Master of Data Science | Faculty of Computer Science

Genetic Engineering Algorithm (GEA):
An Efficient Metaheuristic Algorithm for Solving Combinatorial Optimization Problems

Source: Automation and Remote Control, 2024, Vol. 85, No. 3, pp. 252–262.

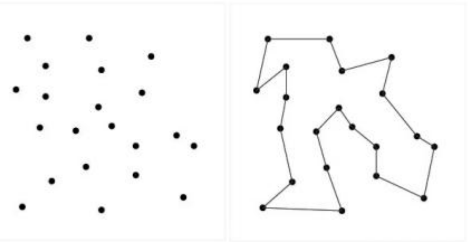
Author: Majid Sohrabi, Amir M. Fathollahi-Fard, and V. A. Gromov

Presenter: Komoltsewa Diana

Date: December 17, 2025

Combinatorial Optimization Problems

- Optimization problems with discrete-valued variables
- **Example problems:** MKP, TSP, graph partitioning, network flow, shortest path problems, matching, graph coloring, ...
- **Examples in real life:** crew scheduling, vehicle routing, facility layout, packing, pick-up and delivery, ...
- Exponential increase in computational complexity as the problem size grows.



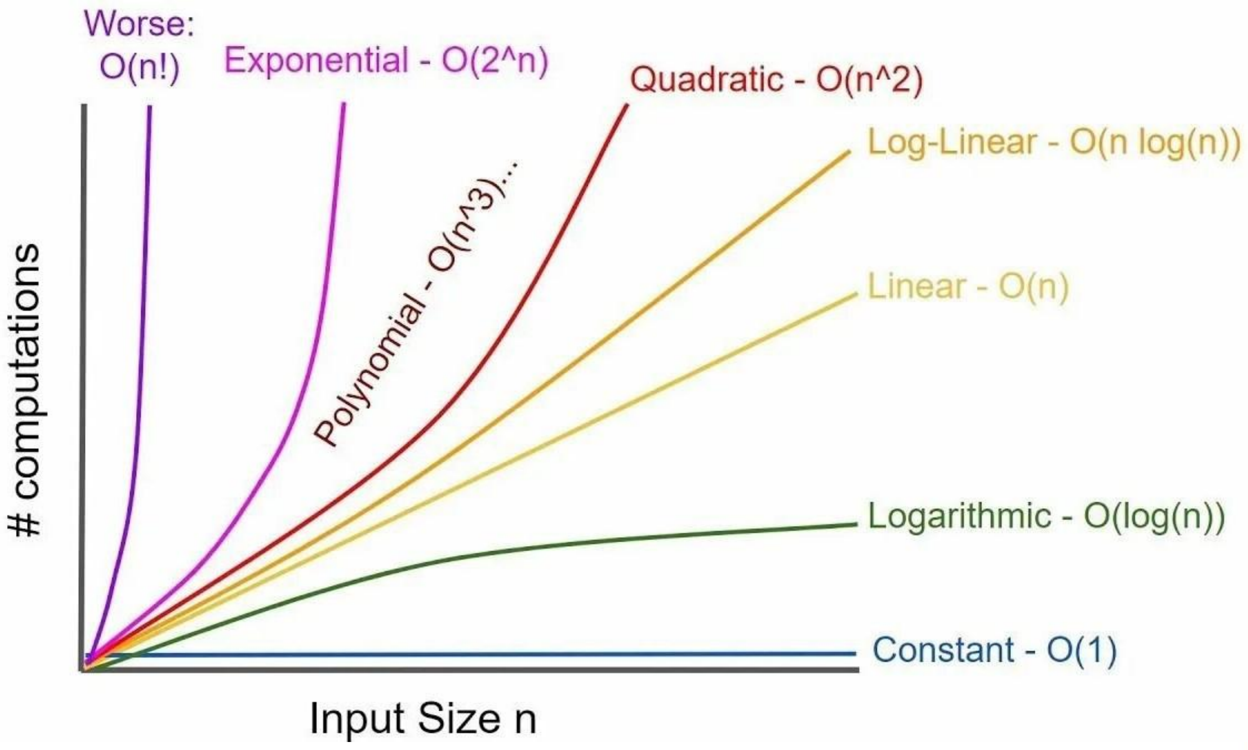
Travelling Salesman Problem
↓
Circuit design

Bin Packing Problem
↓
transportation

Hard problems (**NP**-complete)
3SAT
TRAVELING SALESMAN PROBLEM
LONGEST PATH
3D MATCHING
KNAPSACK
INDEPENDENT SET
INTEGER LINEAR PROGRAMMING
RUDRATA PATH
BALANCED CUT

