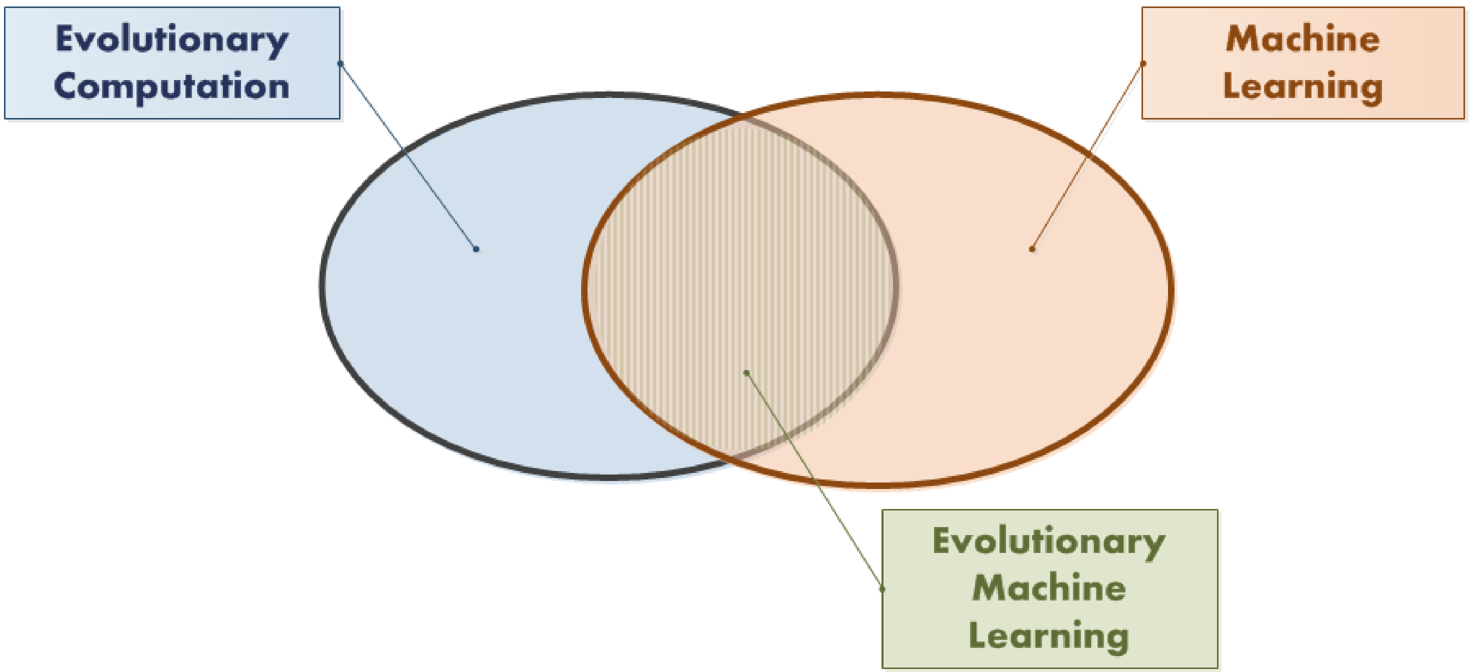
SIBERIAN STATE AEROSPACE UNIVERSITY
Krasnoyarsk, Russia

Cooperative Multi-Objective Genetic Algorithm with Parallel Implementation

Christina Brester and Eugene Semenkin

Integration of Evolutionary Computation and Machine Learning



Integration of Evolutionary Computation and Machine Learning

Pros	***Cons***
✓ The classification accuracy of the best **evolutionary** and **non-evolutionary** methods are **comparable;** ✓ Population-based search is easily **parallelized;** ✓ These methods can work in the **dynamic non-stationary environment;** ✓ **Feature selection** and **learning** in one process might be **combined;** ✓ From an optimization perspective, learning problems are typically **large, non-differentiable, noisy, deceptive, multimodal, high-dimensional, and highly constrained**. Evolutionary algorithms are an **effective tool** for such problems.	X Evolutionary methods are generally much **slower** than the non-evolutionary alternatives ➢ *Possible solution: parallelization* X The **performance of evolutionary algorithms varies** significantly for different problems ➢ *Possible solution: cooperative algorithms*

