$, *if the following condition holds*

$$\Delta_{\delta,x_p}[f](\mathbf{x})|_{x_p=a,x_q=b_1} \neq \Delta_{\delta,x_p}[f](\mathbf{x})|_{x_p=a,x_q=b_2}, \tag{5}$$

then x_p *and* x_q *are non-separable, where*

$$\Delta_{\delta,x_p}[f](\mathbf{x}) = f(\ldots, x_p + \delta, \ldots) - f(\ldots, x_p, \ldots), \tag{6}$$

refers to the forward difference of f *with respect to variable* x_p *with interval* δ.

[1] Mohammad Nabi Omidvar et al. "Cooperative co-evolution with differential grouping for large scale optimization". In: *IEEE Transactions on Evolutionary Computation* 18.3 (2014), pp. 378–393.

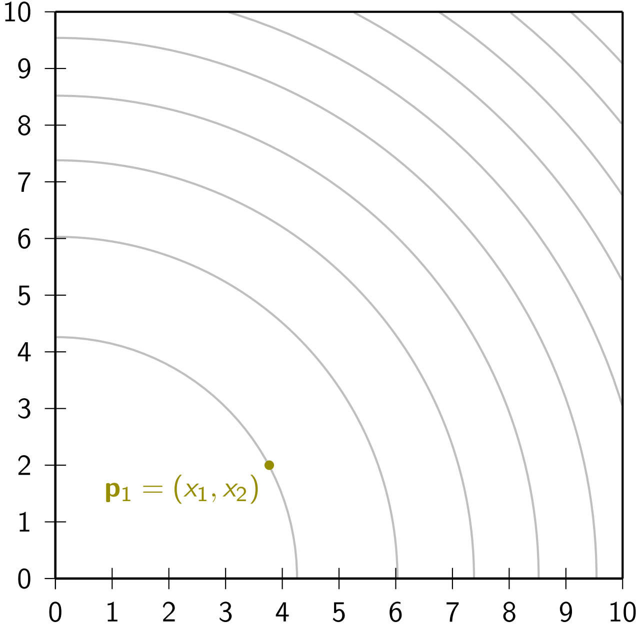


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

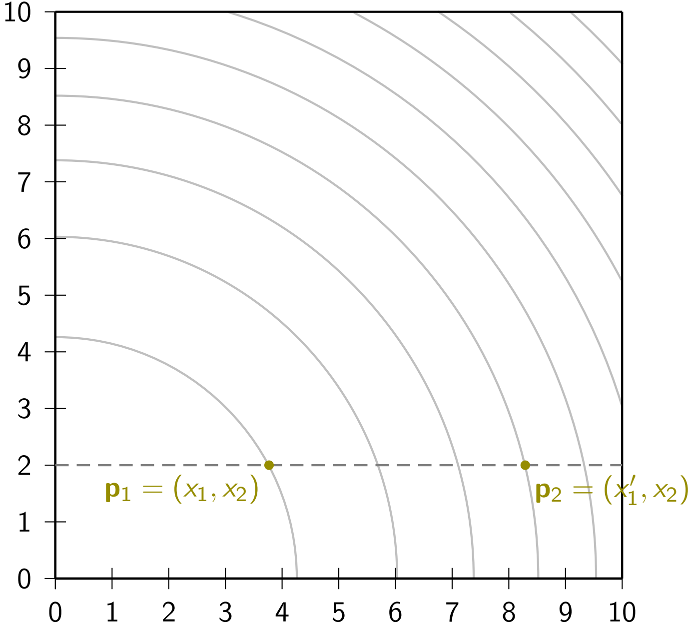


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

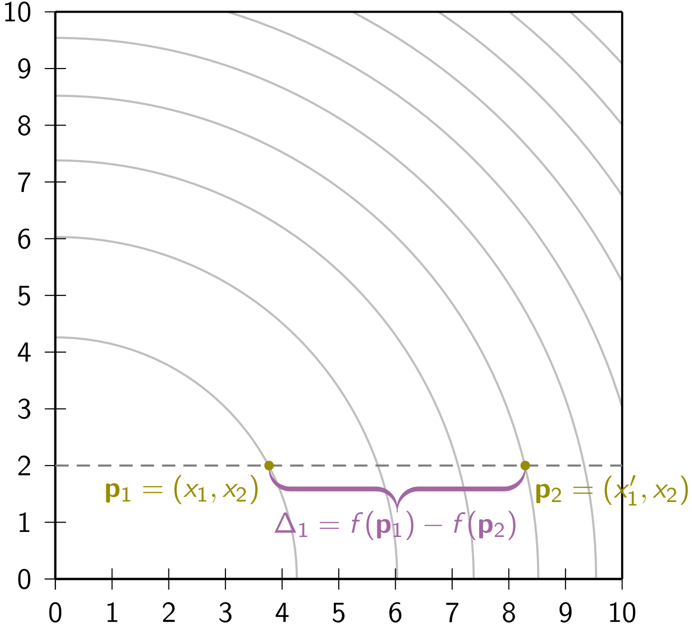


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

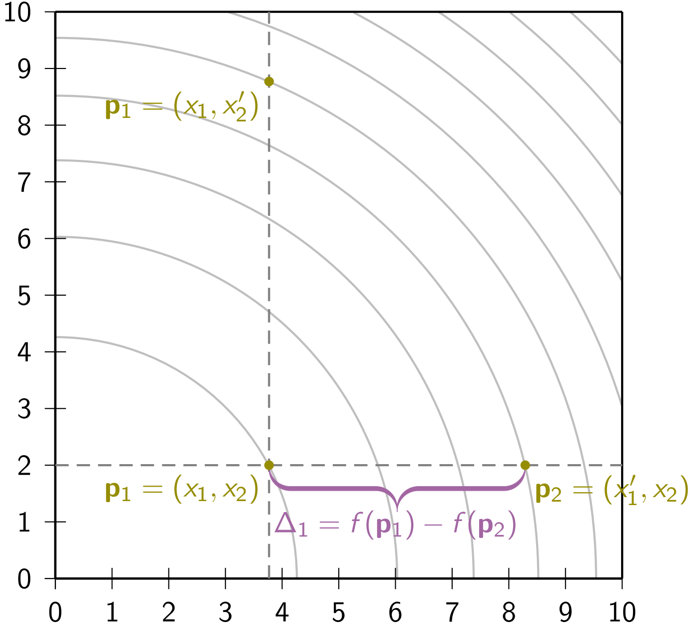


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

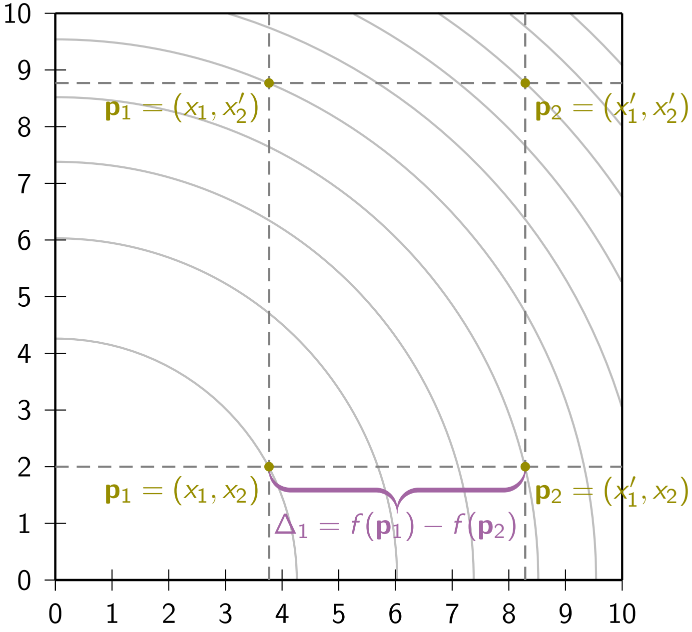


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

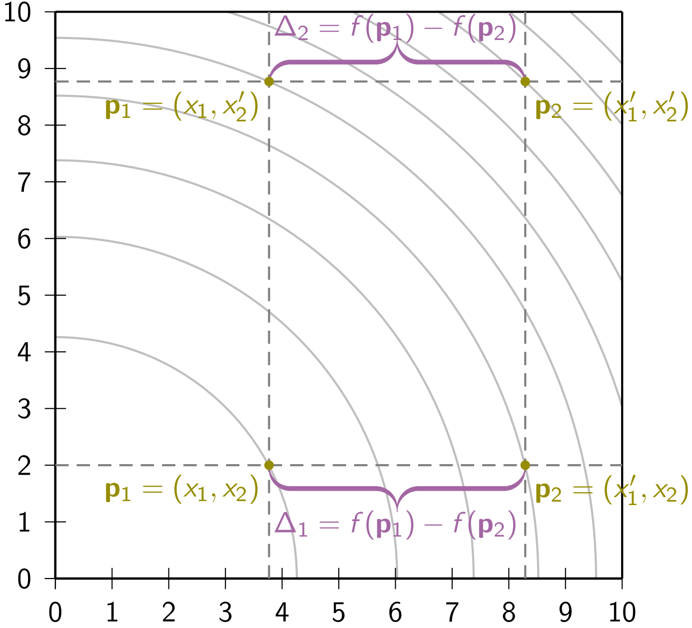


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

