Adaptive Threshold Parameter Estimation with Recursive Differential Grouping for Problem Decomposition

Yuan Sun[1,3] Mohammad Nabi Omidvar [2] Michael Kirley [1]
Xiaodong Li [3]

[1]School of Computing and Information Systems, University of Melbourne

[2]School of Computer Science, University of Birmingham

[3]School of Science, RMIT University

yuan.sun@unimelb.edu.au
yuan.sun@rmit.edu.au

July 17, 2018

Overview

Introduction: Large-Scale Continuous Optimization

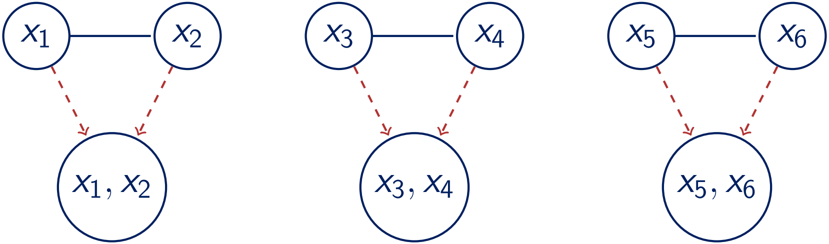
Large-scale (High-dimensional) Continuous Optimization Problems are challenging to solve:

- Search space increases exponentially.
- Problem complexity increases greatly.
- The running time of some evolutionary algorithms increases significantly.

Background: Cooperative Co-evolution[1]

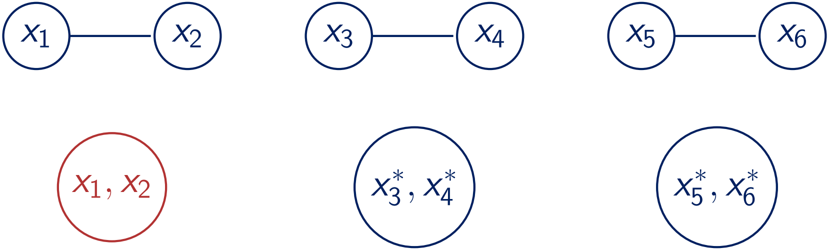
[1]Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1994: 249-257.

Background: Cooperative Co-evolution[1]



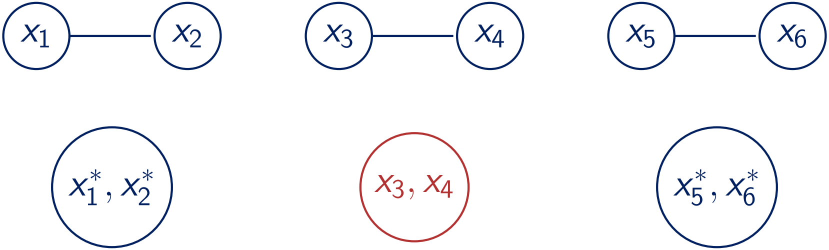
[1] Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1994: 249-257.

Background: Cooperative Co-evolution[1]



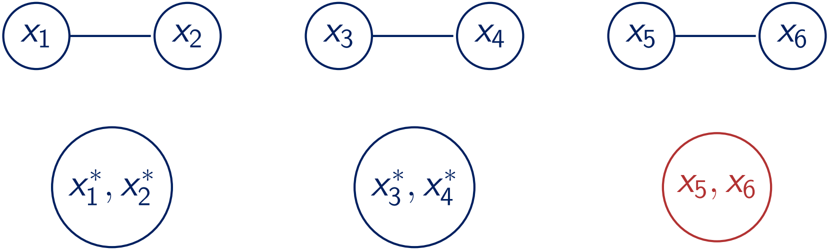
[1] Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1994: 249-257.

Background: Cooperative Co-evolution[1]



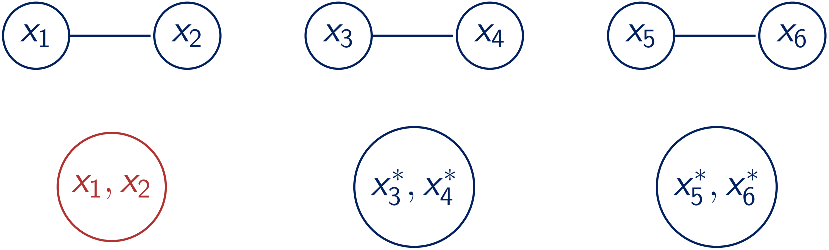
[1] Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1994: 249-257.

Background: Cooperative Co-evolution[1]



[1] Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1994: 249-257.

Background: Cooperative Co-evolution[1]



[1] Potter M A, De Jong K A. A cooperative coevolutionary approach to function optimization[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 1994: 249-257.

Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$

[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

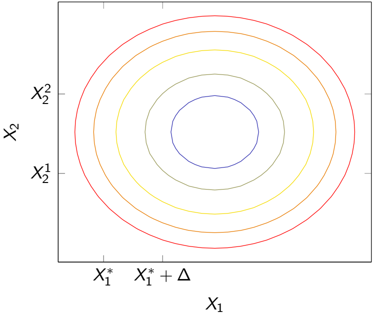
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

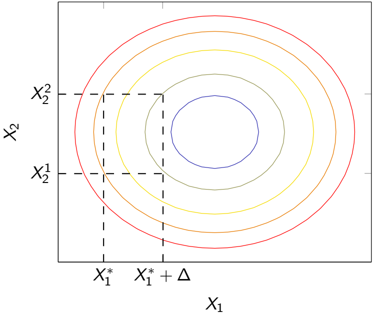
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

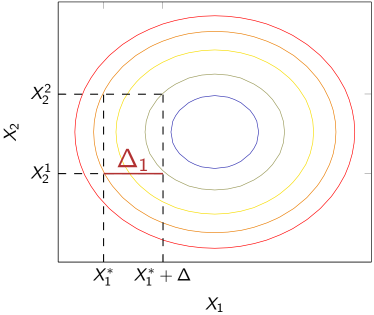
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

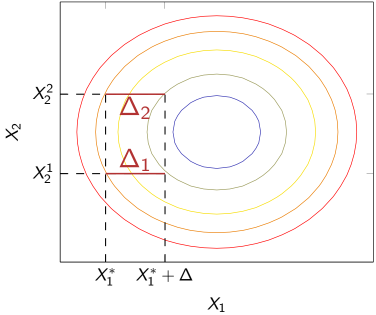
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

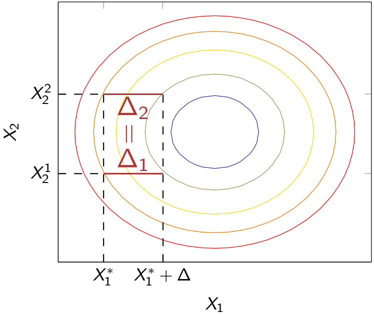
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \tag{1}$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \tag{2}$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

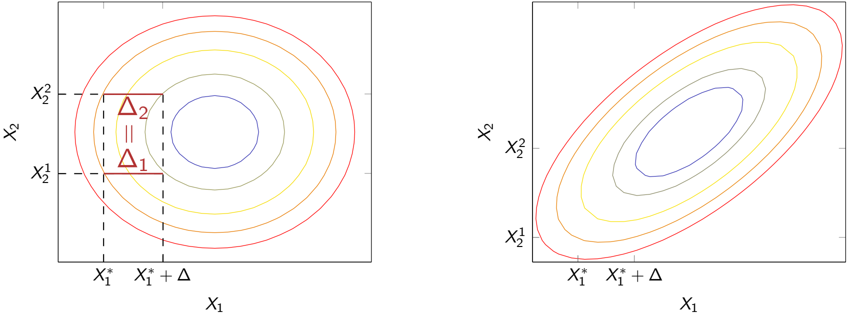
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

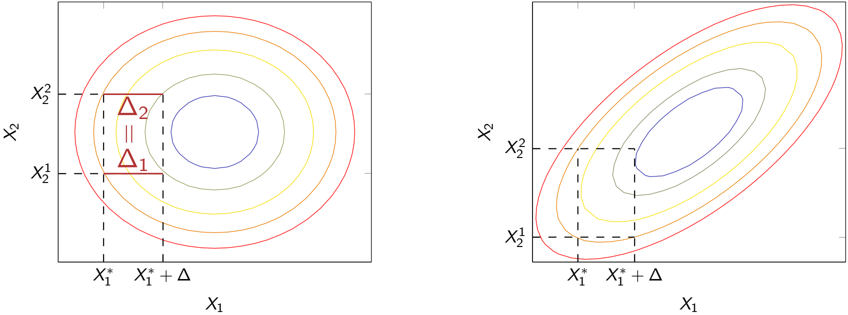
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

