

Sim-EDA: A Multipopulation Estimation of Distribution Algorithm Based on Problem Similarity

Krzysztof Michalak

Department of Information Technologies,
Institute of Business Informatics,
Wroclaw University of Economics, Wroclaw, Poland
krzysztof.michalak@ue.wroc.pl



Presentation Plan

- Basic concepts
 - Estimation of Distribution Algorithms (EDAs)
 - Multipopulation metaheuristics
- The Sim-EDA algorithm
- Test Problems
 - Max-Cut and the TSP
 - Generating similar problem instances
 - Problem-specific EDAs
- Experiments and Results
- Conclusions

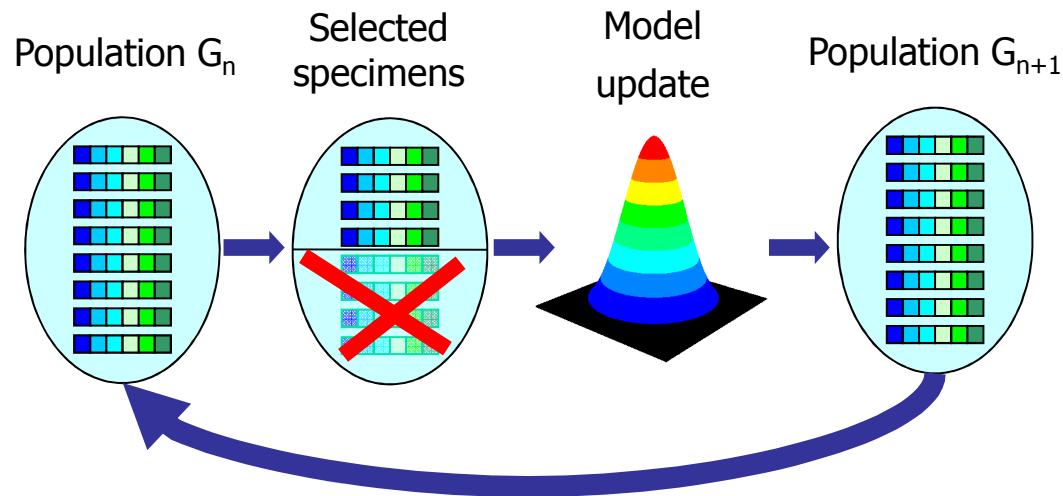


Estimation of Distribution Algorithms

- Population-based algorithms
- Representation of good solutions: a probabilistic model
- In comparison to EAs:
 - Specimens are used for evaluation
 - No genetic operators (instead, a probabilistic model is built)
 - New specimens are drawn from the probabilistic model

Estimation of Distribution Algorithms

- Main steps
 - Select specimens from the population
 - Update the probability model
 - Draw a new population from the model
 - Evaluate



Multipopulation metaheuristics

■ Idea

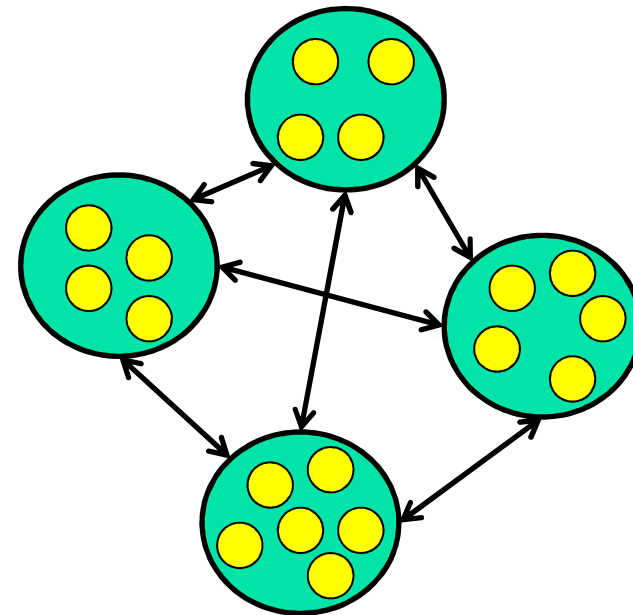
- Instead of one big population use several smaller ones
- Each subpopulation can work in a different part of the search space