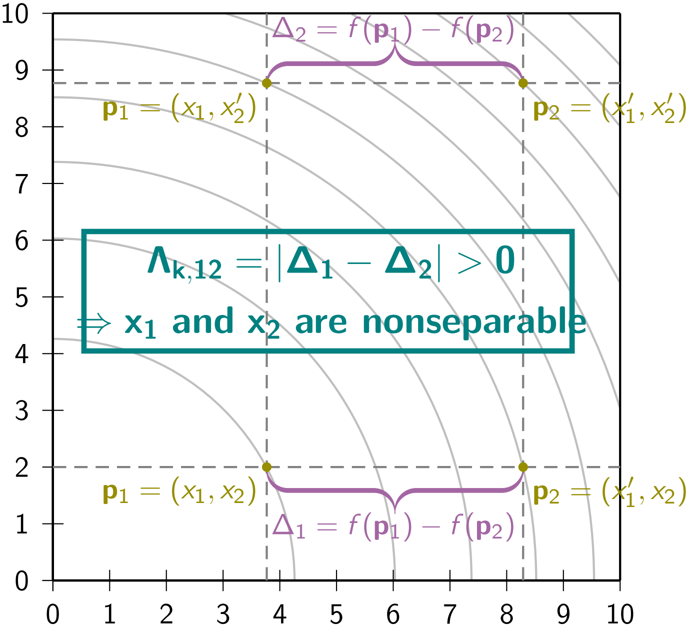


Figure: $f(x_1, x_2) = x_1^2 + x_2^2$

Separability $\Rightarrow \Delta_1 = \Delta_2$

Assuming:

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_i)$$

We prove that:

$$\text{Separability} \;\Rightarrow \Delta_1 = \Delta_2$$

By contraposition ($P \Rightarrow Q \;\equiv\; \neg Q \Rightarrow \neg P$):

$$\Delta_1 \neq \Delta_2 \Rightarrow \;\text{non-separability}$$

or

$$|\Delta_1 - \Delta_2| > \epsilon \Rightarrow \;\text{non-separability}$$

The Differential Grouping Algorithm

Detecting Non-separable Variables

$$|\Delta_1 - \Delta_2| > \epsilon \Rightarrow \text{ non-separability}$$

Detecting Separable Variables

$$|\Delta_1 - \Delta_2| \leq \epsilon \Rightarrow \text{ Separability (more plausible)}$$

Example

Consider the non-separable objective function $f(x_1, x_2) = x_1^2 + \lambda x_1 x_2 + x_2^2$, $\lambda \neq 0$.

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 + \lambda x_2.$$

This clearly shows that the change in the global objective function with respect to x_1 *is a function of* x_1 *and* x_2*. By applying the Theorem:*