Integration of Evolutionary Computation and Machine Learning

Pros	*Cons*
✓ The classification accuracy of the best **evolutionary** and **non-evolutionary** methods are **comparable;**	X Evolutionary methods are generally much **slower** than the non-evolutionary alternatives ➢ *Possible solution: parallelization*
✓ Population-based search is easily **parallelized;**	
✓ These methods can work in the **dynamic non-stationary environment;**	
✓ **Feature selection** and **learning** in one process might be **combined;**	X The **performance of evolutionary algorithms varies** significantly for different problems ➢ *Possible solution: cooperative algorithms*
✓ From an optimization perspective, learning problems are typically **large, non-differentiable, noisy, deceptive, multimodal, high-dimensional, and highly constrained**. Evolutionary algorithms are an **effective tool** for such problems.	

Outline

- **Motivation**
  - The Evolutionary Computation and Machine Learning Integration
- **Background**
  - Conventional Multi-Objective Genetic Algorithms
  - Test Problems
  - Experiment Conditions
  - Experimental Results and Discussion
- **Proposed approach**
  - The Island Model
  - Cooperative Multi-objective Genetic Algorithm
- **Results and Discussion**
  - Experiment Conditions
  - Experimental Results and Discussion
- **Conclusion and Future Plans**

Multi-Objective Genetic Algorithms (MOGAs)

- *Generate the* ***initial population***
- *Evaluate* ***criteria*** *values*
- *While (stop-criterion!=true), do:*

{

- Estimate ***fitness-values****;*

- Choose the most appropriate individuals with the mating ***selection*** *operator based on their fitness-values;*

- Produce new candidate solutions with ***recombination****;*

- Modify the obtained individuals with ***mutation****;*

*- Compose the new population (****environmental selection****);*

}

Multi-Objective Genetic Algorithms

Designing a MOGA, researchers are faced with some issues:

- fitness assignment strategies,
- diversity preservation techniques,
- ways of elitism implementation.

Our task:

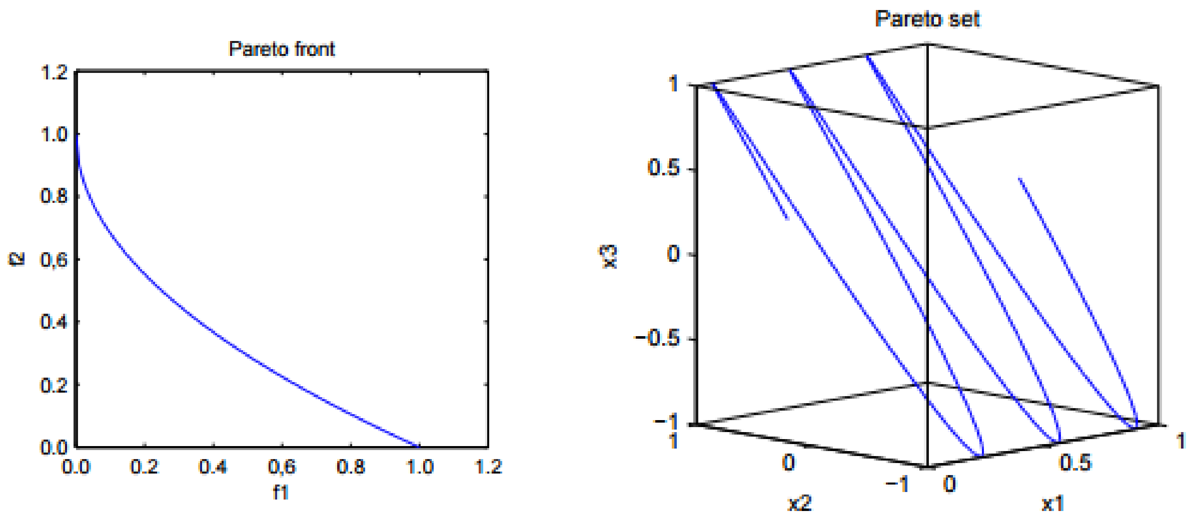
✓ To investigate the **effectiveness of MOGAs**, which are based **on various heuristic mechanisms**

Basic features of the MOGA used

MOGA	Fitness Assignment	Diversity Preservation	Elitism
NSGA-II	Pareto-dominance (niching mechanism) and diversity estimation (crowding distance)	Crowding distance	Combination of the previous population and the offspring
PICEA-g	Pareto-dominance (with generating goal vectors)	Nearest neighbour technique	The archive set and combination of the previous population and the offspring
SPEA2	Pareto-dominance (niching mechanism) and density estimation (the distance to the k-th nearest neighbour in the objective space)	Nearest neighbour technique	The archive set