■ Motivation

- Diversity preservation
- Multimodal problems
- Multiobjective optimization
- Dynamic optimization

■ Island model

- Isolated environments (islands)
- Migration





The Sim-EDA algorithm – overview

- A multipopulation algorithm...
- ...but each subpopulation tackles a **slightly different instance** of the problem
- Island model + migration
- Migration **influenced by similarities** of problem instances
- Each subpopulation processed using an **Estimation of Distribution Algorithm**



The Sim-EDA algorithm

- Main steps
 1. Initialize probabilistic models
 2. The main loop
 - 2.1. Generate new specimens
 - 2.2. Select source populations and specimens for migration
 - 2.3. Migrate specimens
 - 2.4. Update probabilistic models

- Details: [proceedings, page 239](#)



Problem similarity

- Problem instance **similarity matrix**:

$$S_{[N_{prob} \times N_{prob}]}$$

where:

N_{prob} – the number of problem instances

higher values = more similar instances

- Similarity between the i -th and j -th problem instances is expressed numerically as $S_{i,j}$



Problem similarity

- In this paper problem instances are represented as **weighted graphs**
- The i -th problem instance has a **cost matrix**:

$$C_{[N \times N]}^{(i)}$$

- Similarity between i -th and j -th problem instance:

$$S_{i,j} = - \sum_{p=1}^N \sum_{q=1}^N (C_{p,q}^{(i)} - C_{p,q}^{(j)})^2.$$



Migration

- For each destination population i select the source population j based on values of $S_{i,j}$
- Migration strategies:
 - **Nearest** – migrate specimens from the most similar subproblem (largest $S_{i,j}$ for $j \neq i$)
 - **Rank**
 - Rank all subpopulations from the largest $S_{i,j}$ to the smallest (for $j \neq i$)
 - Apply roulette-wheel selection
 - **Uniform** – migrate specimens from a subproblem chosen at random with uniform probability (from $j \neq i$)
 - **None** – no migration



Migration

- Incoming specimens are added one by one
 - x – incoming specimen
 - w – the weakest specimen in target population
 - binary tournament between x and w
 - if x wins it replaces w



Test Problems – The Max-Cut

- Find a partition of a graph $G = \langle V, E \rangle$ with N vertices

$$V' \cup V'' = V \quad V' \cap V'' = \emptyset$$

- Lengths of edges given in a **cost matrix** $C_{[N \times N]}$
- Each solution is a **binary vector**
- Minimization problem:

$$\text{maximize } f(V') = \sum_{i \in V', j \in (V - V')} c_{ij},$$

$$\text{subject to } V' \in 2^V,$$



Test Problems – The TSP

- Find the shortest route through N cities, visiting each city only once
- Distances given in a cost matrix $C_{[N \times N]}$
- Each solution is a permutation
- Minimization problem:

$$\text{minimize } f(\pi) = C_{\pi(N)\pi(1)} + \sum_{i=1}^{N-1} C_{\pi(i)\pi(i+1)}$$

$$\text{subject to } \pi \in \Pi(N)$$

where:

$\Pi(N)$ - the set of all permutations of numbers $1, \dots, N$



Generating similar problem instances

- $N_{sub} = 20$ cost matrices:

$$C_{[N \times N]}^{(1)}, \dots, C_{[N \times N]}^{(20)}$$

- In $C_{[N \times N]}^{(1)}$ the elements were drawn from $U[0, 100]$
- $C_{[K \times K]}^{(j)}$ was generated from $C_{[K \times K]}^{(j-1)}$ by replacing $1 / N_{prob}$ ($1/20 = 5\%$) of elements by random values
- To ensure **symmetry** the elements above and below the diagonal kept identical



Problem-specific EDAs (The Max-Cut)