$$\begin{aligned}\Delta_{\delta,x_1}[f] &= \left[(x_1+\delta)^2 + \lambda(x_1+\delta)x_2 + x_2^2\right] - \left[x_1^2 + \lambda x_1 x_2 + x_2^2\right] \\ &= \boxed{\delta^2 + 2\delta x_1 + \lambda x_2 \delta.}\end{aligned}$$

Differential Grouping vs CCVIL

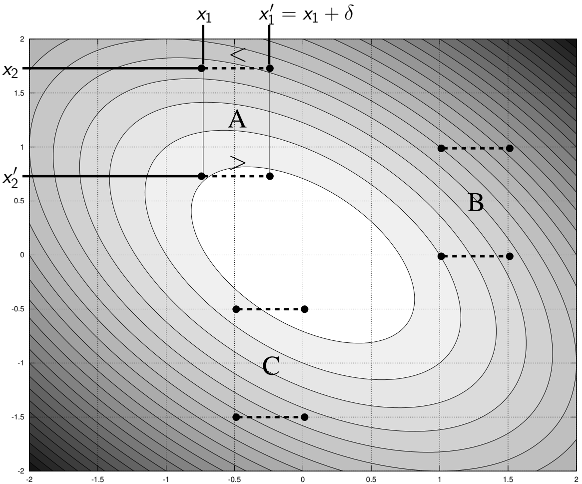


Figure: Detection of interacting variables using differential grouping and CCVIL on different regions of a 2D Schwefel Problem 1.2.

Differential Grouping Family of Algorithms

- Linkage Identification by Non-linearity Check (LINC, LINC-R) [1]
- Differential Grouping (DG) [2]
- Global Differential Grouping (GDG) [3]
- Improved Differential Grouping (IDG) [4]
- eXtended Differential Grouping (XDG) [5]
- Graph-based Differential Grouping (gDG) [6]
- Fast Interaction Identification [7]
- Recursive Differential Grouping (RDG1 and RDG2) [8]

[1] Masaru Tezuka, Masaharu Munetomo, and Kiyoshi Akama. "Linkage identification by nonlinearity check for real-coded genetic algorithms". In: *Genetic and Evolutionary Computation–GECCO 2004*. Springer. 2004, pp. 222–233.

[2] Mohammad Nabi Omidvar et al. "Cooperative co-evolution with differential grouping for large scale optimization". In: *IEEE Transactions on Evolutionary Computation* 18.3 (2014), pp. 378–393.