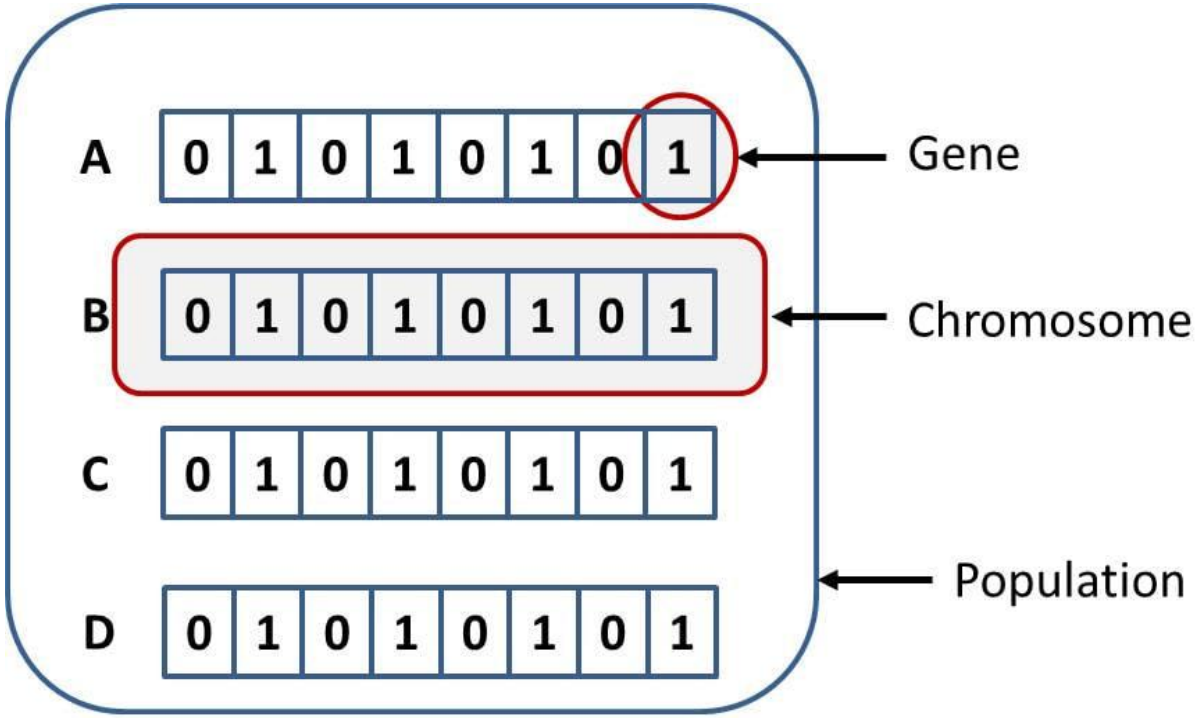
Classical Methods

- **Exhaustive algorithm:** brute-force algorithm that systematically enumerates all possible solutions to a problem and checks each one to see if it is a valid solution.
- **Problems:** slow, computationally expensive for problems with a large search space.
- **Advantage:** guaranteed solution finding, simplicity.
- **Examples:** e.g. linear programming, integer linear programming, branch-and-bound, ...



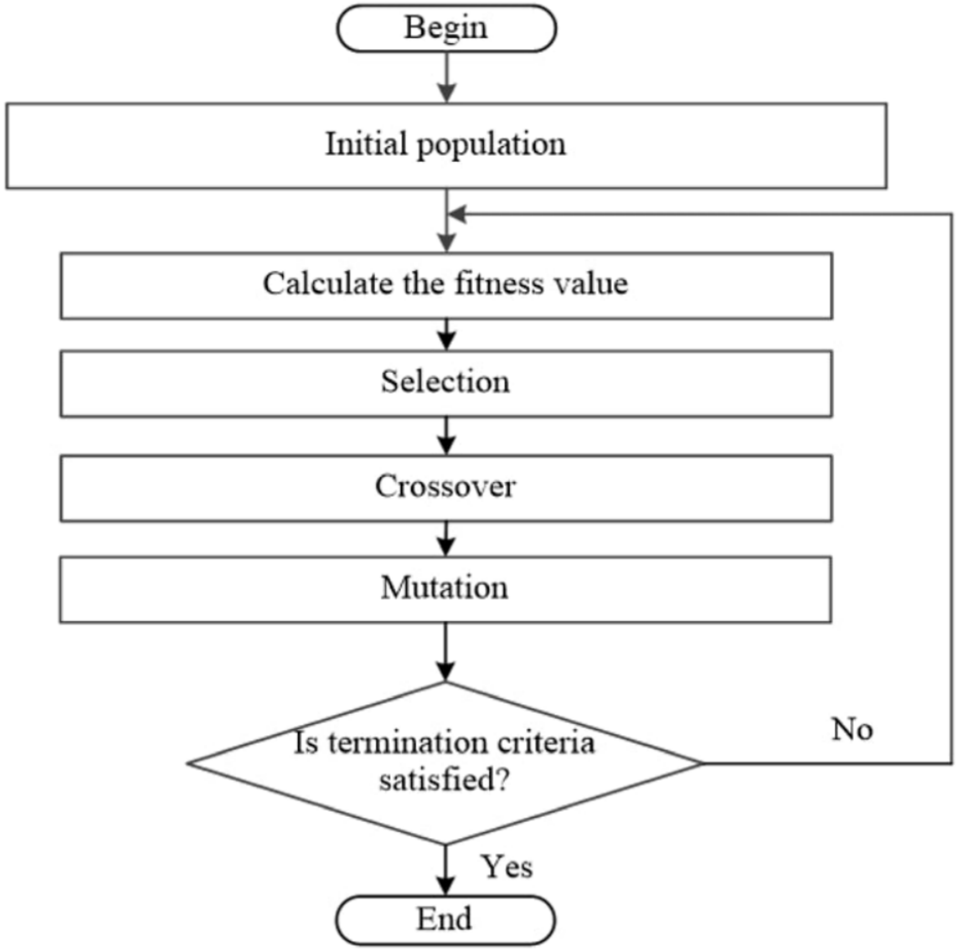
Traditional Genetic Algorithms

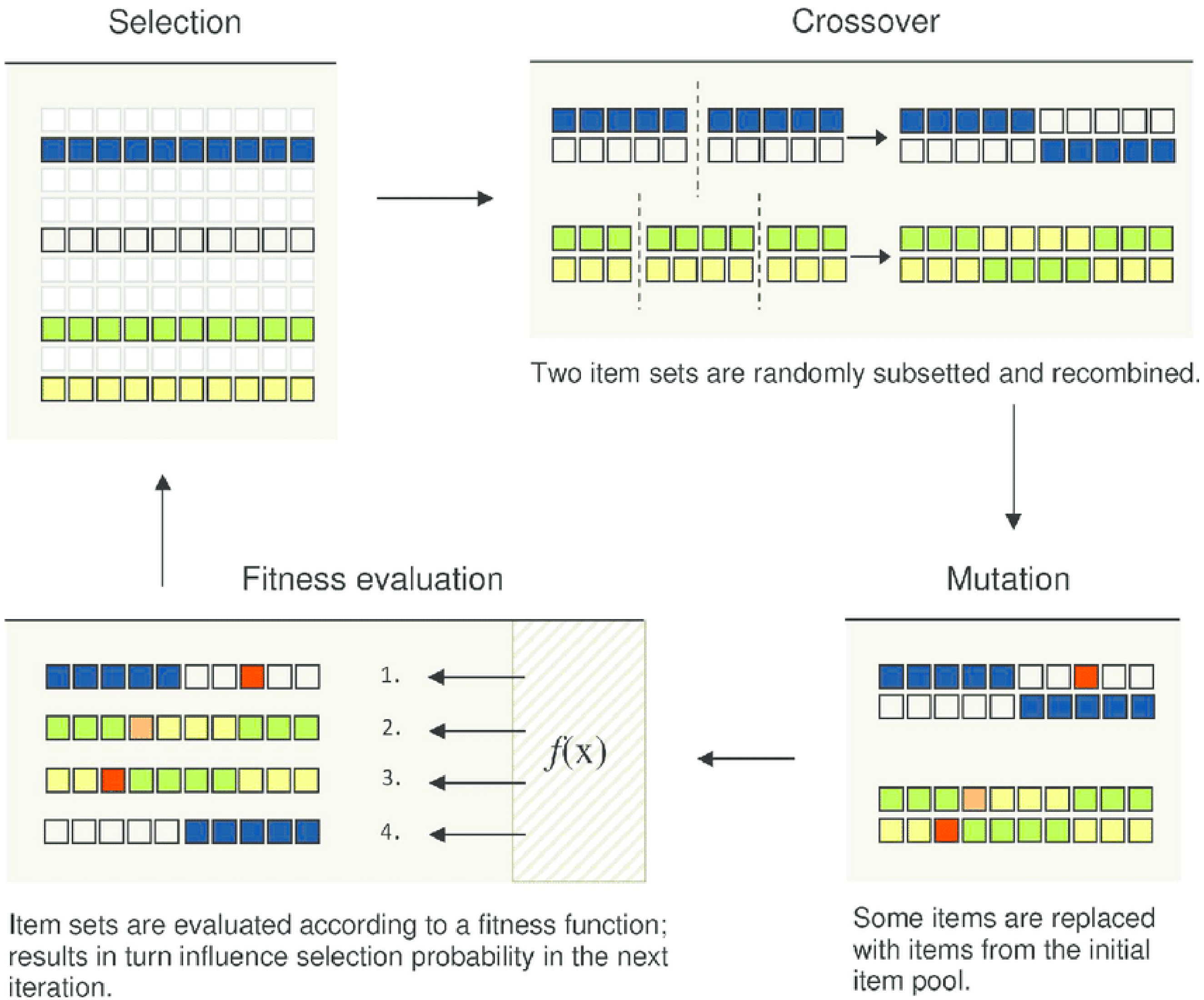
- A traditional Genetic Algorithm is a population-based stochastic optimization method modeled after Darwinian evolution.
- Each candidate solution is encoded as a chromosome, typically represented as a binary vector, a sequence, or a real-valued array.
- The algorithm maintains a finite population of solutions and iteratively updates it through fitness-based **selection**, **crossover**, and **mutation**.