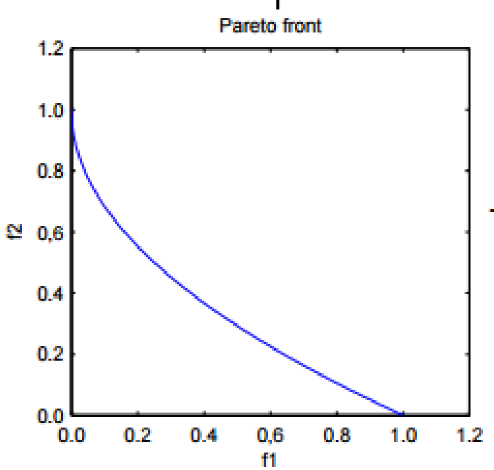
Some of Test Instances CEC'2009

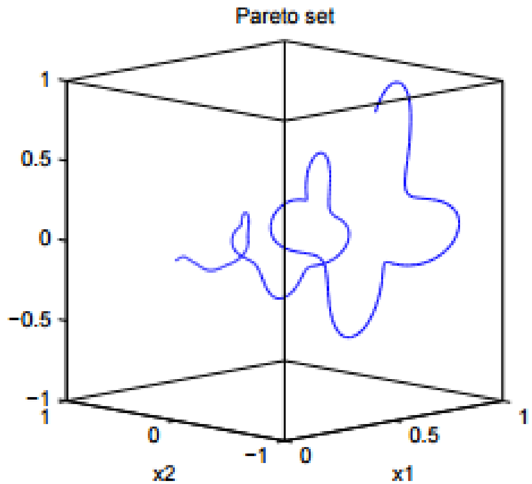
№	Test Problem	Pareto Set and Front
UF1	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert}\sum_{j \epsilon J_1}(x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2 \to min,$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert}\sum_{j \epsilon J_2}(x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2 \to min,$ where $n = 30, J_1 = \{j \mid j\ is\ odd,\ 2 \le j \le n\}$ and $J_2 = \{j \mid j\ is\ even, 2 \le j \le n\}$. $\bar{x} \epsilon\ [0,1] \cdot [-1,1]^{n-1}$.	Pareto Set: $0 \le x_1 \le 1,$ $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, \dots, n.$ Pareto Front: $f_2 = 1 - \sqrt{f_1}, 0 \le f_1 \le 1.$



Some of Test Instances CEC'2009

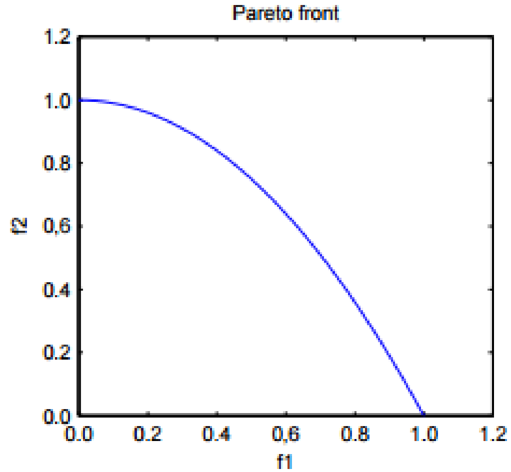
№	Test Problem	Pareto Set and Front
UF2	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert} \sum_{j \epsilon J_1} y_j^2 \to min,$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2 \rvert} \sum_{j \epsilon J_2} y_j^2 \to min,$ where $n = 30, J_1 = \{j \vert j \text{ is odd}, \; 2 \le j \le n\}$ and $J_2 = \{j \vert j \text{ is even}, 2 \le j \le n\}$. $y_j = \begin{cases} x_j - \left(0.3x_1^2 \cdot \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right) \cdot \\ \cdot \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), j\epsilon J_1, \\ x_j - \left(0.3x_1^2 \cdot \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right) \cdot \\ \cdot \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j\epsilon J_2, \end{cases}$ $x \epsilon [0,1] \cdot [-1,1]^{n-1}$.	Pareto Set: $0 \le x_1 \le 1,$ $x_j = \begin{cases} \left(0.3x_1^2 \cdot \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right)\right) + \\ +0.6x_1 \cdot \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), j\epsilon J_1, \\ \left(0.3x_1^2 \cdot \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right)\right) + \\ +0.6x_1 \cdot \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j\epsilon J_2. \end{cases}$ Pareto Front: $f_2 = 1 - \sqrt{f_1},$ $0 \le f_1 \le 1.$