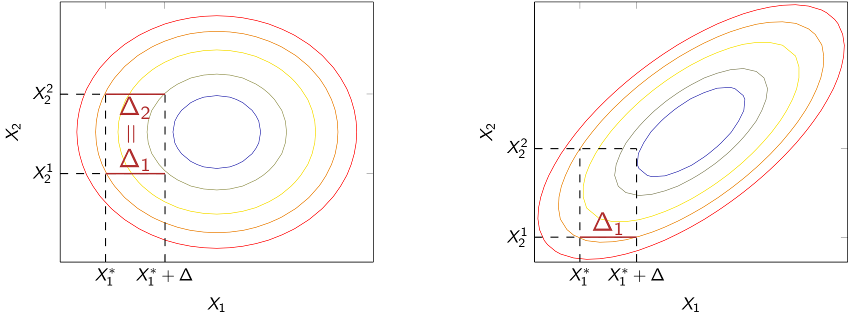
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

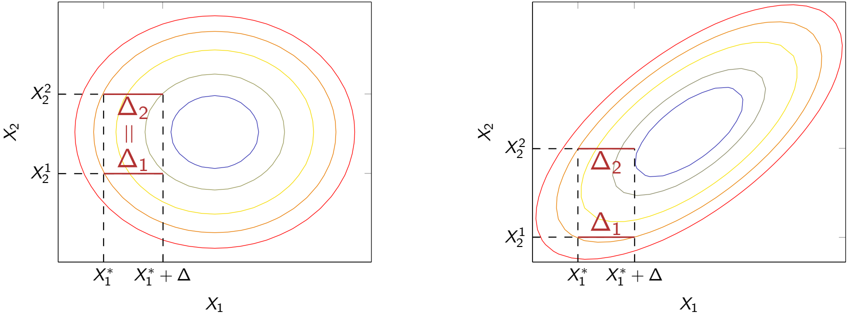
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \quad (1)$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \quad (2)$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

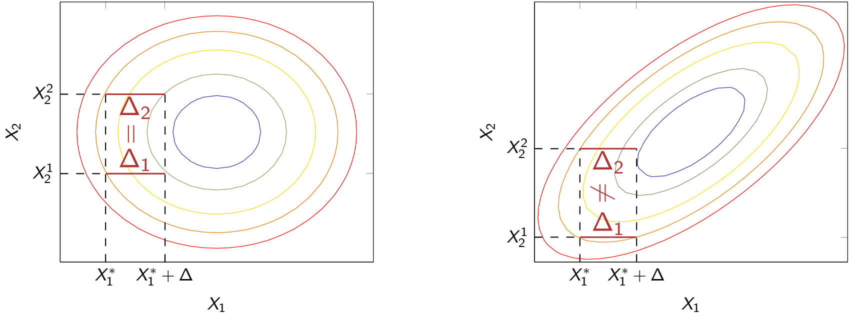
Background: Recursive Differential Grouping (RDG) [2]

There exists some interaction between two subsets of decision variables X_1 and X_2 if

$$\Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^1} \neq \Delta_{X_1} f(\mathbf{x})|_{X_1=X_1^*, X_2=X_2^2}, \tag{1}$$

where

$$\Delta_{X_1} f(\mathbf{x}) = f(\cdots, X_1 + \Delta X_1, \cdots) - f(\cdots, X_1, \cdots). \tag{2}$$



[2] Sun Y, Kirley M, Halgamuge S K. A recursive decomposition method for large scale continuous optimization[J]. IEEE Transactions on Evolutionary Computation, accepted on November 2017

Background: Parameter (Threshold) Setting for RDG

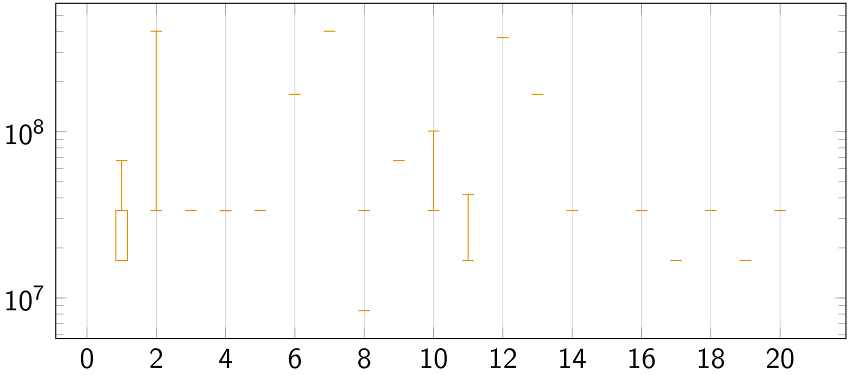
1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.

Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice,

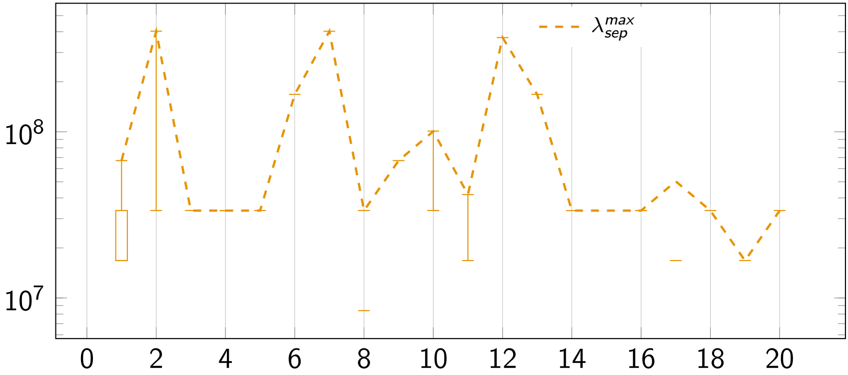
Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice,



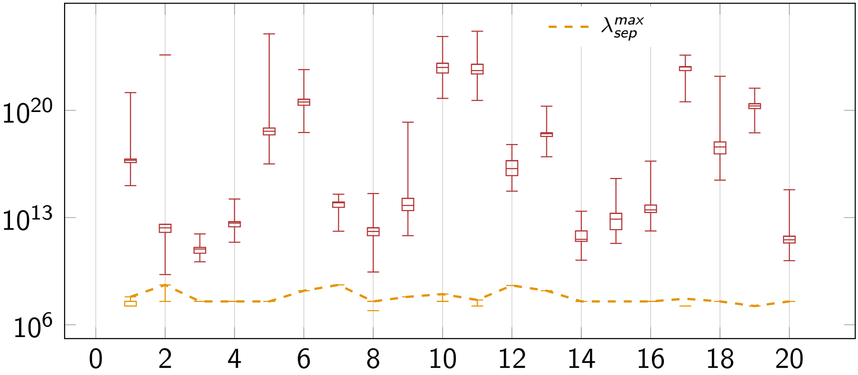
Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice,



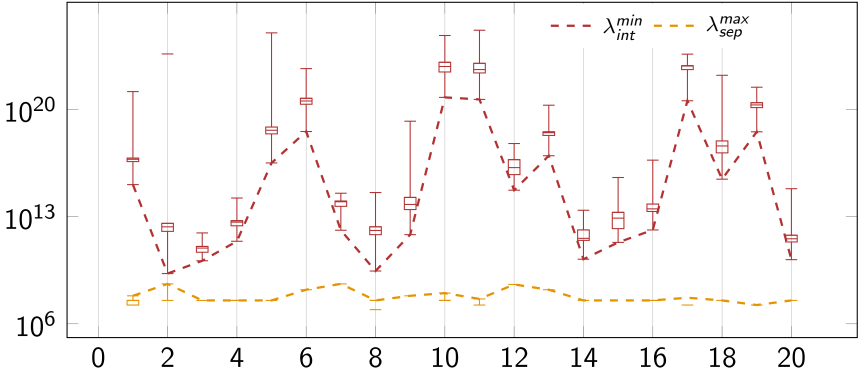
Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice,



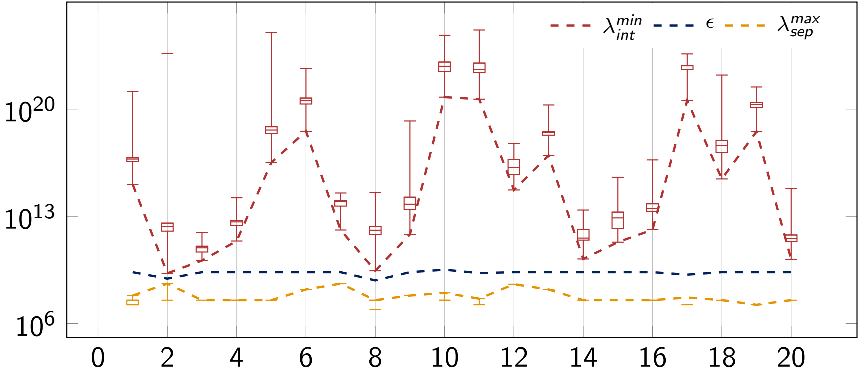
Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice,