[3] Yi Mei et al. "Competitive Divide-and-Conquer Algorithm for Unconstrained Large Scale Black-Box Optimization". In: *ACM Transaction on Mathematical Software* 42.2 (June 2015), p. 13.

[4] Mohammad Nabi Omidvar et al. *IDG: A Faster and More Accurate Differential Grouping Algorithm*. Technical Report CSR-15-04. University of Birmingham, School of Computer Science, Sept. 2015.

[5] Yuan Sun, Michael Kirley, and Saman Kumara Halgamuge. "Extended differential grouping for large scale global optimization with direct and indirect variable interactions". In: *Genetic and Evolutionary Computation Conference*. ACM. 2015, pp. 313–320.

[6] Yingbiao Ling, Haijian Li, and Bin Cao. "Cooperative co-evolution with graph-based differential grouping for large scale global optimization". In: *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*. IEEE. 2016, pp. 95–102.

[7] Xiao-Min Hu et al. "Cooperation coevolution with fast interdependency identification for large scale optimization". In: *Information Sciences* 381 (2017), pp. 142–160.

Shortcomings of Differential Grouping

- Cannot detect the overlapping functions.
- Slow if all interactions are to be checked.
- Requires a threshold parameter (ϵ).
- Can be sensitive to the choice of the threshold parameter (ϵ).

Direct/Indirect Interactions

Indirect Interactions

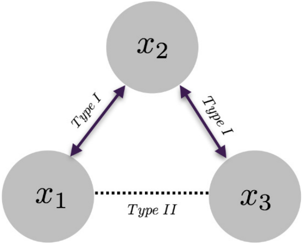
In an objective function $f(\mathbf{x})$, decision variables x_i and x_j interact directly (denoted by $x_i \leftrightarrow x_j$) if

$$\exists \mathbf{a} : \left. \frac{\partial f}{\partial x_i \partial x_j} \right|_{\mathbf{x}=\mathbf{a}} \neq 0,$$

decision variables x_i and x_j interact *indirectly* if

$$\frac{\partial f}{\partial x_i \partial x_j} = 0,$$

and there exists a set of decision variables $\{x_{k1}, ..., x_{ks}\}$ such that $x_i \leftrightarrow x_{l1}, ..., x_{ks} \leftrightarrow x_j$.



Efficiency vs Accuracy

Saving budget at the expense of missing overlaps:

- eXtended Differential Grouping [1].
- Fast Interdependence Identification [2].

Figure: The interaction structures represented by the two graphs cannot be distinguished by XDG.

[1] Yuan Sun, Michael Kirley, and Saman Kumara Halgamuge. "Extended differential grouping for large scale global optimization with direct and indirect variable interactions". In: *Genetic and Evolutionary Computation Conference*. ACM. 2015, pp. 313–320.

[2] Xiao-Min Hu et al. "Cooperation coevolution with fast interdependency identification for large scale optimization". In: *Information Sciences* 381 (2017), pp. 142–160.

Differential Grouping 2: Improving Accuracy [1]

DG2 Estimates the computational round-off errors as the threshold value:

$$e_{\inf} := \gamma_2 \max\{f(\mathbf{x}) + f(\mathbf{y}'), f(\mathbf{y}) + f(\mathbf{x}')\} \tag{7}$$

$$e_{\sup} = \gamma_{\sqrt{n}} \max\{f(\mathbf{x}), f(\mathbf{x}'), f(\mathbf{y}), f(\mathbf{y}')\} \tag{8}$$

- $\lambda < e_{\inf} \rightarrow$ separable;
- $\lambda > e_{\sup} \rightarrow$ non-separable.

Otherwise

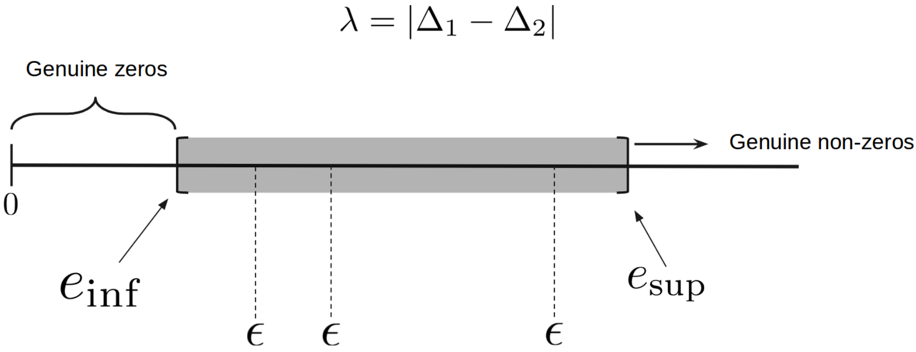
$$\epsilon = \frac{\eta_0}{\eta_0 + \eta_1} e_{\inf} + \frac{\eta_1}{\eta_0 + \eta_1} e_{\sup}, \tag{9}$$

- $\lambda < \epsilon \rightarrow$ separable;
- $\lambda \geq \epsilon \rightarrow$ non-separable.