Gene, Chromosomes and Population

Traditional Genetic Algorithms





Limitations of Traditional GAs

- **Premature convergence** — the population becomes too similar too early, trapping the algorithm in a local optimum;
- **Blind crossover** — recombination mixes genes randomly without preserving meaningful structures;
- **Blind mutation** — random changes are applied uniformly, often damaging good patterns;
- **Lack of elite analysis** — the algorithm does not extract useful gene patterns from the best solutions;
- **High sensitivity to hyperparameters** — performance strongly depends on parameter settings and is unstable to small changes.

Genetic Engineering Algorithm

- GEA integrates **genetic engineering techniques** into the evolutionary process.
- Uses operators inspired by **gene isolation, purification, insertion, and expression**.
- Allows flexible customization: any operator (crossover → gene injection) may be enabled or skipped.
- Still begins with a standard initial population and problem-specific fitness evaluation.
- Can handle various combinatorial optimization problems (routing, scheduling, knapsack, facility location).
- In this study, chromosomes are represented as **binary strings**, and three new GE-based operators are introduced.

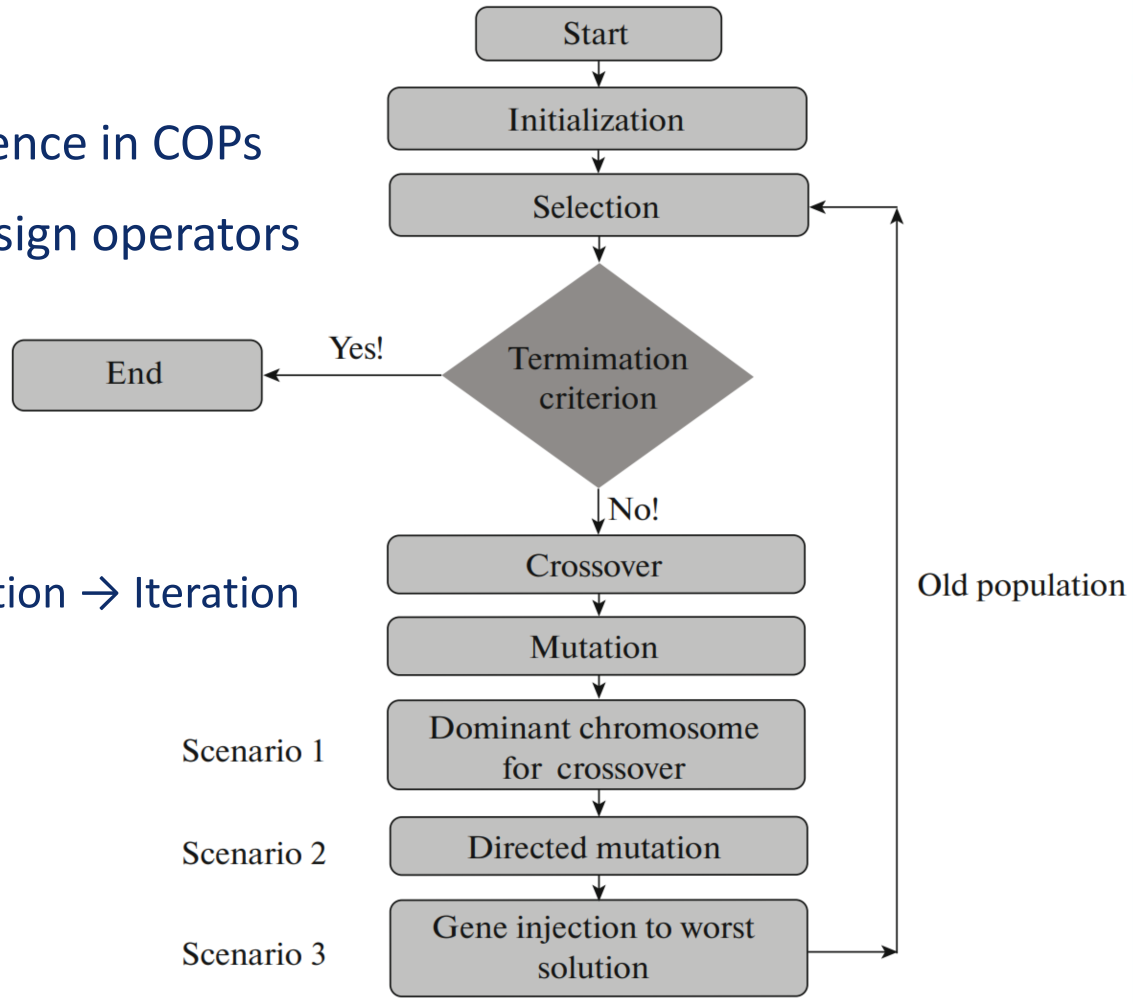
GEA = GA Main Loop + 3 GE Operators

Genetic Engineering Algorithm (GEA)

- **Goal:** address GA's randomness and premature convergence in COPs
- **Core idea:** borrow genetic engineering concepts to redesign operators
- **Output:** three plug-in GE operators (Scenario 1/2/3)

From GA to GEA: Framework Difference

- **Same as GA**
  - Initialization → Fitness → Selection → Crossover → Mutation → Iteration
- **New modules after crossover/mutation**
  - Dominant chromosome extraction
  - Directed mutation
  - Gene injection
- **Operators are switchable** → GEA1/2/3 for ablation



Scenario 1: Dominant Chromosome

How does it work?