- Model: a vector $P \in [0, 1]^N$
 - p_i – how probable it is that i -th element equals 1?
- Initially: $P = [0.5, 0.5, \dots, 0.5]$
- Model update from **Population-Based Incremental Learning (PBIL)** is used
 - Updates each component p_i of P separately...
 - ... based on the **best solution** $g^{(+)}$ and **worst solution** $g^{(-)}$
 - Parameters: the **positive learning rate** η_+ and the **negative learning rate** η_-
 - Denote $\eta = \eta_+ + \eta_-$



Problem-specific EDAs (The Max-Cut)

- Update rule

- if $g_j^{(+)} = g_j^{(-)}$

$$p_i = p_i \cdot (1 - \eta_+) + g_i^{(+)} \cdot \eta_+$$

- if $g_j^{(+)} \neq g_j^{(-)}$

$$p_i = p_i \cdot (1 - \eta) + g_i^{(+)} \cdot \eta$$

- Mutation

$$p_i = p_i \cdot (1 - \mu) + \alpha * \mu$$

where:

α - a 0 or 1 value drawn randomly with equal probabilities

μ - a mutation-shift parameter controlling the intensity of mutation



Problem-specific EDAs (The TSP)

- Model: a matrix $Q_{[N \times N]}$ with $q_{i,j} \in [0, 1]^N$
 - $q_{i,j}$ – how probable it is that j is placed right after i
- Initially:

$$q_{i,j} = \begin{cases} 0 & \text{for } i = j \\ \frac{1}{N-1} & \text{for } i \neq j \end{cases}$$

- Model update from **Population-Based Incremental Learning (PBIL)** is used
- **Best and worst permutations** converted to Q_{best} and Q_{worst} by setting 1 iff i follows j in the permutation



Problem-specific EDAs (The TSP)

- Obtaining permutations from $Q_{[N \times N]}$:

Algorithm 2. Initialization of a genotype for a new specimen for the TSP based on the probabilistic model Q .

IN: Q - the probabilistic model
 N - genotype length

OUT: X - a new genotype

$X[1] = \text{UniformSelection}(\{1, \dots, N\})$

for $k = 2, \dots, N$ **do**

$U = \{1, \dots, N\} - \{X[1], \dots, X[k-1]\}$

$i = X[k-1]$

$S = \sum_{j \in U} Q[i, j]$

if $S > 0$ **then**

$X[k] = \text{RouletteWheelSelection}(U, \{Q[i, j]\}_{j \in U})$

else

$X[k] = \text{UniformSelection}(U)$

end if

end for



Experiments

- Aimed at comparing the migration strategies
- Problem instances:
 - The Max-Cut Problem with $N = 12, 25, 50, 100, 250$ and 500 nodes
 - The Travelling Salesman Problem (TSP) with $N = 12, 25, 50, 100, 150, 200$ and 250 cities
- 30 runs for each instance and migration strategy
- Results compared at the common time limit (a time in which the None strategy completed 200 generations)
- Comparison performed using the Wilcoxon statistical test



Results

Table 1. Results of the tests on the Max-Cut problem (higher values are better).

Instance size	Migration strategy	Mean	Median	Std. dev.	Comparison to “none”	
					<i>p</i> -value	Interp.
12	None	2052.2833	2043.2797	68.7213		
	Nearest	2052.7526	2043.2797	67.1400	0.43307	Insignificant
	Rank	2054.2401	2043.2797	66.5184	0.0003651	Significant
	Uniform	2053.6531	2043.2797	67.5602	0.0066871	Significant
25	None	8671.3856	8664.3626	136.6357		
	Nearest	8686.8558	8674.5851	132.2364	1.68E-05	Significant
	Rank	8701.5194	8686.9078	131.8232	2.00E-20	Significant
	Uniform	8702.0109	8689.4084	133.8095	9.00E-21	Significant
50	None	34169.0427	34157.2758	262.0070		
	Nearest	34213.7399	34183.5902	259.1844	3.61E-06	Significant
	Rank	34262.5991	34233.2313	302.8604	4.88E-21	Significant
	Uniform	34252.5286	34246.1427	292.6219	1.73E-19	Significant
100	None	133281.7688	133292.2493	505.6580		
	Nearest	133405.0093	133433.3314	509.5968	7.09E-08	Significant
	Rank	133486.5788	133651.4388	817.3963	4.58E-19	Significant
	Uniform	133195.4960	133514.7050	1114.0702	1.27E-03	Significant
250	None	817251.5645	817303.3203	1411.8478		
	Nearest	818132.9153	818057.8073	1488.0998	7.94E-26	Significant
	Rank	813343.0263	815448.3402	6016.6470	3.58E-37	Worse
	Uniform	809292.8215	809954.2165	7203.6887	3.48E-79	Worse
500	None	3214635.065	3214426.042	3388.4932		
	Nearest	3214979.768	3215141.950	3875.1273	4.89E-02	Significant
	Rank	3187810.589	3190216.098	18081.6794	7.56E-98	Worse
	Uniform	3175792.355	3168177.289	20508.3657	8.48E-98	Worse