---

[1] Mohammad Nabi Omidvar et al. "DG2: A faster and more accurate differential grouping for large-scale black-box optimization". In: *IEEE Transactions on Evolutionary Computation* 21.6 (2017), pp. 929–942.

DG2 Error Analysis



Differential Grouping 2: Improving Efficiency

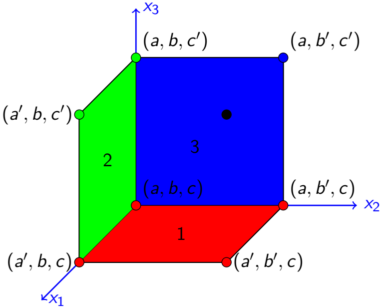


Figure: Geometric representation of point generation in DG2 for a 3D function.

$$x_1 \leftrightarrow x_2 : \Delta^{(1)} = f(a', b, c) - f(a, b, c), \Delta^{(2)} = f(a', b', c) - f(a, b', c)$$

$$x_1 \leftrightarrow x_3 : \Delta^{(1)} = f(a', b, c) - f(a, b, c), \Delta^{(2)} = f(a', b, c') - f(a, b, c')$$

$$x_2 \leftrightarrow x_3 : \Delta^{(1)} = f(a, b', c) - f(a, b, c), \Delta^{(2)} = f(a, b', c') - f(a, b, c'),$$

$$\lambda = |\Delta^{(1)} - \Delta^{(2)}|$$

Differential Grouping 2: Improving Efficiency

Minimum Evaluations

The minimum number of unique function evaluations in order to detect the interactions between all pairs of variables is

$$h(n) \geq \frac{n(n+1)}{2} + 1. \tag{10}$$

Improving efficiency beyond the given lower bound is impossible unless:

- Sacrifice on the accuracy (partial variable interaction matrix);
- and/or
- Extending the DG theorem.

Extending the Theorem: Further Improving Efficiency

1. Fast Interaction Identification (FII) [1]
   - identifying separable variables by checking the interaction between a single variable with remaining variables.
   - examining pairwise interaction for non-separable variables.

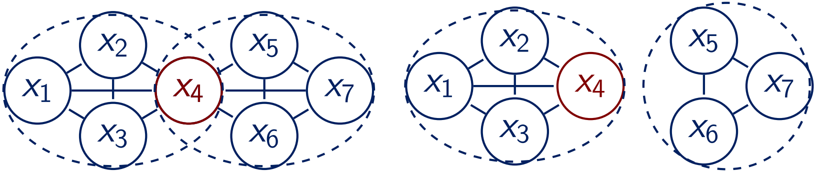
2. Recursive Differential Grouping (RDG) [2] ($\mathcal{O}(n \log(n))$)
   - examining interaction between two variable subsets.
   - using a recursive procedure to group variables.

[1] Xiao-Min Hu et al. "Cooperation coevolution with fast interdependency identification for large scale optimization". In: *Information Sciences* 381 (2017), pp. 142–160.

[2] Yuan Sun, Michael Kirley, and Saman K Halgamuge. "A recursive decomposition method for large scale continuous optimization". In: *IEEE Transactions on Evolutionary Computation* 22.5 (2017), pp. 647–661.

Variants of RDG

1. RDG2 [1]: Combining the efficiency of RDG and accuracy of DG2.
2. RDG3 [2]: extending RDG2 for decomposing overlapping problems.



[1] Yuan Sun et al. "Adaptive threshold parameter estimation with recursive differential grouping for problem decomposition". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2018, pp. 889–896.

[2] Yuan Sun et al. "Decomposition for Large-scale Optimization Problems with Overlapping Components". In: *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE. 2019.

Benchmark Suites

- CEC'2005 Benchmark Suite (non-modular)
- CEC'2008 LSGO Benchmark Suite (non-modular)
- CEC'2010 LSGO Benchmark Suite
- CEC'2013 LSGO Benchmark Suite

Challenges of CC

Main Questions

1. How to decompose the problem?
2. How to allocated resources?
3. How to coordinate?

The Imbalance Problem

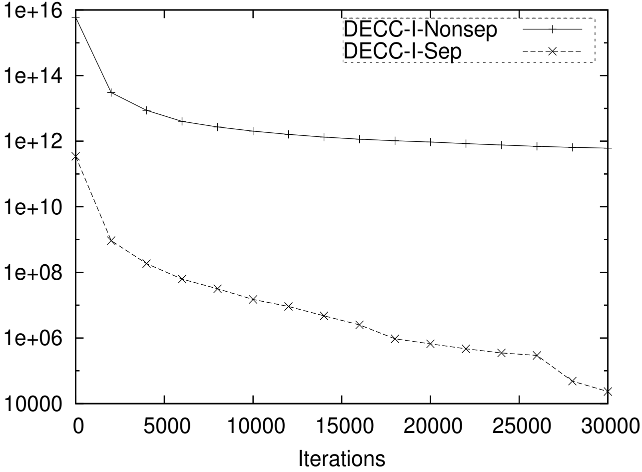
- Non-uniform contribution of components.