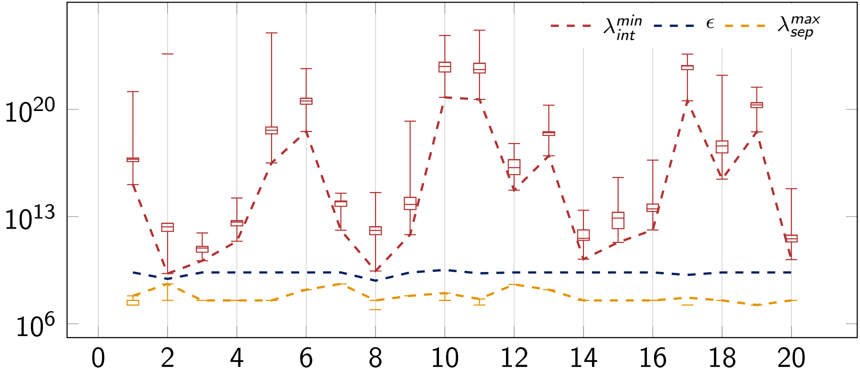
Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice,



Background: Parameter (Threshold) Setting for RDG

1. In theory, if $\lambda = 0$, X_1 and X_2 are separable; if $\lambda > 0$, X_1 and X_2 interact, where $\lambda = |\Delta_1 - \Delta_2|$.
2. In practice, if $\lambda \leq \epsilon$, X_1 and X_2 are separable; if $\lambda > \epsilon$, X_1 and X_2 interact.



Background: Parameter (Threshold) Setting for RDG

The RDG method estimates a threshold value based on the magnitude of the objective values:

$$\epsilon := \alpha \cdot \min \left\{ |f(\mathbf{x}_1)|, \cdots, |f(\mathbf{x}_k)| \right\}, \tag{3}$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_k$ are k randomly generated candidate solutions, and α is the control coefficient [3].

[3] Mei Y, Omidvar M N, Li X, et al. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization[J]. ACM Transactions on Mathematical Software (TOMS), 2016, 42(2): 13.

Background: Parameter (Threshold) Setting for RDG

The RDG method estimates a threshold value based on the magnitude of the objective values:

$$\epsilon := \alpha \cdot \min \left\{ |f(\mathbf{x}_1)|, \cdots, |f(\mathbf{x}_k)| \right\}, \tag{3}$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_k$ are k randomly generated candidate solutions, and α is the control coefficient [3].

Limitations:

1. Lack of theoretical foundation.

[3] Mei Y, Omidvar M N, Li X, et al. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization[J]. ACM Transactions on Mathematical Software (TOMS), 2016, 42(2): 13.

The RDG method estimates a threshold value based on the magnitude of the objective values:

$$\epsilon := \alpha \cdot \min \left\{ |f(\mathbf{x}_1)|, \cdots, |f(\mathbf{x}_k)| \right\}, \tag{3}$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_k$ are k randomly generated candidate solutions, and α is the control coefficient [3].

Limitations:

1. Lack of theoretical foundation.
2. Non-trivial to select an appropriate value for α.

[3] Mei Y, Omidvar M N, Li X, et al. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization[J]. ACM Transactions on Mathematical Software (TOMS), 2016, 42(2): 13.

Background: Parameter (Threshold) Setting for RDG

The RDG method estimates a threshold value based on the magnitude of the objective values:

$$\epsilon := \alpha \cdot \min \left\{ |f(\mathbf{x}_1)|, \cdots, |f(\mathbf{x}_k)| \right\}, \quad (3)$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_k$ are k randomly generated candidate solutions, and α is the control coefficient [3].

Limitations:

1. Lack of theoretical foundation.
2. Non-trivial to select an appropriate value for α.
3. Insufficient to deal with problems with imbalanced components.

[3] Mei Y, Omidvar M N, Li X, et al. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization[J]. ACM Transactions on Mathematical Software (TOMS), 2016, 42(2): 13.

Adaptive Threshold Estimation: Round-off Errors (S1)

The round-off errors involved in the calculation of the non-linearity term $\lambda = \left|\left(f(\mathbf{x}_{l,l}) - f(\mathbf{x}_{u,l})\right) - \left(f(\mathbf{x}_{l,m}) - f(\mathbf{x}_{u,m})\right)\right|$ come from two sources:

[4] $\hat{\Delta}$ denotes the floating-point number of Δ; $\ominus$ denotes floating-point substraction; μ_{M} is a machine dependent constant ($\mu_{\mathrm{M}} = 2^{-53}$ in MATLAB).

Adaptive Threshold Estimation: Round-off Errors (S1)

The round-off errors involved in the calculation of the non-linearity term $\lambda = \left| \left(f(\mathbf{x}_{l,l}) - f(\mathbf{x}_{u,l}) \right) - \left(f(\mathbf{x}_{l,m}) - f(\mathbf{x}_{u,m}) \right) \right|$ come from two sources:

S1: the arithmetic floating-point subtraction between fitness values $f(\mathbf{x})$.

[4] $\hat{\Delta}$ denotes the floating-point number of Δ; $\ominus$ denotes floating-point substraction; μ_{M} is a machine dependent constant ($\mu_{\mathrm{M}} = 2^{-53}$ in MATLAB).

Adaptive Threshold Estimation: Round-off Errors (S1)

The round-off errors involved in the calculation of the non-linearity term $\lambda = \left|\left(f(\mathbf{x}_{l,l}) - f(\mathbf{x}_{u,l})\right) - \left(f(\mathbf{x}_{l,m}) - f(\mathbf{x}_{u,m})\right)\right|$ come from two sources:

S1: the arithmetic floating-point subtraction between fitness values $f(\mathbf{x})$.

S2: the calculation of the fitness values $f(\mathbf{x})$.

[4] $\hat{\Delta}$ denotes the floating-point number of Δ; $\ominus$ denotes floating-point substraction; μ_{M} is a machine dependent constant ($\mu_{\mathrm{M}} = 2^{-53}$ in MATLAB).

Adaptive Threshold Estimation: Round-off Errors (S1)

The round-off errors involved in the calculation of the non-linearity term $\lambda = \left|\big(f(\mathbf{x}_{l,l}) - f(\mathbf{x}_{u,l})\big) - \big(f(\mathbf{x}_{l,m}) - f(\mathbf{x}_{u,m})\big)\right|$ come from two sources:

S1: the arithmetic floating-point subtraction between fitness values $f(\mathbf{x})$.

S2: the calculation of the fitness values $f(\mathbf{x})$.

Round-off Errors (S1):

$$\hat{\Delta}_1 = \hat{f}(\mathbf{x}_{l,l}) \ominus \hat{f}(\mathbf{x}_{u,l}) = \big(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\big)(1+\delta_1), \text{ where } |\delta_1| < \mu_{\mathrm{M}};^{4} \tag{4}$$

[4] $\hat{\Delta}$ denotes the floating-point number of Δ; $\ominus$ denotes floating-point substraction; μ_{M} is a machine dependent constant ($\mu_{\mathrm{M}} = 2^{-53}$ in MATLAB).

Adaptive Threshold Estimation: Round-off Errors (S1)

The round-off errors involved in the calculation of the non-linearity term $\lambda = \left|\left(f(\mathbf{x}_{l,l}) - f(\mathbf{x}_{u,l})\right) - \left(f(\mathbf{x}_{l,m}) - f(\mathbf{x}_{u,m})\right)\right|$ come from two sources:

S1: the arithmetic floating-point subtraction between fitness values $f(\mathbf{x})$.

S2: the calculation of the fitness values $f(\mathbf{x})$.

Round-off Errors (S1):

$$\hat{\Delta}_1 = \hat{f}(\mathbf{x}_{l,l}) \ominus \hat{f}(\mathbf{x}_{u,l}) = \left(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\right)(1+\delta_1), \text{ where } |\delta_1| < \mu_{\mathrm{M}}; \tag{4}$$

[4]

$$\hat{\Delta}_2 = \hat{f}(\mathbf{x}_{l,m}) \ominus \hat{f}(\mathbf{x}_{u,m}) = \left(\hat{f}(\mathbf{x}_{l,m}) - \hat{f}(\mathbf{x}_{u,m})\right)(1+\delta_2), \text{ where } |\delta_2| < \mu_{\mathrm{M}}; \tag{5}$$

[4] $\hat{\Delta}$ denotes the floating-point number of Δ; $\ominus$ denotes floating-point substraction; μ_{M} is a machine dependent constant ($\mu_{\mathrm{M}} = 2^{-53}$ in MATLAB).