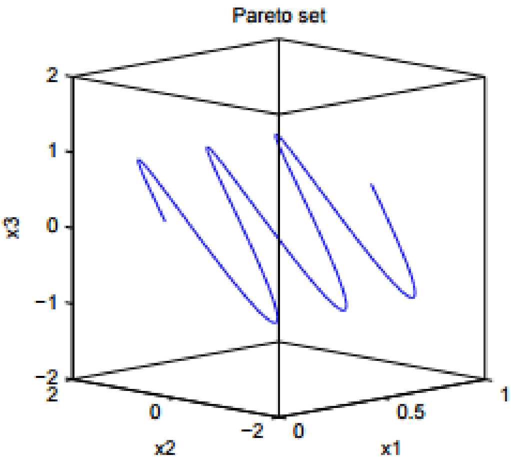




Some of Test Instances CEC'2009

№	Test Problem	Pareto Set and Front
UF4	$f_1 = x_1 + \frac{2}{\lvert J_1 \rvert}\sum_{j\epsilon J_1} h(y_j) \rightarrow min,$ $f_2 = 1 - x_1^2 + \frac{2}{\lvert J_2 \rvert}\sum_{j\epsilon J_2} h(y_j) \rightarrow min,$ where $n = 30, J_1 = \{j \mid j \text{ is odd}, \ 2 \le j \le n\}$ and $J_2 = \{j \mid j \text{ is even}, 2 \le j \le n\},$ $y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n,$ $h(t) = \frac{\lvert t \rvert}{1 + e^{2\lvert t \rvert}},$ $x \epsilon [0,1] \cdot [-2,2]^{n-1}.$	Pareto Set: $0 \le x_1 \le 1,$ $x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n.$ Pareto Front: $f_2 = 1 - f_1^2,$ $0 \le f_1 \le 1.$





Performance Metric

The metric ***IGD*** was used to estimate the quality of obtained Pareto Front approximations:

$$IGD(A, P^*) = \frac{\sum_{v \epsilon P^*} d(v, A)}{|P^*|},$$

where P^* is a set of uniformly distributed points along the Pareto Front (in the objective space), A is an approximate set to the Pareto Front, $d(v, A)$ is the minimum Euclidean distance between v and the points in A.
In short, the $IGD(A, P^*)$ value reflects the average distance from P^* to A.

Experiment Conditions

- **The maximal number of function evaluations** was equal to **300 000**.
- **The maximal number of solutions** in the approximate set produced by each algorithm for computing the IGD metric was **100** and **150** for two-objective and three-objective problems respectively.
- For all of the test instances IGD values were averaged over **25 runs** of each algorithm.

Experiment Conditions

For all of the algorithms the following settings were defined:

- **binary tournament selection,**
- **uniform recombination,**
- **the mutation probability** $p_m=1/n$, where n is the length of the chromosome.

- As usual, MOGAs (NSGA-II, SPEA2, and PICEA-g) operated with binary strings and therefore, we used **standard binary coding** to get real values of variables.

Experimental Results

Test func.	NSGA-II		PICEA-g		SPEA2	
	IGD	Time (sec.)	IGD	Time (sec.)	IGD	Time (sec.)
UF1	**0.097**	196.060	0.107	42.327	0.100	236.677
UF2	0.061	181.520	**0.060**	84.538	0.078	262.089
UF3	**0.191**	181.150	0.222	36.781	0.326	237.594
UF4	**0.055**	182.233	0.0570	75.837	0.083	243.208
UF5	**0.426**	181.509	0.498	33.844	0.518	240.198
UF6	0.335	183.085	0.346	34.997	**0.319**	237.906
UF7	**0.085**	181.039	0.091	75.556	0.125	245.891
UF8	0.269	190.269	**0.191**	166.056	0.259	253.813
UF9	0.319	191.105	**0.290**	107.157	0.407	406.996
UF10	0.626	186.267	**0.421**	118.744	0.534	290.870