- Select the **top p% elite individuals**
- Compute per-gene repetition frequency → **repetition matrix**
- Take the most frequent allele at each locus → **dominant chromosome (DC)**
- Use DC as **a guided crossover template** (and later for injection)

Why is it useful?

- DC represents **reliable building blocks** from elites
- Crossover **preserves good structures** more often
- **Reduces destructive randomness**
- Provides **high-quality genes** for Scenarios 2 & 3

p% of population

1	0	1	0	1
0	0	0	0	0
0	0	0	0	1
1	0	1	1	1
1	1	0	0	0
1	0	1	1	0
1	0	1	0	0
1	1	1	0	1
1	1	1	0	0
1	0	0	0	0

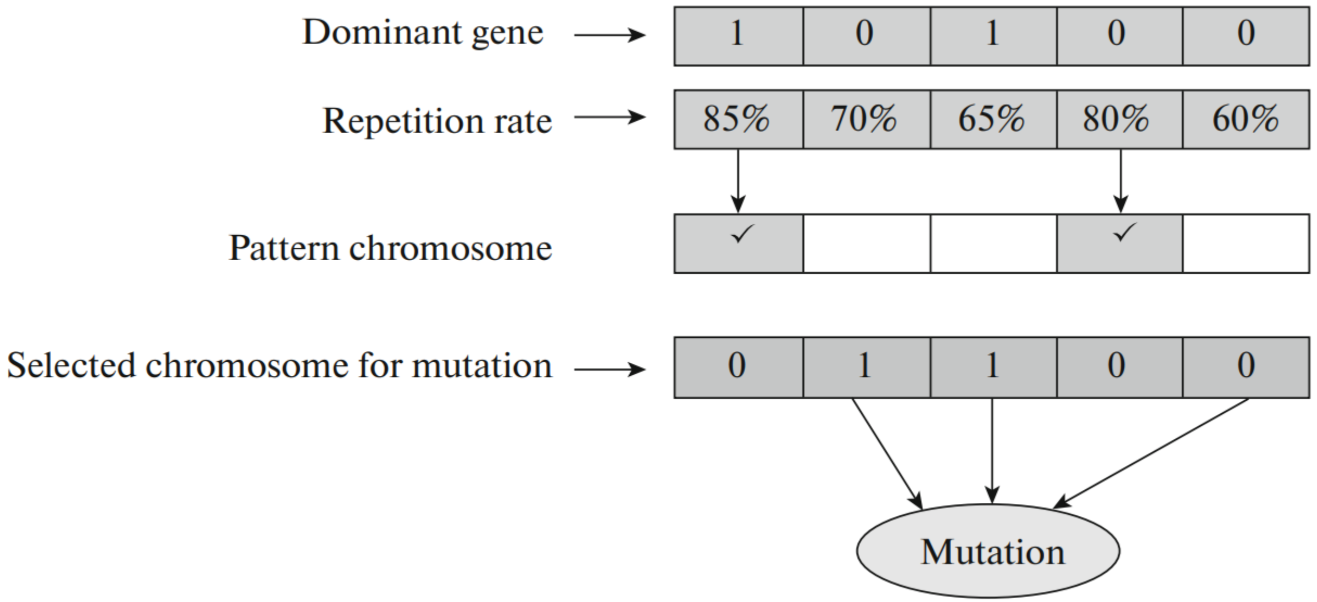
Domiant gene →

1	0	1	0	0

Scenario 2: Directed Mutation

How does it work?

- Start from **top p% elites**
- Identify high-repetition loci → **desired / informative genes**
- Build **Mask/Pattern matrix**:
    - 1 = informative (locked)
    - 0 = uninformative (mutable)
- Flip bits **only where Mask = 0**



Why is it useful?

- Random mutation may destroy good genes
- Directed mutation:
    - **protects elite consensus**
    - focuses **search budget** on unresolved loci
- **Improves convergence speed and stability**

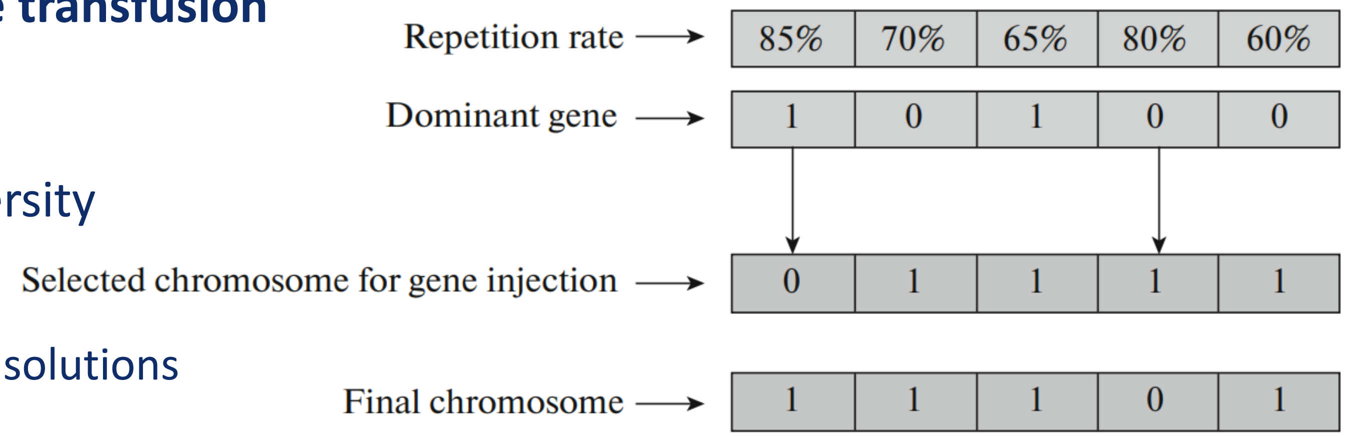
Scenario 3: Gene Injection

How does it work?

- Focus on the **worst 1–p% individuals**
- Use the Mask to **locate informative loci**
- Copy DC alleles into these loci (**replace poor genes**)
- **Rapidly upgrades weak individuals** via “**gene transfusion**”

Why is it useful?