Adaptive Threshold Estimation: Round-off Errors (S1)

The round-off errors involved in the calculation of the non-linearity term $\lambda = \left|\big(f(\mathbf{x}_{l,l}) - f(\mathbf{x}_{u,l})\big) - \big(f(\mathbf{x}_{l,m}) - f(\mathbf{x}_{u,m})\big)\right|$ come from two sources:

S1: the arithmetic floating-point subtraction between fitness values $f(\mathbf{x})$.

S2: the calculation of the fitness values $f(\mathbf{x})$.

Round-off Errors (S1):

$$\hat{\Delta}_1 = \hat{f}(\mathbf{x}_{l,l}) \ominus \hat{f}(\mathbf{x}_{u,l}) = \big(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\big)(1+\delta_1), \text{ where } |\delta_1| < \mu_{\mathrm{M}};^{4} \quad (4)$$

$$\hat{\Delta}_2 = \hat{f}(\mathbf{x}_{l,m}) \ominus \hat{f}(\mathbf{x}_{u,m}) = \big(\hat{f}(\mathbf{x}_{l,m}) - \hat{f}(\mathbf{x}_{u,m})\big)(1+\delta_2), \text{ where } |\delta_2| < \mu_{\mathrm{M}}; \quad (5)$$

$$\begin{aligned}\hat{\lambda} = \left|\hat{\Delta}_1 \ominus \hat{\Delta}_2\right| = \left|(\hat{\Delta}_1 - \hat{\Delta}_2)(1+\delta_3)\right| = \Big|\big(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\big)(1+\delta_1)(1+\delta_3) \\ - \big(\hat{f}(\mathbf{x}_{l,m}) - \hat{f}(\mathbf{x}_{u,m})\big)(1+\delta_2)(1+\delta_3)\Big|, \text{ where } |\delta_1|,\ |\delta_2|,\ |\delta_3| < \mu_{\mathrm{M}}.\end{aligned} \quad (6)$$

---

[4] $\hat{\Delta}$ denotes the floating-point number of Δ; $\ominus$ denotes floating-point substraction; μ_{M} is a machine dependent constant ($\mu_{\mathrm{M}} = 2^{-53}$ in MATLAB).

Adaptive Threshold Estimation: Round-off Errors (S1)

Theorem

Given a floating-point number system that satisfies IEEE 754 Standard such that $|\delta_i| < \mu_{\mathrm{M}}$*, and* $k\mu_{\mathrm{M}} < 1$*, we have:*

$$\prod_{i=1}^{k}(1+\delta_i)^{e_i} = 1+\theta_k, \text{ where } |\theta_k| \le \frac{k\mu_{\mathrm{M}}}{1-k\mu_{\mathrm{M}}} := \gamma_k \text{ and } e_i = \pm 1.^{a} \quad (7)$$

[a] Corless R M, Fillion N. A graduate introduction to numerical methods[J]. AMC, 2013, 10: 12, Springer.

Example: $(1+\delta_1)(1+\delta_3) = (1+\theta_2)$, where $|\theta_2| \le \gamma_2$.

Adaptive Threshold Estimation: Round-off Errors (S1)

Theorem

Given a floating-point number system that satisfies IEEE 754 Standard such that $|\delta_i| < \mu_{\mathrm{M}}$, *and* $k\mu_{\mathrm{M}} < 1$, *we have:*

$$\prod_{i=1}^{k}(1+\delta_i)^{e_i} = 1+\theta_k, \text{ where } |\theta_k| \le \frac{k\mu_{\mathrm{M}}}{1-k\mu_{\mathrm{M}}} := \gamma_k \text{ and } e_i = \pm 1.^{a} \quad (7)$$

[a] Corless R M, Fillion N. A graduate introduction to numerical methods[J]. AMC, 2013, 10: 12, Springer.

Example: $(1+\delta_1)(1+\delta_3) = (1+\theta_2)$, where $|\theta_2| \le \gamma_2$.

Estimating an upper bound for S1:

$$\hat{\lambda} = \left|\left(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\right)(1+\theta_2) - \left(\hat{f}(\mathbf{x}_{l,m}) - \hat{f}(\mathbf{x}_{u,m})\right)(1+\theta_2')\right|, \quad (8)$$

where $|\theta_2| \le \gamma_2$ and $|\theta_2'| \le \gamma_2$.

Adaptive Threshold Estimation: Round-off Errors (S2)

Assumption 1: The number of floating-point operations (Φ) involved in the calculation of a black-box objective function is in the order of $\Theta(n)$, where n is the dimensionality of the objective function[5] :

$$\Phi \approx n. \tag{9}$$

[5]Omidvar M N, Yang M, Mei Y, et al. DG2: A faster and more accurate differential grouping for large-scale black-box optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(6): 929-942.

[6]Higham N J. Accuracy and stability of numerical algorithms[M]. SIAM, 2002.

Adaptive Threshold Estimation: Round-off Errors (S2)

Assumption 1: The number of floating-point operations (Φ) involved in the calculation of a black-box objective function is in the order of $\Theta(n)$, where n is the dimensionality of the objective function[5] :

$$\Phi \approx n. \tag{9}$$

Assumption 2: The round-off error grows with the square root of the number of floating-point operations (Φ) involved in a calculation[6]:

$$k \approx \sqrt{\Phi}. \tag{10}$$

[5] Omidvar M N, Yang M, Mei Y, et al. DG2: A faster and more accurate differential grouping for large-scale black-box optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(6): 929-942.

[6] Higham N J. Accuracy and stability of numerical algorithms[M]. SIAM, 2002.

Adaptive Threshold Estimation: Round-off Errors (S2)

Assumption 1: The number of floating-point operations (Φ) involved in the calculation of a black-box objective function is in the order of $\Theta(n)$, where n is the dimensionality of the objective function[5] :

$$\Phi \approx n. \tag{9}$$

Assumption 2: The round-off error grows with the square root of the number of floating-point operations (Φ) involved in a calculation[6]:

$$k \approx \sqrt{\Phi}. \tag{10}$$

Estimating an upper bound for S2:

$$\hat{f}(\mathbf{x}) = (1 + \theta_{\sqrt{n}}) f(\mathbf{x}), \text{ where } \left|\theta_{\sqrt{n}}\right| \leq \gamma_{\sqrt{n}}. \tag{11}$$

[5] Omidvar M N, Yang M, Mei Y, et al. DG2: A faster and more accurate differential grouping for large-scale black-box optimization[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(6): 929-942.

[6] Higham N J. Accuracy and stability of numerical algorithms[M]. SIAM, 2002.

Adaptive Threshold Estimation: An Upper Bound

Theorem

Under Assumption 1 and Assumption 2, an upper bound on the round-off errors associated with the calculation of the non-linearity term λ *is given by*

$$|\lambda - \hat{\lambda}| \leq \gamma_{\sqrt{n}+2}\Big(|f(\mathbf{x}_{l,l})| + |f(\mathbf{x}_{u,l})| + |f(\mathbf{x}_{l,m})| + |f(\mathbf{x}_{u,m})|\Big). \quad (12)$$

Adaptive Threshold Estimation: An Upper Bound

Theorem

Under Assumption 1 and Assumption 2, an upper bound on the round-off errors associated with the calculation of the non-linearity term λ is given by

$$|\lambda - \hat{\lambda}| \leq \gamma_{\sqrt{n}+2}\Big(|f(\mathbf{x}_{l,l})| + |f(\mathbf{x}_{u,l})| + |f(\mathbf{x}_{l,m})| + |f(\mathbf{x}_{u,m})|\Big). \qquad (12)$$

Proof.

Substitute $\hat{f}(\mathbf{x}) = (1 + \theta_{\sqrt{n}})f(\mathbf{x})$ into

$$\hat{\lambda} = \Big|\Big(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\Big)(1 + \theta_2) - \Big(\hat{f}(\mathbf{x}_{l,m}) - \hat{f}(\mathbf{x}_{u,m})\Big)(1 + \theta_2')\Big|. \qquad (13)$$

□

Adaptive Threshold Estimation: An Upper Bound

Theorem

Under Assumption 1 and Assumption 2, an upper bound on the round-off errors associated with the calculation of the non-linearity term λ *is given by*

$$|\lambda - \hat{\lambda}| \leq \gamma_{\sqrt{n}+2}\big(|f(\mathbf{x}_{l,l})| + |f(\mathbf{x}_{u,l})| + |f(\mathbf{x}_{l,m})| + |f(\mathbf{x}_{u,m})|\big). \quad (12)$$

Proof.