Imbalanced Functions

$$f(\mathbf{x}) = \sum_{i=1}^{m} w_i f_i(\mathbf{x}_i), \tag{11}$$

$$w_i = 10^{s\mathcal{N}(0,1)},$$

The Imbalance Problem (2)



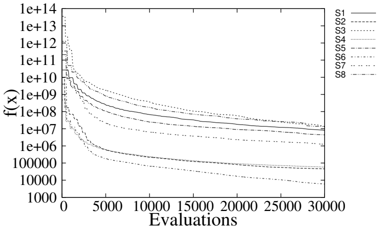
Contribution-Based Cooperative Co-evolution (CBCC)

Types of CC

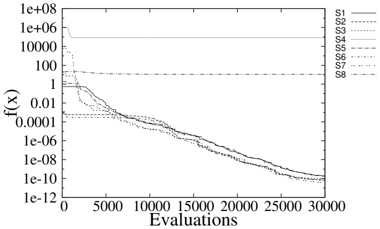
- CC: round-robin optimization of components.
- CBCC: favors components with a higher contribution.
  - Quantifies the contribution of components.
  - Optimizes the one with the highest contribution.

How to Quantify the Contribution

- For quantification of contributions a relatively accurate decomposition is needed.
- Changes in the objective value while other components are kept constant.



(c) Round-Robin CC



(d) Contribution-Based CC

Contribution-Aware Algorithms

- Contribution-Based Cooperative Co-evolution (CBCC) [1], [2].
- Bandit-based Cooperative Coevolution (BBCC) [3].
- Incremental Cooperative Coevolution [4]
- Multilevel Framework for LSGO [5]

[1] Mohammad Nabi Omidvar, Xiaodong Li, and Xin Yao. "Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms". In: *Proc. of Genetic and Evolutionary Computation Conference*. ACM, 2011, pp. 1115–1122.

[2] Mohammad Nabi Omidvar et al. "CBCC3 – A Contribution-Based Cooperative Co-evolutionary Algorithm with Improved Exploration/Exploitation Balance". In: *Proc. IEEE Congr. Evolutionary Computation*. 2016, pp. 3541–3548.

[3] **kazimipour2018bandit**.

[4] Sedigheh Mahdavi, Shahryar Rahnamayan, and Mohammad Ebrahim Shiri. "Incremental cooperative coevolution for large-scale global optimization". In: *Soft Computing* (2016), pp. 1–20.

[5] Sedigheh Mahdavi, Shahryar Rahnamayan, and Mohammad Ebrahim Shiri. "Multilevel framework for large-scale global optimization". In: *Soft Computing* (2016), pp. 1–30.

Some Auxiliary Topics

- Variable Interaction and Constraint Handling [1], [2], [3]
- Large-Scale Multiobjective Optimization
- Available Benchmark Suites

[1] Eman Sayed et al. "Decomposition-based evolutionary algorithm for large scale constrained problems". In: *Information Sciences* 316 (2015), pp. 457–486.

[2] Adan E Aguilar-Justo and Efrén Mezura-Montes. "Towards an improvement of variable interaction identification for large-scale constrained problems". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2016, pp. 4167–4174.

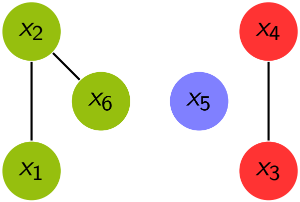
[3] Julien Blanchard, Charlotte Beauthier, and Timoteo Carletti. "A cooperative co-evolutionary algorithm for solving large-scale constrained problems with interaction detection". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2017, pp. 697–704.

Variable Interaction and Constraint Handling [1]

$$
\begin{aligned}
\min \ f(x) &= x_1^2 x_2 + 4x_5 \\
s.t \ g_1(x) &= \frac{x_3}{x_4^2} + \sqrt{x_5} - x_6 \leq 0 \\
g_2(x) &= x_1 - x_2 e^{-x6} \leq 0
\end{aligned}
$$

$$
\Theta_0 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \Theta_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \Theta_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},
$$

$$
\Theta_{glob} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.
$$



[1] Julien Blanchard, Charlotte Beauthier, and Timoteo Carletti. "A cooperative co-evolutionary algorithm for solving large-scale constrained problems with interaction detection". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM. 2017, pp. 697–704.