Nearest vs. None

N = 12: Insignificant

N > 12: Significant

Rank & Uniform vs. None

N ≤ 100: Significant

N ≥ 250: Worse



Results

Table 2. Results of the tests on the TSP (lower values are better).

Instance size	Migration strategy	Mean	Median	Std. dev.	Comparison to “none”	
					<i>p</i> -value	Interp.
12	None	217.3898	216.1576	13.3413		
	Nearest	217.3898	216.1576	13.3413	1	Insignificant
	Rank	217.3898	216.1576	13.3413	1	Insignificant
	Uniform	217.3898	216.1576	13.3413	0.22656	Insignificant
25	None	164.4112	156.5974	27.472		
	Nearest	164.3935	156.5974	27.4952	0.38316	Insignificant
	Rank	164.3905	156.5974	27.4855	0.91015	Insignificant
	Uniform	164.3892	156.5974	27.486	0.40226	Insignificant
50	None	223.0045	220.3383	27.4457		
	Nearest	222.3719	219.5455	28.2546	0.088667	Insignificant
	Rank	224.3653	220.6636	27.8668	0.00072179	Worse
	Uniform	223.5734	219.0627	27.6526	0.30527	Insignificant
100	None	365.9890	364.7559	28.1933		
	Nearest	359.3402	360.5102	24.6559	4.2904E-18	Significant
	Rank	365.0730	364.5377	26.6054	0.012205	Significant
	Uniform	365.4464	364.6945	26.9252	0.21189	Insignificant
150	None	404.8644	403.0477	26.8652		
	Nearest	403.4235	401.3445	25.9356	1.6479e-008	Significant
	Rank	404.7463	402.9095	26.7110	0.03125	Significant
	Uniform	404.7797	402.9095	26.7887	0.125	Insignificant
200	None	456.2797	456.6390	23.8788		
	Nearest	455.8308	456.2452	23.5170	6.1035e-005	Significant
	Rank	456.2793	456.6390	23.8788	1	Insignificant
	Uniform	456.1204	456.5193	23.7636	0.0625	Insignificant
250	None	510.5612	510.1462	24.3592		
	Nearest	510.3769	510.0111	24.1334	0.0078125	Significant
	Rank	510.5425	510.1462	24.3338	1	Insignificant
	Uniform	510.5477	510.1462	24.3529	0.5	Insignificant

Nearest vs. None
 $N \leq 50$: Insignificant
 $N \geq 100$: Significant

Rank & Uniform vs. None
 In most cases
 Insignificant



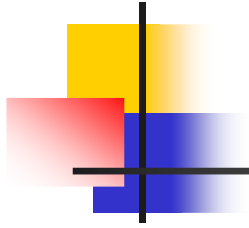
Conclusions

- An algorithm was proposed which combines a multipopulation approach with EDAs
- Sim-EDA is a general framework that can be combined with various EDAs
- Tests performed on Max-Cut Problem and TSP instances
- The strategy of migrating specimens solely from the nearest subpopulation
 - Consistently produced better results than those obtained when no migration was used for larger instances
 - Was not significantly different from no migration for some smaller instances



Further work

- Migration of probabilistic models instead of specimens
- Use the similarity of the distributions / models
- Real-life applications
 - Image segmentation
 - Transportation problems
- EDAs for the Firefighter Problem (FFP)



Thank you!
(questions?)