Substitute $\hat{f}(\mathbf{x}) = (1 + \theta_{\sqrt{n}})f(\mathbf{x})$ into

$$\hat{\lambda} = \Big|\big(\hat{f}(\mathbf{x}_{l,l}) - \hat{f}(\mathbf{x}_{u,l})\big)(1 + \theta_2) - \big(\hat{f}(\mathbf{x}_{l,m}) - \hat{f}(\mathbf{x}_{u,m})\big)(1 + \theta_2')\Big|. \quad (13)$$

□

Adaptive Threshold:

$$\epsilon := \gamma_{\sqrt{n}+2}\big(|f(\mathbf{x}_{l,l})| + |f(\mathbf{x}_{u,l})| + |f(\mathbf{x}_{l,m})| + |f(\mathbf{x}_{u,m})|\big). \quad (14)$$

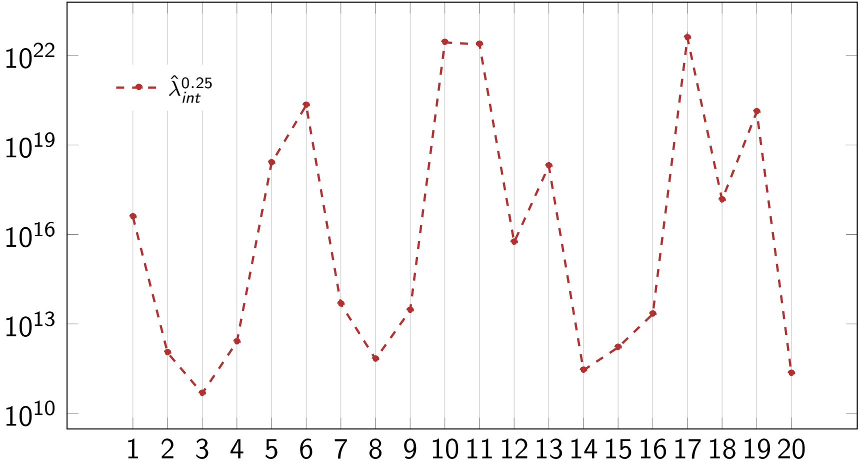
Variables are regarded as interacting if $\hat{\lambda} > \epsilon$, and separable if $\hat{\lambda} \leq \epsilon$.

Experimental Results: Decomposition Comparison

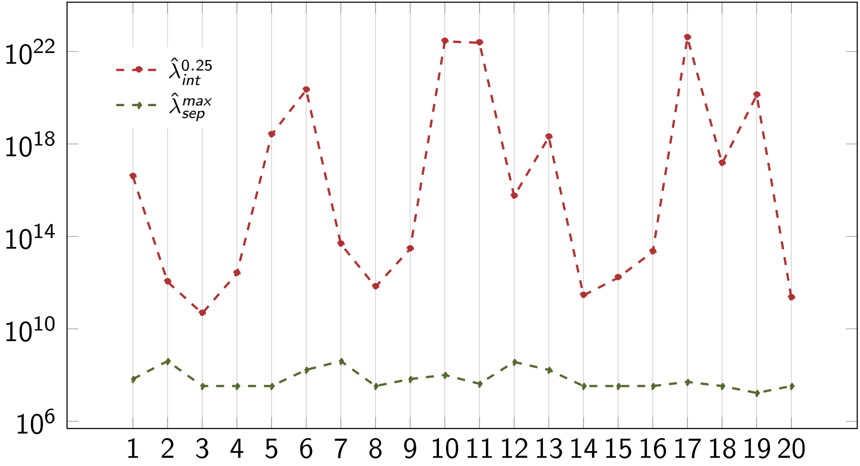
Table: The decomposition results of the RDG2, RDG (with $\alpha = 10^{-12}$) and DG2 methods when used to decompose the CEC'2013 benchmark problems. "*a*" denotes the decomposition accuracy; "FEs" denotes the function evaluations used.

Func	RDG2		RDG ($\alpha = 10^{-12}$)		DG2	
ID	*a*	FEs	*a*	FEs	*a*	FEs
f_7	100%	9.81e+03	100%	9.82e+03	83.3%	5.00e+05
f_8	80.0%	1.91e+04	80.0%	1.95e+04	78.5%	5.00e+05
f_{10}	100%	1.93e+04	82.7%	1.91e+04	100%	5.00e+05
f_{11}	100%	1.93e+04	10.0%	1.06e+04	100%	5.00e+05

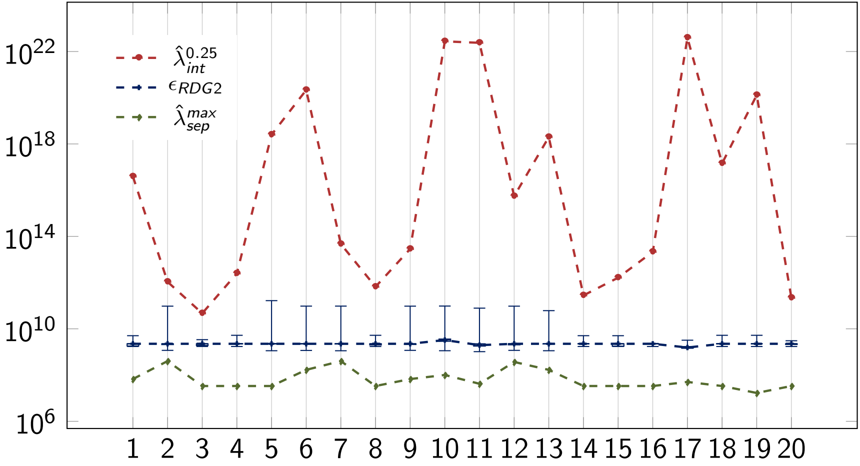
Experimental Results: Decomposition Details (f_{11})



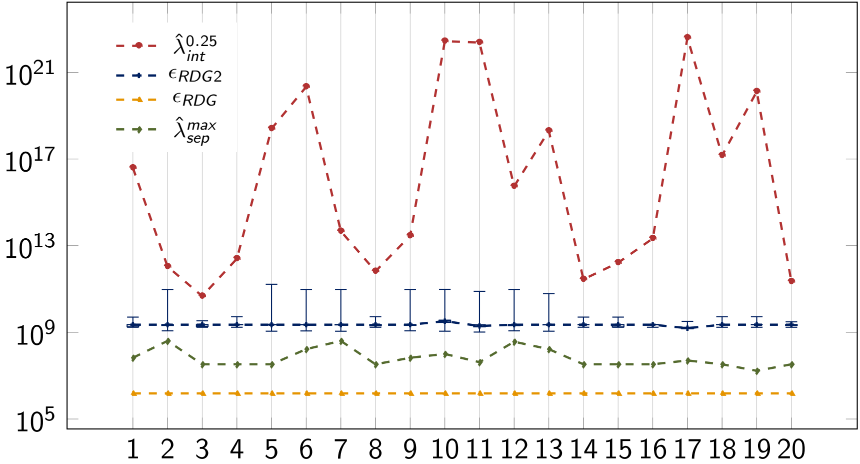
Experimental Results: Decomposition Details (f_{11})



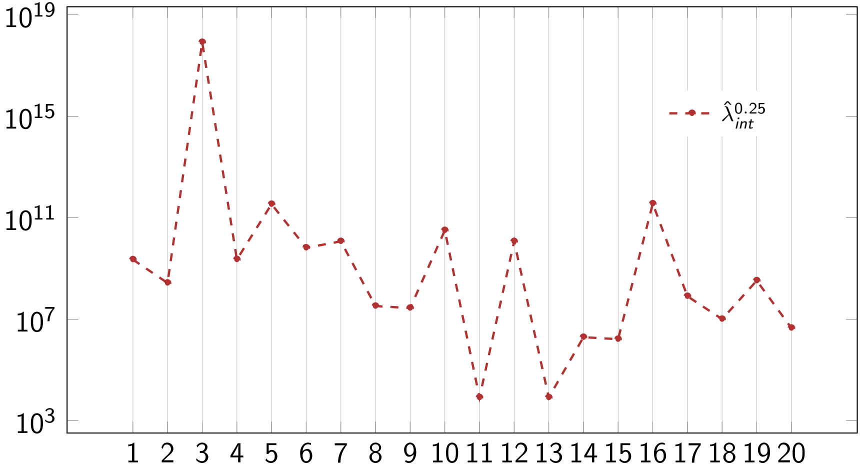
Experimental Results: Decomposition Details (f_{11})



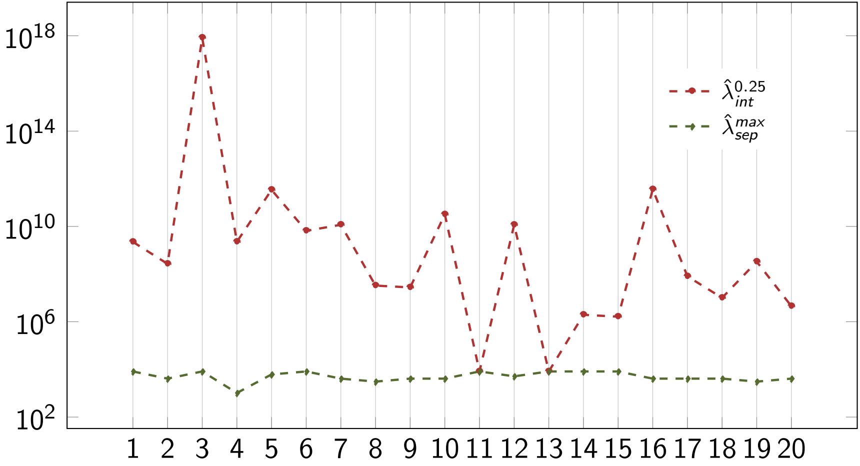
Experimental Results: Decomposition Details (f_{11})