Large-Scale Multiobjective Optimization

Large-scale multiobjective optimization is growing popularity:

- Benchmark development and analysis:
  - Development of a benchmark [1].
  - Analysis of the existing benchmarks [2].
- Algorithm development:
  - Exploiting modularity using CC [3], [4], [5], [6].
  - Problem transformation [7].

---

[1] Ran Cheng et al. "Test problems for large-scale multiobjective and many-objective optimization". In: *IEEE Transactions on Cybernetics* (2016).

[2] Ke Li et al. "Variable Interaction in Multi-objective Optimization Problems". In: *Parallel Problem Solving from Nature*. Springer International Publishing. 2016, pp. 399–409.

[3] Luis Miguel Antonio and Carlos A Coello Coello. "Use of cooperative coevolution for solving large scale multiobjective optimization problems". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2013, pp. 2758–2765.

[4] Luis Miguel Antonio and Carlos A Coello Coello. "Decomposition-Based Approach for Solving Large Scale Multi-objective Problems". In: *Parallel Problem Solving from Nature*. Springer. 2016, pp. 525–534.

[5] Xiaoliang Ma et al. "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables". In: *IEEE Transactions on Evolutionary Computation* 20.2 (2016), pp. 275–298.

[6] Xingyi Zhang et al. "A Decision Variable Clustering-Based Evolutionary Algorithm for Large-scale Many-objective Optimization". In: *IEEE Transactions on Evolutionary Computation* (2016).

[7] Heiner Zille et al. "A Framework for Large-Scale Multiobjective Optimization Based on Problem Transformation". In: *IEEE Transactions on Evolutionary Computation* 22.2 (2018), pp. 260–275.

Analysis of ZDT

$$\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \end{array}\begin{array}{c}\begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \\ \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array}\end{array}$$

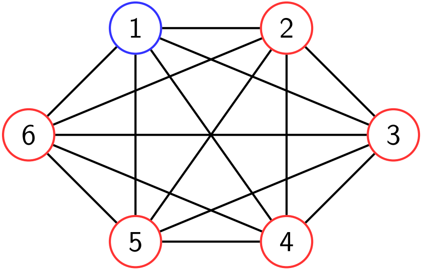


Figure: Variable interaction structures of the f_2 function of ZDT test suite [1].

[1] Ke Li et al. "Variable Interaction in Multi-objective Optimization Problems". In: *Parallel Problem Solving from Nature*. Springer International Publishing. 2016, pp. 399–409.

Analysis of DTLZ1-DTLZ4

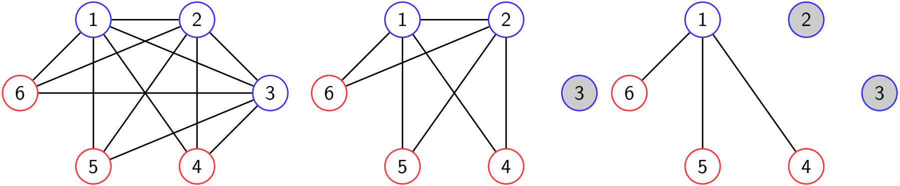


Figure: Variable interaction graphs of DTLZ1 to DTLZ4 .

Proposition 1

For DTLZ1 to DTLZ4, $\forall f_i, i \in \{1, \cdots, m\}$, we divide the corresponding decision variables into two non-overlapping sets: $\mathbf{x}_I = (x_1, \cdots, x_\ell)^T$, $\ell = m - 1$ for $i \in \{1, 2\}$ while $\ell = m - i + 1$ for $i \in \{3, \cdots, m\}$; and $\mathbf{x}_{II} = (x_m, \cdots, x_n)^T$. All members of $\mathbf{x}_I$ not only interact with each other, but also interact with those of $\mathbf{x}_{II}$; all members of $\mathbf{x}_{II}$ are independent from each other.

Analysis of DTLZ5-DTLZ7

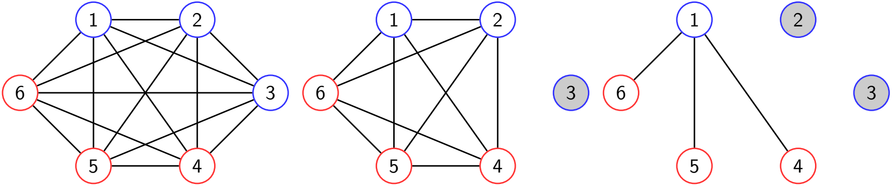


Figure: Variable interaction graphs of DTLZ5 and DTLZ6.