- Weak individuals may still contain useful diversity
- Injection:
  - **shortens** the climb from poor to competitive solutions
  - raises overall **population quality** faster
  - **keeps diversity** by **editing only informative loci**



Experimental Setup

- **Five algorithms** are considered: the GEA, traditional GA and three variations of GEA, namely GEA1, GEA2, and GEA3, each utilizing a specific scenario as explained earlier. In GEA, the main loop uses all the operators in sequence in each iteration.
- A standard **vehicle routing optimization problem** is being solved by the algorithms. This problem involves determining optimal routes for a fleet of vehicles to visit a set of demand points while minimizing transportation costs.
- **Six** well-established **instances** from the literature are selected, instance is (demand points × number of vehicle).
- To ensure consistency, the **maximuiterations** is set to 1000 and **the population size** to 100 for all algorithms. The **crossover and mutation percentages m number of** are uniformly set to 0.8 and 0.1, respectively, across all algorithms.
- **Ten independent runs** of each algorithm on every test instance are performed and the best, worst, average, and standard deviation of the solutions obtained by each algorithm are reported.

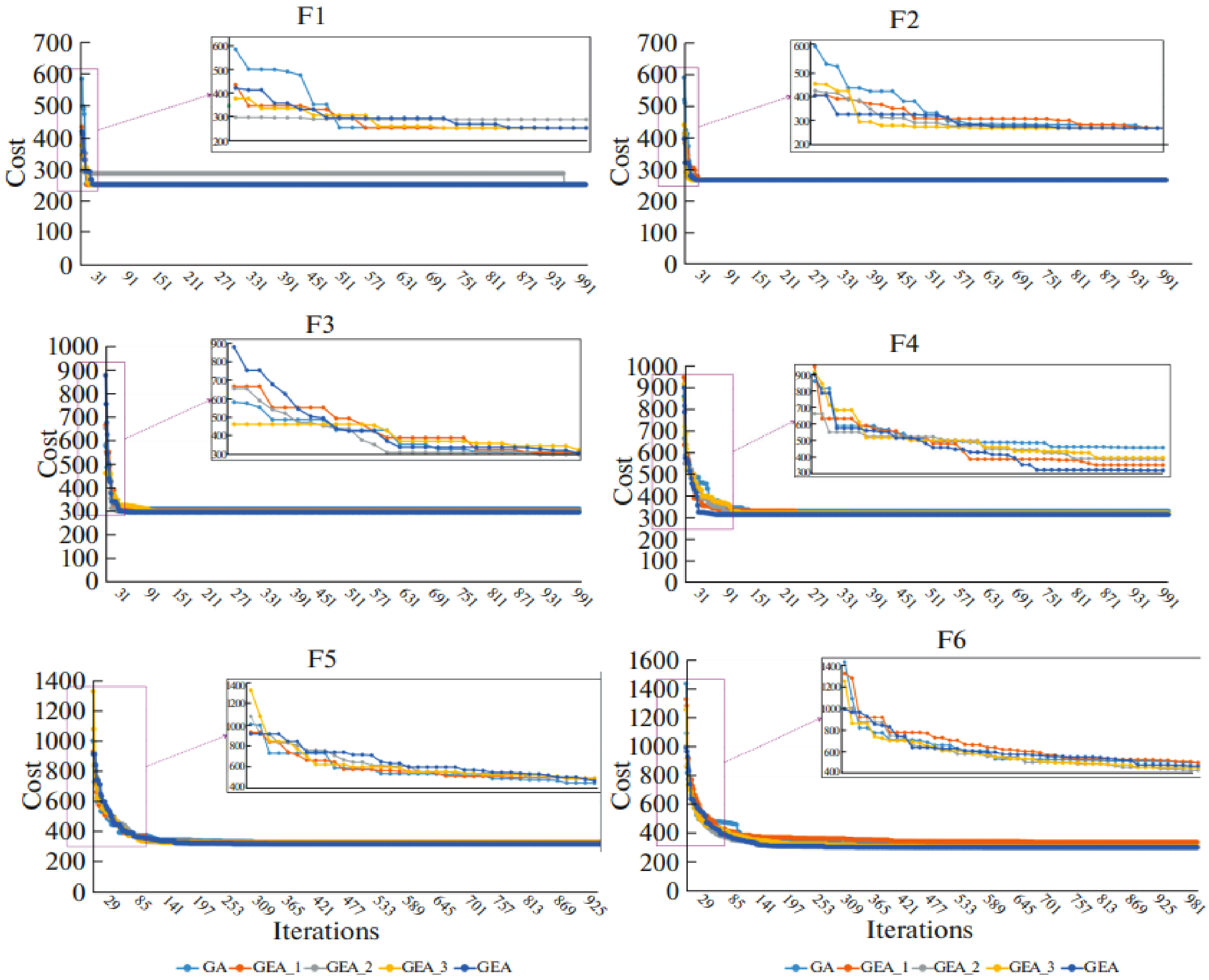
Algorithms Performances

Table 1. Report of the algorithms results based on criteria of the Best = B, Worst = W, Mean = M, and Standard deviation = Std. (The best values in each criterion and test instance are highlighted in bold.)

Test instance		F1	F2	F3	F4	F5	F6
demand points × number of vehicle		8 × 3	10 × 3	14 × 4	20 × 4	25 × 5	30 × 5
GA	B	**257.3492**	**268.1687**	**301.6661**	317.6503	326.5457	308.8542
	W	291.6624	269.0742	316.0882	351.2298	363.0131	343.9097
	M	260.7805	268.7120	305.8615	333.5178	338.1742	321.1532
	Std	10.8507	0.4675	5.4002	12.3733	11.3976	11.5798
GEA_1	B	**257.3492**	**268.1687**	**301.6661**	319.3303	319.5602	307.1991
	W	**257.3492**	269.0742	318.8057	342.2278	359.0854	370.7047
	M	**257.3492**	268.2593	304.7409	324.0378	330.8722	328.0479
	Std	**5.99E-14**	0.2863	5.4787	6.8149	10.8851	21.2861
GEA_2	B	**257.3492**	**268.1687**	**301.6661**	**317.1235**	321.5556	**302.5377**
	W	**257.3492**	269.0742	306.3834	353.7992	359.0854	**322.7266**
	M	**257.3492**	268.3498	302.8296	327.4803	333.1713	**311.4745**
	Std	**5.99E-14**	0.3817	1.6555	12.1645	13.4915	**7.3910**
GEA_3	B	**257.3492**	**268.1687**	**301.6661**	317.6503	319.0169	308.8834
	W	**257.3492**	269.0742	306.3834	**331.8416**	**331.3571**	346.2497
	M	**257.3492**	268.4404	302.3684	323.3476	326.245	323.5826
	Std	**5.99E-14**	0.4373	1.58597	5.45505	**4.1059**	13.7657
GEA	B	**257.3492**	**268.1687**	**301.6661**	317.6503	**317.7347**	304.4598
	W	**257.3492**	**268.1687**	**301.6661**	**331.8416**	331.4877	343.6004
	M	**257.3492**	**268.1687**	**301.6661**	**321.4611**	**323.1822**	313.4242
	Std	**5.99E-14**	**5.99E-14**	**0**	**4.7726**	6.0097	11.8294