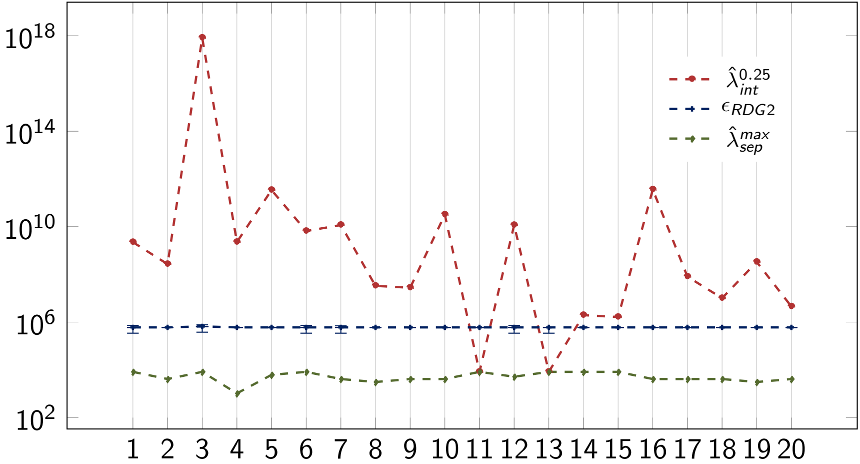
Experimental Results: Decomposition Details (f_8)



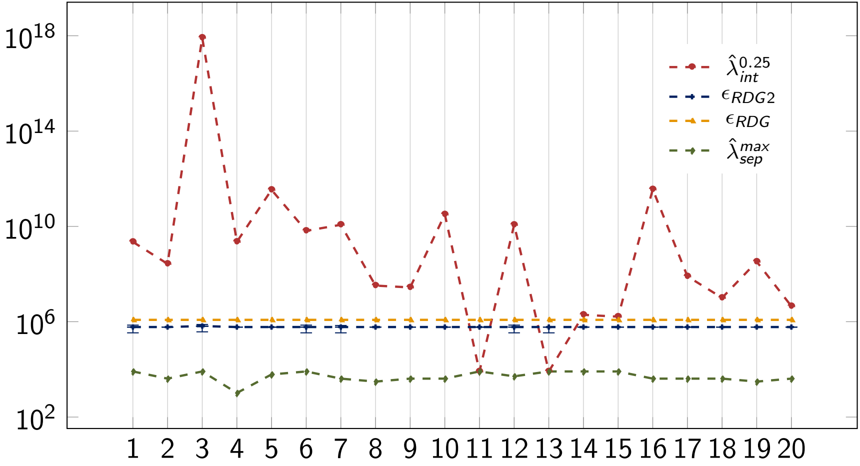
Experimental Results: Decomposition Details (f_8)



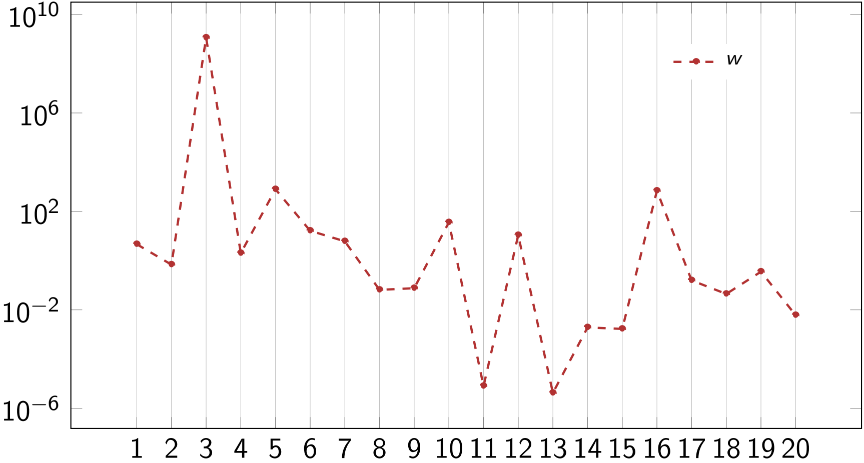
Experimental Results: Decomposition Details (f_8)



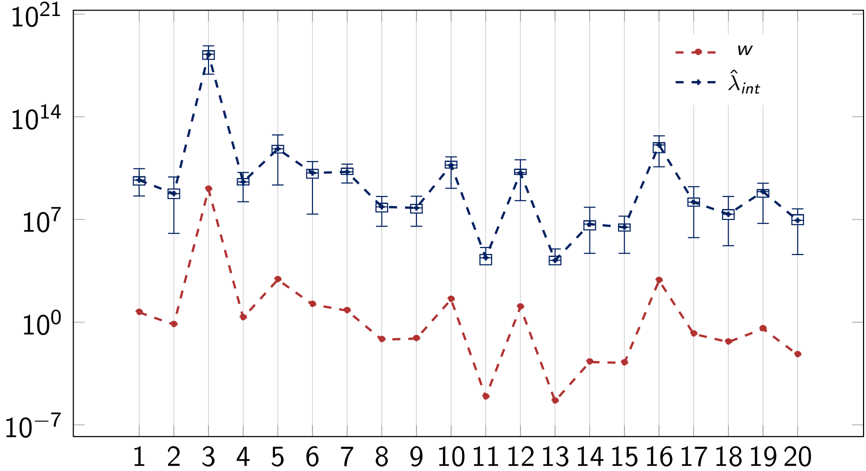
Experimental Results: Decomposition Details (f_8)



Experimental Results: Decomposition Details (f_8)



Experimental Results: Decomposition Details (f_8)



Experimental Results: Optimization Comparison

Table: The optimization results of RDG2, RDG and DG2 when embedded into a CC framework to solve CEC'2013 benchmark problems (Wilcoxon rank-sum tests).

Func	Stats	RDG2	RDG	DG2
f_7	median	3.12e-19	2.93e-20	1.00e+03
	mean	4.04e-16	8.11e-17	1.05e+03
	std	1.48e-15	2.17e-16	2.78e+02
f_8	median	8.15e+06	8.26e+06	3.56e+07
	mean	8.70e+06	8.50e+06	3.84e+07
	std	3.61e+06	2.91e+06	1.08e+07
f_{10}	median	9.05e+07	9.05e+07	9.05e+07
	mean	9.10e+07	9.10e+07	9.13e+07
	std	1.30e+06	1.29e+06	1.50e+06
f_{11}	median	2.81e+03	1.68e+07	1.55e+05
	mean	8.68e+03	1.67e+07	2.47e+05
	std	1.24e+04	1.61e+06	2.36e+05

Experimental Results: Optimization Comparison

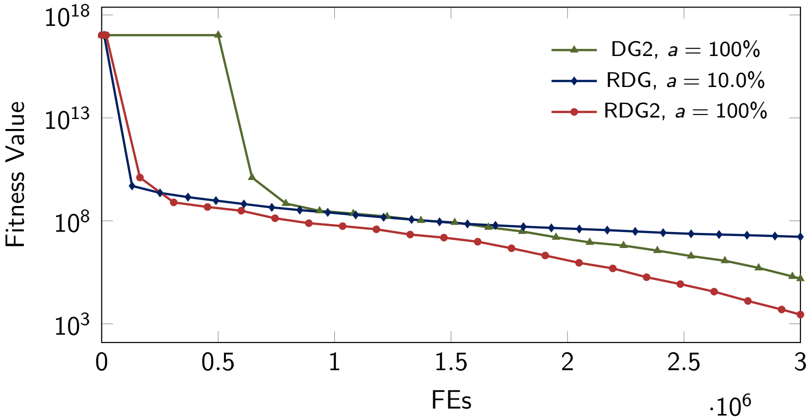


Figure: The convergence curves of the RDG2, RDG and DG2 methods when embedded into the CC framework to solve the CEC'2013 f_{11}.

Conclusion and Future Work

1. Conclusion

   - Derived an upper bound on the computational round-off errors involved in calculating the non-linearity term for RDG.

Conclusion and Future Work

1. Conclusion

   - Derived an upper bound on the computational round-off errors involved in calculating the non-linearity term for RDG.
   - Showed that the upper bound was able to be used as the threshold value to identify variable interactions across a wide range of benchmark problems.