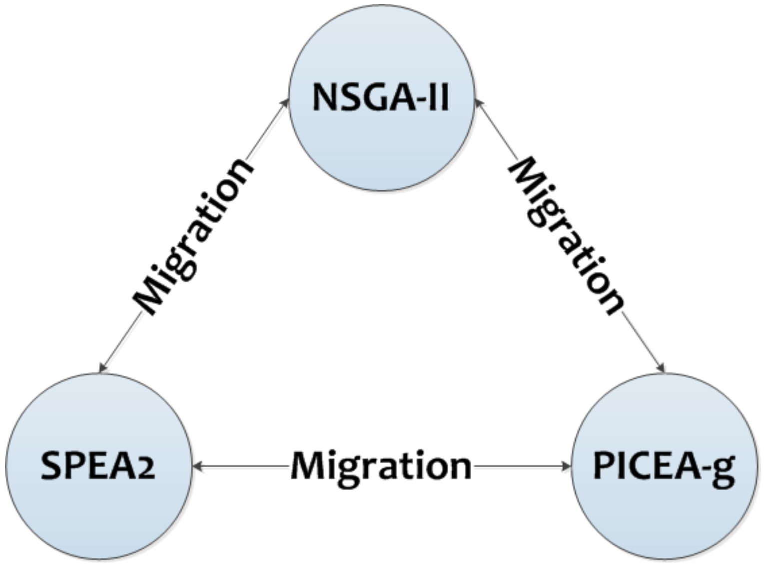
Discussion

- A **t-test** (with the significance level p=0.05) was used to compare the results:

there was no one MOGA which demonstrated the highest effectiveness (in the sense of the IGD metric) for all of the test problems.

- Possible solution: Cooperation of genetic algorithms which are based on different concepts (study NSGA-II, PICEA-g, and SPEA2)

Cooperative Multi-Objective Genetic Algorithm



Island model …

- ✓ is based on parallel work of islands;
- ✓ has an ability to preserve genetic diversity;
- ✓ could be applied to separable problems.

Experiment Conditions

- The computational resources (300 000 function evaluations) **were distributed** to all of the components equally.
- The migration size was **50** (in total each island got 100 points from two others).
- The migration interval was **25 generations**.
- Again all results were averaged over **25 runs**.

Experimental Results

Test func.	NSGA-II		PICEA-g		SPEA2		Cooperative algorithm		Result of t-test
	IGD	Time (sec.)	IGD	Time (sec.)	IGD	Time (sec.)	IGD	Time (sec.)	
UF1	**0.097**	196.06	0.107	42.33	0.010	236.68	0.068	56.57	**Outperforms the best value**
UF2	0.061	181.52	**0.060**	84.54	0.078	262.09	0.056	64.84	Corresponds to the **best** value
UF3	**0.191**	181.15	0.222	36.78	0.326	237.59	0.202	55.95	Corresponds to the **best** value
UF4	**0.055**	182.23	0.0570	75.84	0.083	243.21	0.058	60.27	Corresponds to the **best** value
UF5	**0.426**	181.51	0.498	33.84	0.518	240.20	0.338	56.39	**Outperforms the best value**
UF6	0.335	183.09	0.346	35.00	**0.319**	237.91	0.254	56.01	**Outperforms the best value**
UF7	**0.085**	181.04	0.091	75.56	0.125	245.89	0.084	60.27	**Outperforms** the **best** value
UF8	0.269	190.27	**0.191**	166.06	0.259	253.81	0.259	87.24	Corresponds to the **second** value
UF9	0.319	191.11	**0.290**	107.16	0.407	407.00	0.314	78.53	Corresponds to the **best** value
UF10	0.626	186.27	**0.421**	118.74	0.534	290.87	0.533	75.12	Corresponds to the **best** value

Conclusions and Future Plans

The proposed multi-agent heuristic procedure:

- ✓ does **not require additional experiments** to expose the most appropriate algorithm for the problem considered,
- ✓ might be **effectively** used instead of any of its component,
- ✓ allows us to **decrease the computational time** significantly due to the parallel work of island model components.

The algorithm developed has already been applied:

- ➢ **to select informative features** from data bases (two criteria were introduced – the Intra- and Inter-class distances).
- ➢ **to design neural network models** taking into account two criteria (the computational complexity and the accuracy).

Thanks a lot!