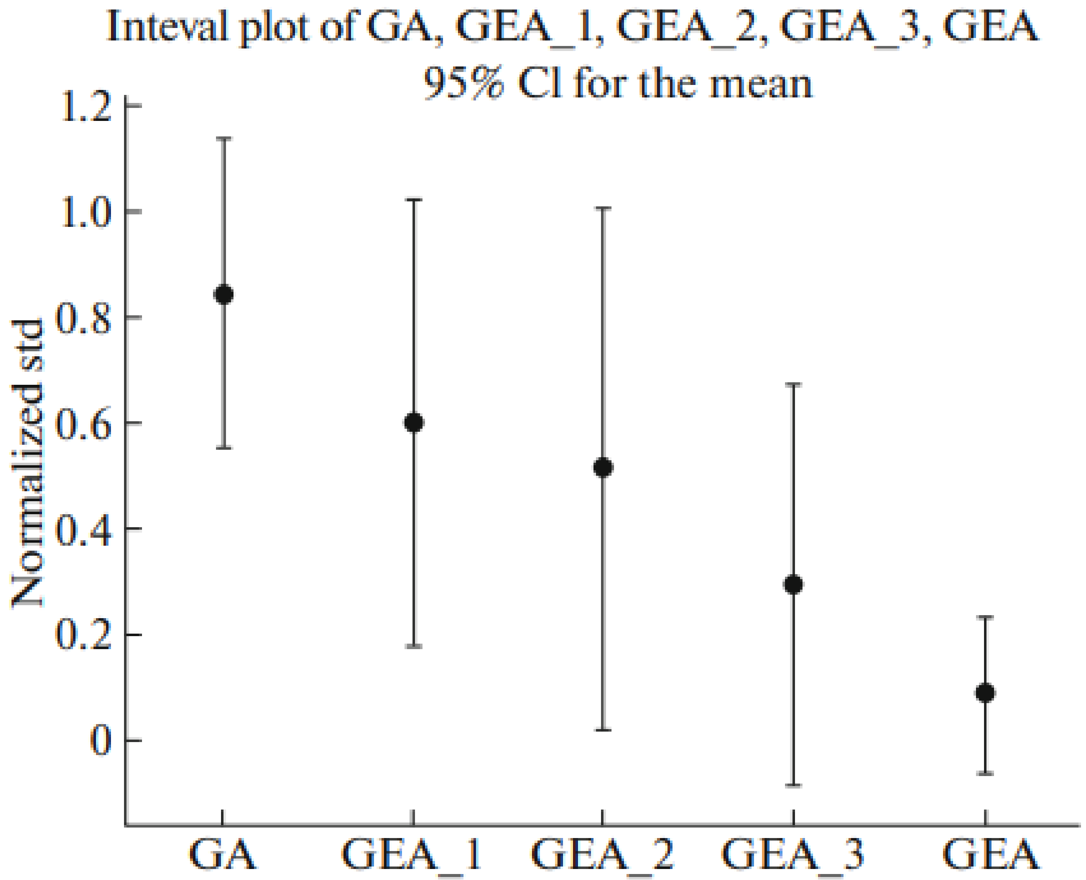
In most instances, the GEA, when utilizing all scenarios, discovers near-optimal solutions superior to those obtained by the other algorithms. Among the GEA variations, GEA2 stands out as the most successful, confirming the strength of the second scenario in exploring better near-optimal solutions.

Convergence Rates of Algorithms



All algorithms exhibit an acceptable convergence rate across the test instances, with similar solution quality.