Conclusion and Future Work

1. Conclusion
   - Derived an upper bound on the computational round-off errors involved in calculating the non-linearity term for RDG.
   - Showed that the upper bound was able to be used as the threshold value to identify variable interactions across a wide range of benchmark problems.

2. Future Work
   - Systematically investigate the correlation between the non-linearity term for interacting variables and the weight of the components.

Conclusion and Future Work

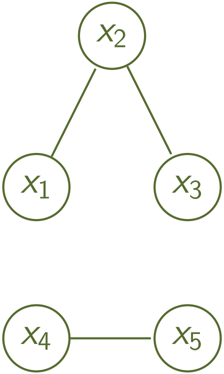
1. Conclusion
   - Derived an upper bound on the computational round-off errors involved in calculating the non-linearity term for RDG.
   - Showed that the upper bound was able to be used as the threshold value to identify variable interactions across a wide range of benchmark problems.

2. Future Work
   - Systematically investigate the correlation between the non-linearity term for interacting variables and the weight of the components.
   - Generate a more effective decomposition for large-scale problems with overlapping components.

Thank You! & Questions?

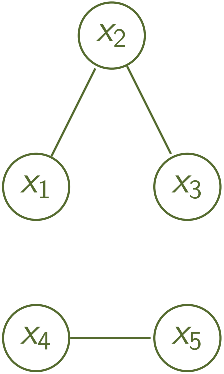
Back-up: Decomposition Process of RDG

Interaction Structure



Back-up: Decomposition Process of RDG

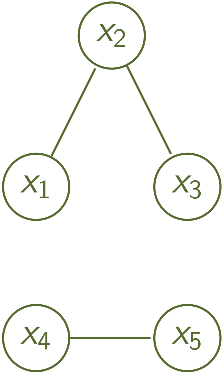
Interaction Structure



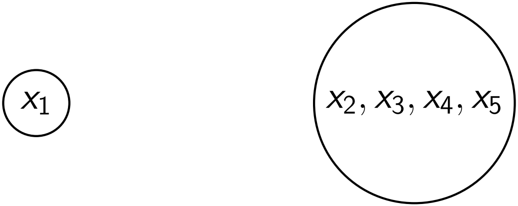
Decomposition Process

Back-up: Decomposition Process of RDG

Interaction Structure

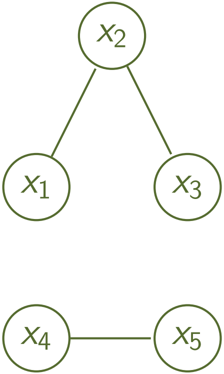


Decomposition Process

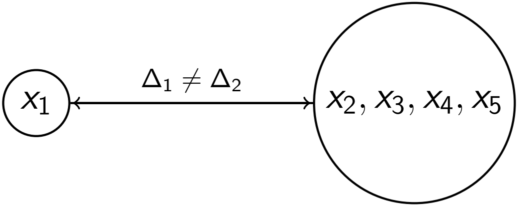


Back-up: Decomposition Process of RDG

Interaction Structure

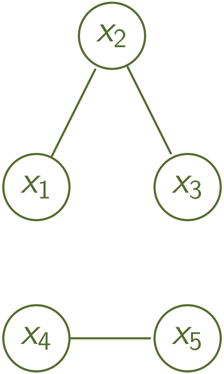


Decomposition Process

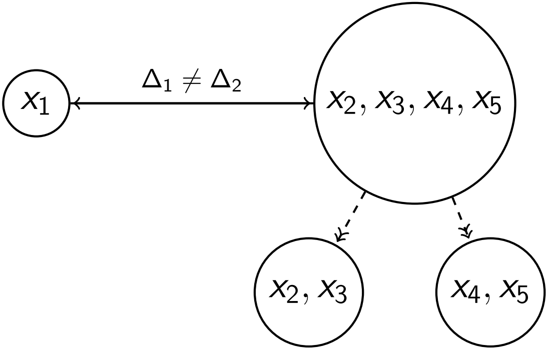


Back-up: Decomposition Process of RDG

Interaction Structure

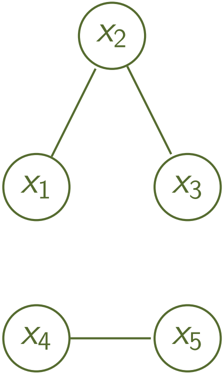


Decomposition Process

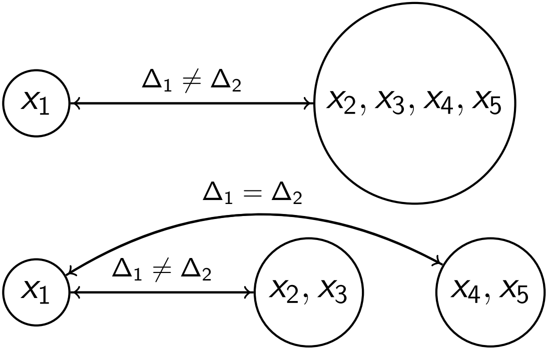


Back-up: Decomposition Process of RDG

Interaction Structure

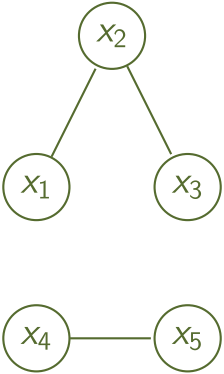


Decomposition Process

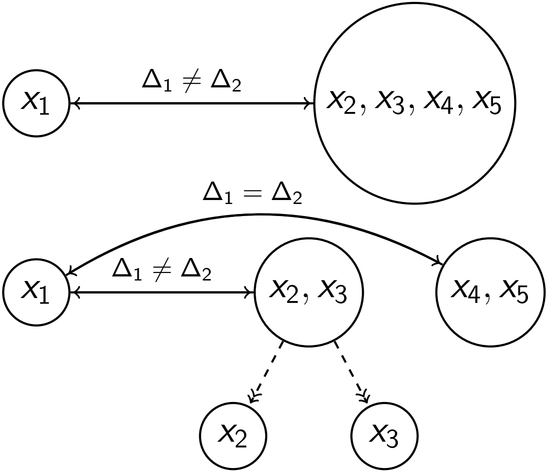


Back-up: Decomposition Process of RDG

Interaction Structure

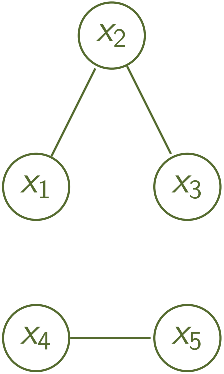


Decomposition Process

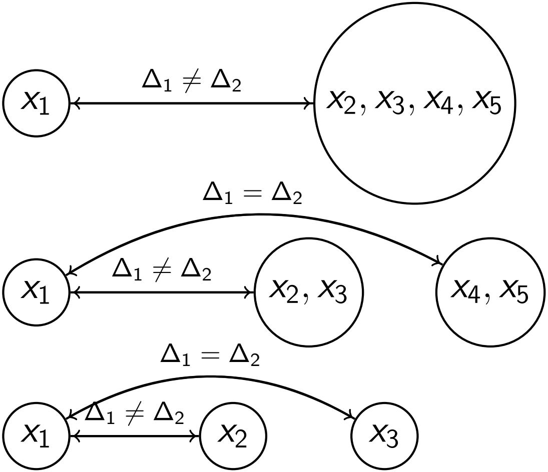