Proposition 2

For DTLZ5 and DTLZ6, $\forall f_i, i \in \{1, \cdots, m\}$, we divide the corresponding decision variables into two non-overlapping sets: $\mathbf{x}_I = (x_1, \cdots, x_\ell)^T$, $\ell = m - 1$ for $i \in \{1, 2\}$ while $\ell = m - i + 1$ for $i \in \{3, \cdots, m\}$; and $\mathbf{x}_{II} = (x_m, \cdots, x_n)^T$. For f_i, where $i \in \{1, \cdots, m-1\}$, all members of $\mathbf{x}_I$ and $\mathbf{x}_{II}$ interact with each other; for f_m, we have the same interaction structure as DTLZ1-DTLZ4.

Proposition 3

All objective functions of DTLZ7 are fully separable.

Decomposition Based Large-Scale EMO

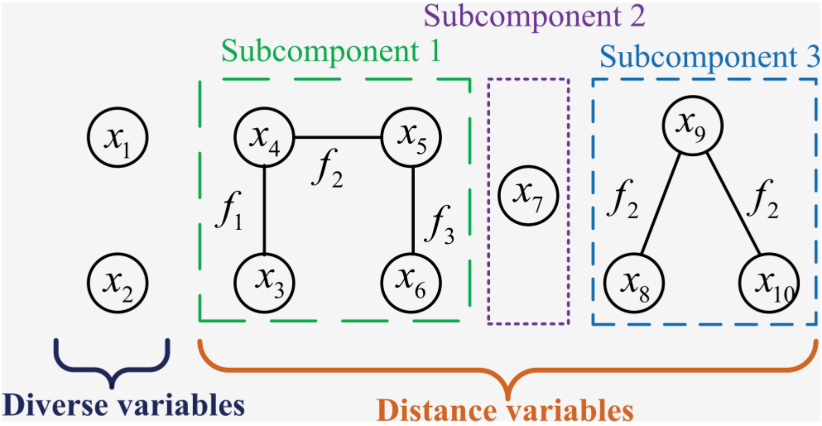


Figure: Image taken from [1]

[1] Xiaoliang Ma et al. "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables". In: *IEEE Transactions on Evolutionary Computation* 20.2 (2016), pp. 275–298.

Weighted Optimization Framework (WOF) [1], [2]

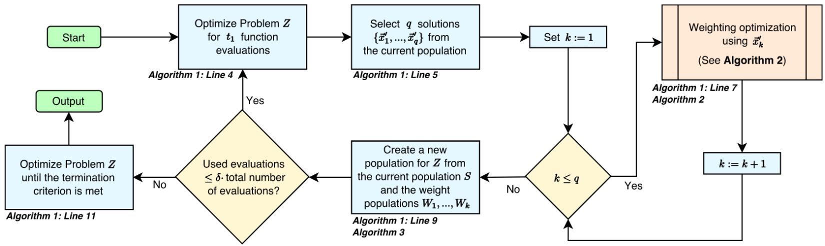


Figure: Weighted Optimization Framework

[1] Heiner Zille et al. "Weighted Optimization Framework for Large-scale Multi-objective Optimization". In: *Genetic and Evolutionary Computation Conference*. ACM. 2016, pp. 83–84.

[2] Heiner Zille et al. "A Framework for Large-Scale Multiobjective Optimization Based on Problem Transformation". In: *IEEE Transactions on Evolutionary Computation* 22.2 (2018), pp. 260–275.

Some Future Directions (I)

- What if the components have overlap?
- Differential group is time-consuming. Is there a more efficient method?
- Do we need to get 100% accurate grouping? What is the relationship between grouping accuracy and optimality achieved by a CC algorithm?

Some Future Directions (II)

- CC for combinatorial optimization, e.g.,
  - Y. Mei, X. Li and X. Yao, "Cooperative Co-evolution with Route Distance Grouping for Large-Scale Capacitated Arc Routing Problems," IEEE Transactions on Evolutionary Computation, 18(3):435-449, June 2014.
- However, every combinatorial optimization problem has its own characteristics. We need to investigate CC for other combinatorial optimization problems.

Some Future Directions (III)

- Learning variable interdependencies is a strength of estimation of distribution algorithms (EDAs), e.g.,
  - W. Dong, T. Chen, P. Tino and X. Yao, "Scaling Up Estimation of Distribution Algorithms for Continuous Optimization," IEEE Transactions on Evolutionary Computation, 17(6):797-822, December 2013.
  - A. Kaban, J. Bootkrajang and R.J. Durrant. "Towards Large Scale Continuous EDA: A Random Matrix Theory Perspective." Evolutionary Computation
- Interestingly, few work exists on scaling up EDAs.

LSGO Resources

- There is an IEEE Computational Intelligence Society (CIS) Task Force on LSGO:
- LSGO Repository: `http://www.cercia.ac.uk/projects/lsgo`

Acknowledgement

Thanks to

- Dr. Yuan Sun from RMIT University, for assistance in revising the slides;
- Dr. Ata Kaban and Dr. Momodou L. Sanyang for allowing us to use some figures from their publications.

Questions

Thanks for your attention!