Standard Deviations across Algorithms

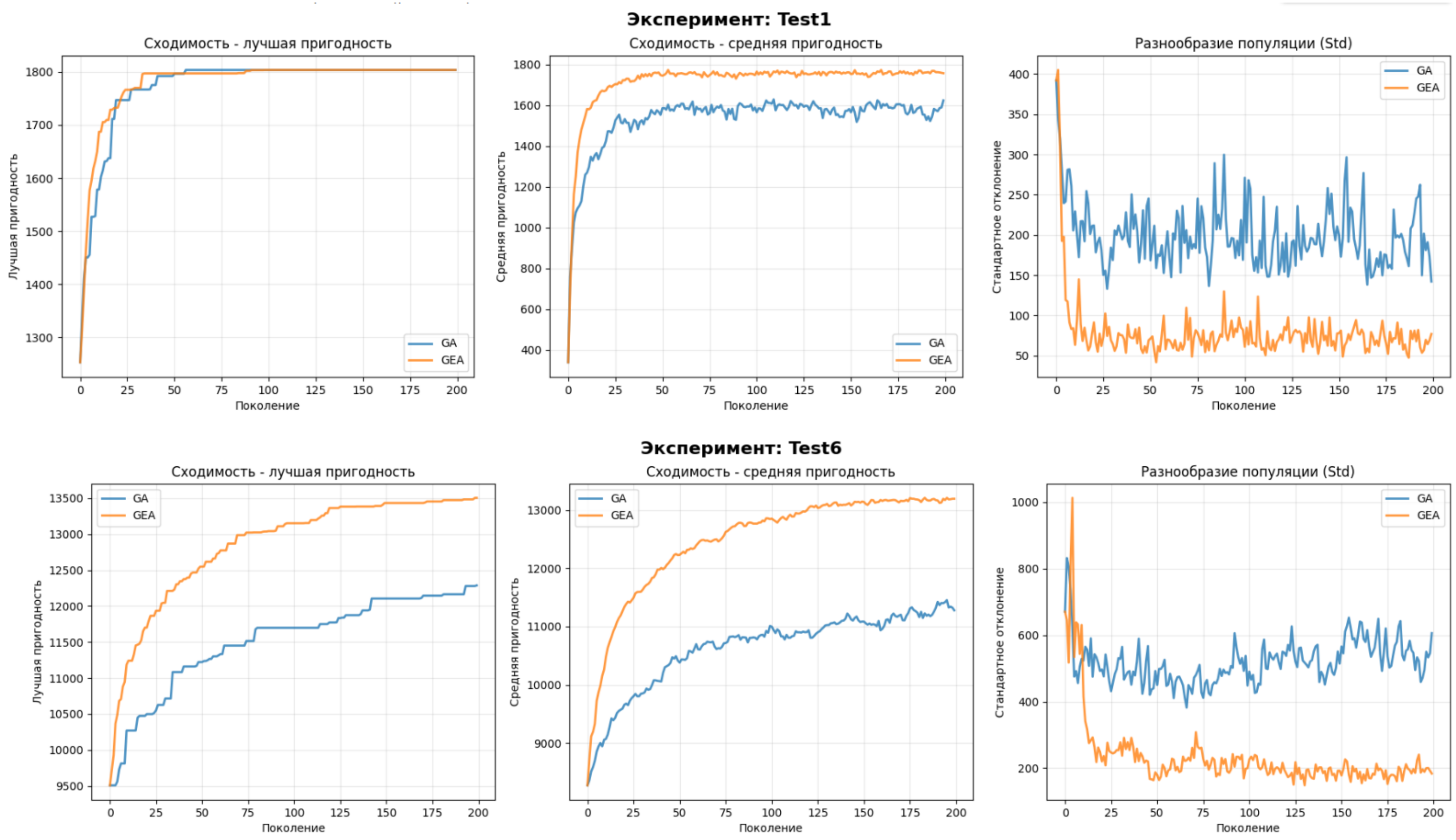


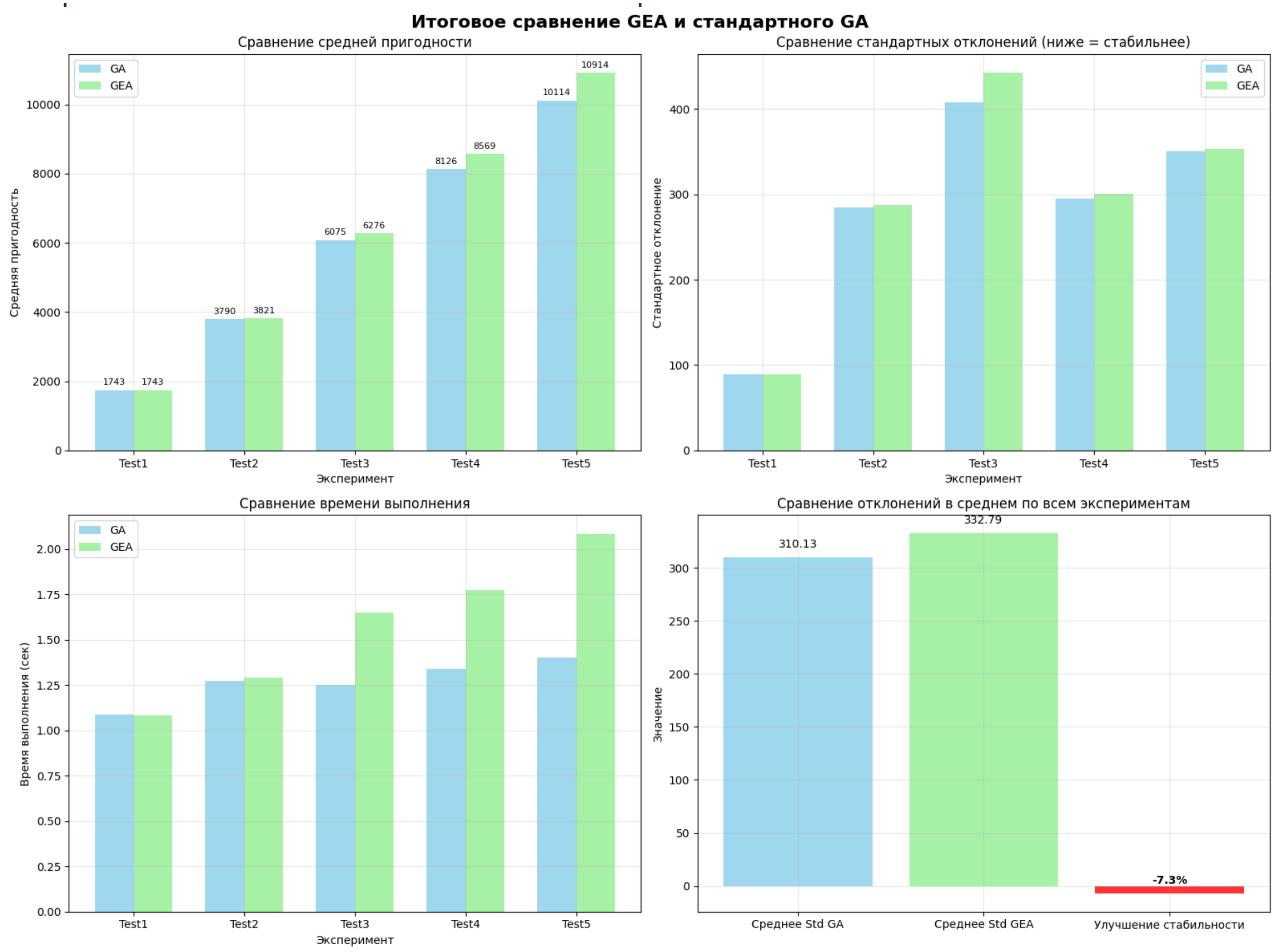
Individual standard deviations are used to calculate the intervals.

Statistical analysis using a 0.95 confidence level, employing normalized standard deviations across all algorithms, is performed. The results of it support the highest accuracy of the proposed GEA compared to the other algorithms.

Knapsack Problem

Maximum iterations is set to 200 and **the population size** to 100 for all algorithms. The **crossover and mutation percentages m number of** are uniformly set to 0.8 and 0.02, respectively, across all algorithms.





Эксперимент	Кол-во предметов	Вместимость	Средний GA	Средний GEA	Std GA	Std GEA	Время GA (с)	Время GEA (с)
Test1	50	100	1743.08	1743.08	88.74	88.74	1.086	**1.083**
Test2	100	200	**3789.77**	3821.07	**284.18**	287.25	**1.272**	1.292
Test3	150	300	6074.88	**6276.33**	**407.58**	442.39	**1.253**	1.650
Test4	200	400	8125.72	**8568.68**	**295.01**	300.56	**1.341**	1.771
Test5	250	500	10114.35	**10913.99**	**350.38**	353.00	**1.400**	2.083
Test6	300	600	12059.41	**13137.81**	**434.92**	524.81	**1.481**	2.486

Algorithms being compared

The following **methods** were tested in the study:

1. The Standard Genetic Algorithm (GA) with roulette selection.
2. Genetic Engineering Algorithm (GEA) with directional heuristics.
3. GA with tournament selection is the selection of the best individual from a random subgroup.
4. Adaptive GA — dynamic adjustment of crossover and mutation probabilities.
5. Hybrid GA with local search (Memetic Algorithm) — combining global GA with local optimization.