Back-up: Decomposition Process of RDG

Interaction Structure

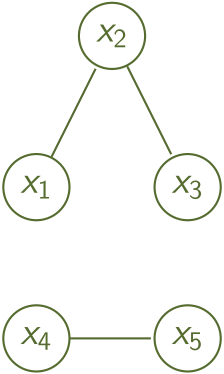


Decomposition Process



Back-up: Decomposition Process of RDG

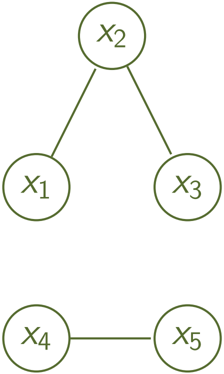
Interaction Structure



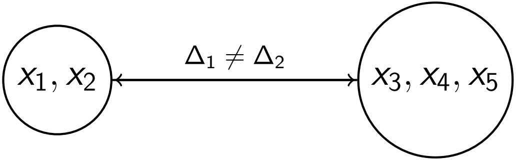
Decomposition Process

Back-up: Decomposition Process of RDG

Interaction Structure

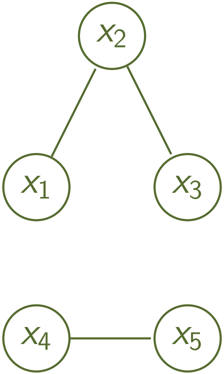


Decomposition Process

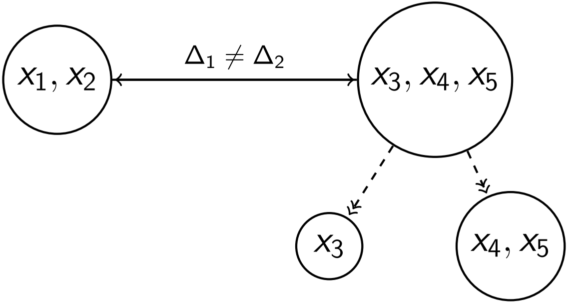


Back-up: Decomposition Process of RDG

Interaction Structure

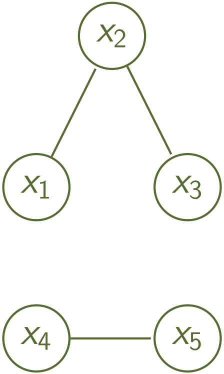


Decomposition Process

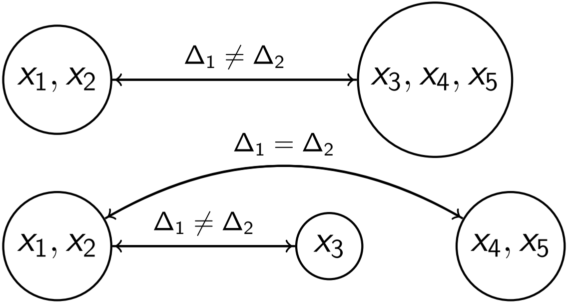


Back-up: Decomposition Process of RDG

Interaction Structure

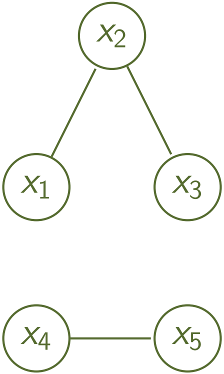


Decomposition Process

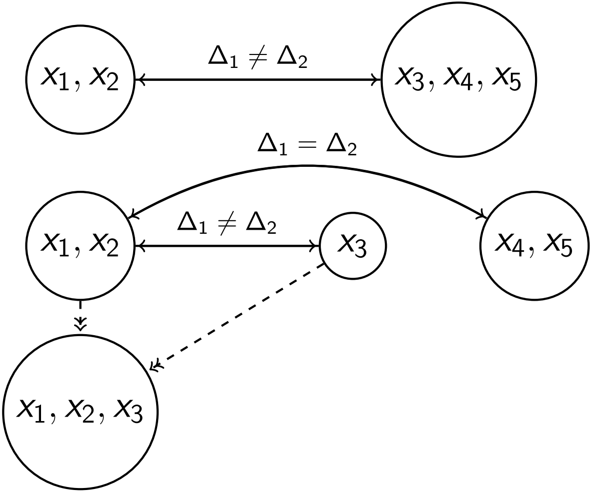


Back-up: Decomposition Process of RDG

Interaction Structure

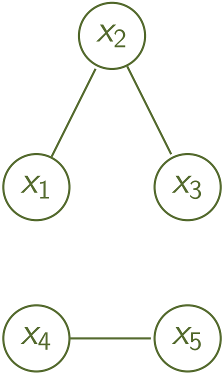


Decomposition Process

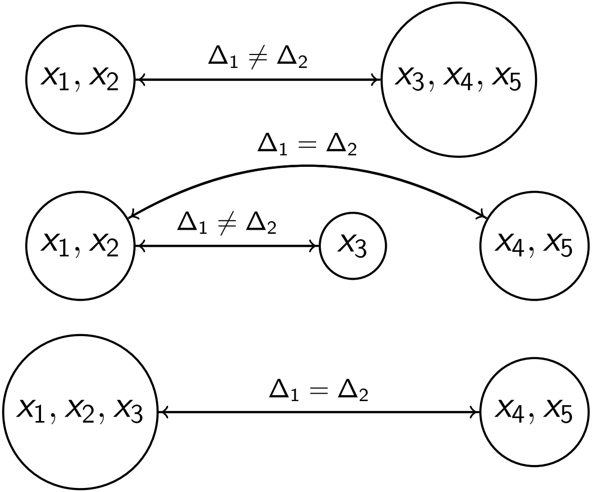


Back-up: Decomposition Process of RDG

Interaction Structure



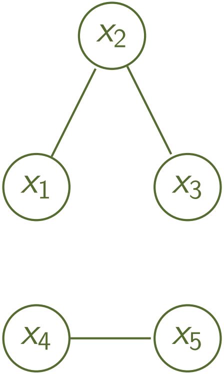
Decomposition Process

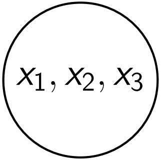


Back-up: Decomposition Process of RDG

Interaction Structure

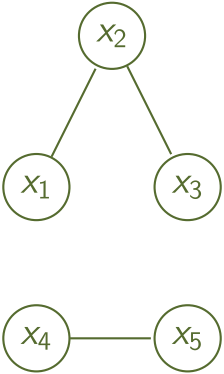
Decomposition Process



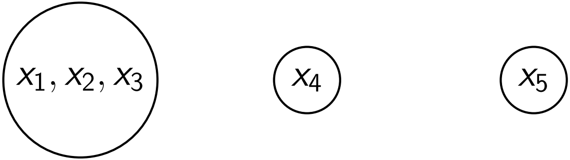


Back-up: Decomposition Process of RDG

Interaction Structure

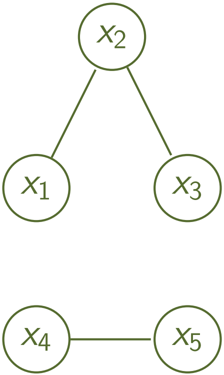


Decomposition Process

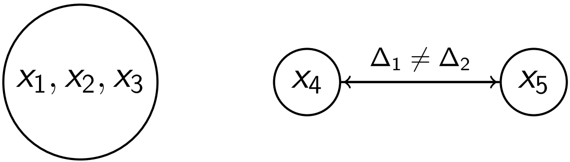


Back-up: Decomposition Process of RDG

Interaction Structure

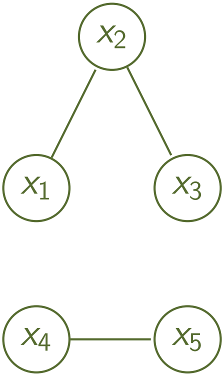


Decomposition Process

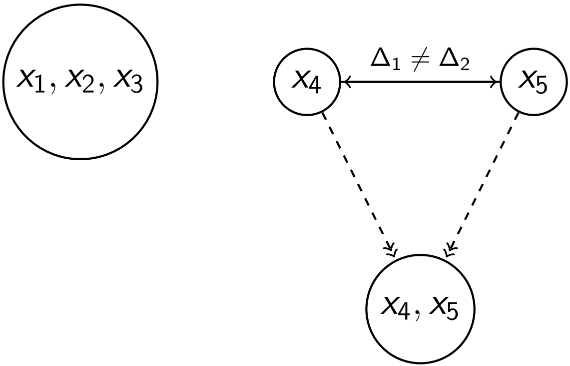


Back-up: Decomposition Process of RDG

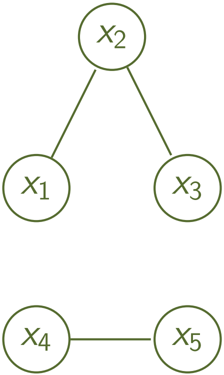
Interaction Structure



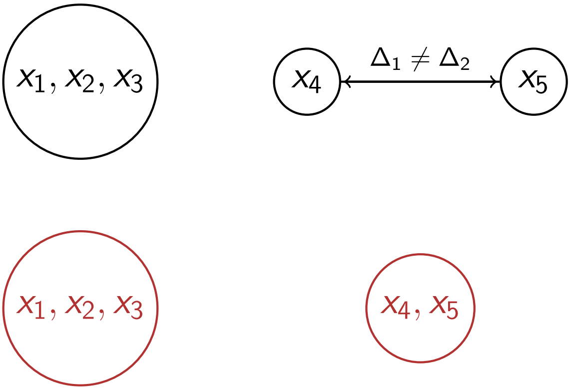
Decomposition Process



Interaction Structure



Decomposition Process



Back-up: Time Complexity of RDG

Time Complexity: $\mathcal{O}\big(n\log(n)\big)$

1. Fully separable problem: $3n \in \Theta(n)$.
2. Fully non-separable problem: $6n \in \Theta(n)$.
3. Partially separable problem: $6n\log_2(n) \in \Theta\big(n\log(n)\big)$.
4. Overlapping problem $6n\log_2(n) \in \Theta\big(n\log(n)\big)$.