Comparison metrics:

- Quality of the solution (average fitness)
- Execution time (seconds)
- Generations before convergence
- Efficiency (suitability/time)


Эксперимент: Small
Сходимость - лучшая пригодность
Лучшая пригодность
Поколение
Сходимость - средняя пригодность
Средняя пригодность
Разнообразие популяции (Std)
Стандартное отклонение
Стандартный GA
GEA
Турнирный GA
Адаптивный GA
Гибридный GA

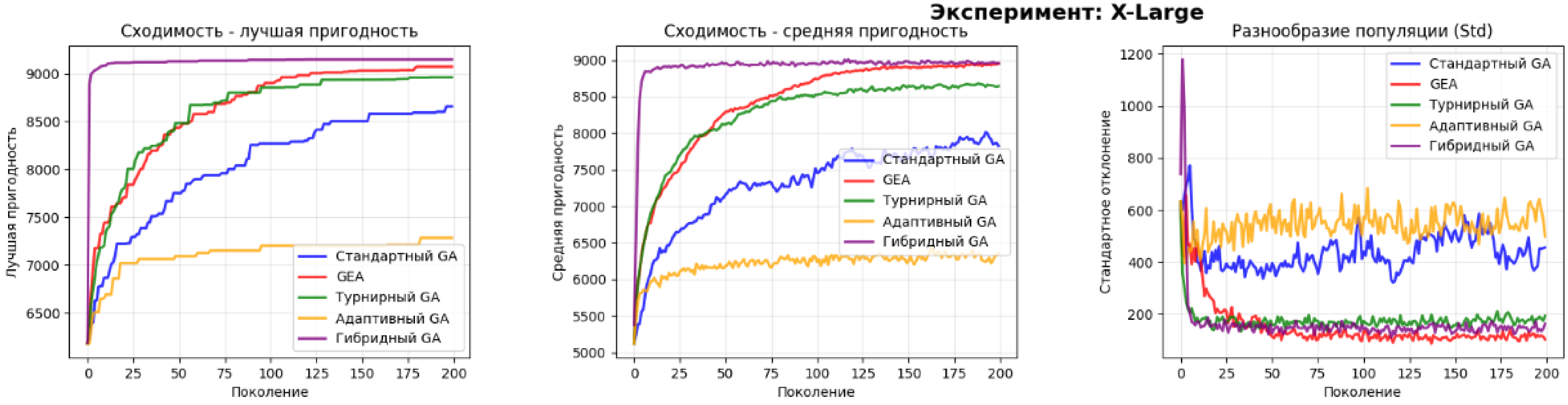

Эксперимент: X-Large
Сходимость - лучшая пригодность
Лучшая пригодность
Поколение
Сходимость - средняя пригодность
Средняя пригодность
Разнообразие популяции (Std)
Стандартное отклонение
Стандартный GA
GEA
Турнирный GA
Адаптивный GA
Гибридный GA

Таблица 1. Качество решения (средняя пригодность)

Эксп.	Размер	Вместимость	Стандартный GA	GEA	Турнирный GA	Адаптивный GA	Гибридный GA
Small	50	100	1754.9	1754.9	1754.9	1599.6	1754.9
Medium	100	200	3794.4	3827.7	3827.9	3233.6	3828.2
Large	150	300	6275.8	6497.8	6468.1	5327.0	6508.3
X-Large	200	400	8154.3	8535.7	8484.4	6864.7	8644.0
Average	125	250	4994.9	5154.0	5133.8	4256.2	5183.9

Таблица 2. Время выполнения (секунды)

Эксп.	Стандартный GA	GEA	Турнирный GA	Адаптивный GA	Гибридный GA
Small	1.149	1.249	0.988	7.016	26.048
Medium	1.145	1.517	1.157	9.002	98.328
Large	1.163	1.777	1.107	10.954	222.726
X-Large	1.487	2.211	1.100	12.640	396.802
Average	1.236	1.688	1.088	9.903	185.976

Таблица 3. Количество итераций до сходимости (поколений)

Эксп.	Стандартный GA	GEA	Турнирный GA	Адаптивный GA	Гибридный GA
Small	47	39	36	24	12
Medium	73	86	73	30	18
Large	82	99	80	27	19
X-Large	85	111	86	25	20
Average	72	83	68	27	17

Таблица 4. Процентное улучшение GEA относительно других алгоритмов (%)

Сравнение	Пригодность	Время	Сходимость
GEA vs Стандартный GA	3.2	-36.6	-16.3
GEA vs Турнирный GA	0.4	-55.2	-21.9
GEA vs Адаптивный GA	21.1	82.9	-214.3
GEA vs Гибридный GA	-0.6	99.1	-382.4

Conclusion of GEA's Efficacy

Core Findings

- **Superior Performance:** Full GEA (integrating all 3 scenarios) outperforms traditional GA and partial variants in combinatorial optimization (validated via vehicle routing problems).
- **Key Success Factors:** Precise gene manipulation (dominant chromosome selection, directed mutation, gene injection) reduces randomness and leverages elite solution information.
- **Practical Value:** Provides a robust metaheuristic for real-world optimization (e.g., transportation routing, scheduling) requiring efficiency and reliability.

Academic Contribution

- Expands metaheuristic research by bridging genetic engineering and evolutionary algorithms.
- Validates the effectiveness of problem-specific gene manipulation in overcoming GA limitations.

Future Research Directions

1. Parameter Optimization

- Investigate the impact of key parameters (p% for elite population, mutation threshold, scenario weights) on performance across different problem types.
- Develop adaptive parameter tuning mechanisms to enhance GEA's versatility.

2. Expansion of Application Scenarios

- Extend evaluation to other combinatorial optimization problems (e.g., flow-shop scheduling, knapsack, facility location planning)
- Compare with state-of-the-art metaheuristics (e.g., Whale Optimization Algorithm, Harris Hawks Optimization) beyond traditional GA.

3. Algorithm Hybridization

- Integrate GEA with machine learning techniques (e.g., neural networks) or other metaheuristics to further improve optimization capabilities.
- Explore hybrid models for multi-objective optimization scenarios.

4. Scalability Enhancement

- Test GEA on large-scale problem instances (more demand points, complex constraints) to assess scalability.
- Optimize computational efficiency for industrial-level applications with massive solution spaces.

Master of Data Science

Faculty of Computer Science

Thank you for your attention!

Presenter: Komoltsewa Diana

Date: December 17, 2025