Cooperative Multi-objective Genetic Algorithm with Parallel Implementation

Christina Brester(✉) and Eugene Semenkin

Siberian State Aerospace University, Krasnoyarsk, Russia
christina.brester@gmail.com, eugenesemenkin@yandex.ru

Abstract. In this paper we introduce the multi-agent heuristic procedure to solve multi-objective optimization problems. To diminish the drawbacks of the evolutionary search, an island model is used to involve various genetic algorithms which are based on different concepts (NSGA-II, SPEA2, and PICEA-g). The main benefit of our proposal is that it does not require additional experiments to expose the most appropriate algorithm for the problem considered. For most of the test problems the effectiveness of the developed algorithmic scheme is comparable with (or even better than) the performance of its component which provides the best results separately. Owing to the parallel work of island model components we have managed to decrease computational time significantly (approximately by a factor of 2.7).

Keywords: Heuristic search · Multi-objective genetic algorithm · Multi-agent approach · Island model · Cooperation

1 Introduction

In recent times there has been a growing interest in the sphere of Evolutionary Ma-chine Learning: owing to a number of benefits which heuristic-based optimization methods have demonstrated, researchers have proposed several effective applications of Evolutionary Computation in the Machine Learning field [1], [2], [3]. This has become possible for several reasons: evolutionary algorithms are universal and might be used to find the optimal solution in both continuous and discrete search spaces; they could be applied in a dynamic environment; in most cases the effectiveness of evolutionary approaches is not lower than the performance of non-evolutionary ones [4].

However, some researchers highlight the negative sides of the Evolutionary Computation and Machine Learning integration. Firstly, it is always necessary to investigate a number of algorithms to define the most effective one for the problem considered because the performance of evolutionary algorithms varies significantly for different problems. Secondly, these methods require more computational resources compared with alternative non-evolutionary algorithms.

This paper is devoted to solving optimization problems with several criteria, and therefore, we attempt to develop a modified multi-objective genetic algorithm (MOGA) with these drawbacks removed.

Y. Tan et al. (Eds.): ICSI-CCI 2015, Part I, LNCS 9140, pp. 471–478, 2015.
DOI: 10.1007/978-3-319-20466-6_49

To overcome the disadvantages of the evolutionary search, an island model is used to involve genetic algorithms (GA) which are based on different concepts (NSGA-II, SPEA2, and PICEA-g). Moreover, this model allows us to parallelize calculations and, consequently, to reduce computational time.

As a result, we have managed to implement the multi-agent heuristic procedure to solve multi-objective optimization problems, which does not require additional experiments to expose the most appropriate algorithm for the problem considered. Besides, due to the parallel work of island model components we have achieved a significant decrease in computational time (roughly by a factor of 2.7). According to the results obtained, for most of the test problems the effectiveness of the developed algorithmic scheme is comparable with the performance of its component which provides the best results separately.

The rest of the paper is organized as follows: in Section 2 a description of the cooperative algorithm developed is presented. The test problems used to investigate the effectiveness of our proposal are introduced in Section 3. The experiments conducted, the results obtained, and the main inferences are included in Section 4. The conclusion and future work are presented in Section 5.

2 Developed Approach

2.1 Cooperative Multi-objective Genetic Algorithm

Designing a MOGA, researchers are faced with some issues which are referred to fitness assignment strategies, diversity preservation techniques, and ways of elitism implementation. However, the common scheme of any MOGA includes the same steps as any conventional one-criterion GA:

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);
    }

In contrast to one-criterion GAs, the outcome of MOGAs is the set of non-dominated points which form the Pareto set approximation.

To eliminate a number of questions which are raised while designing multi-criteria evolutionary methods, in this study we propose a cooperation of several GAs based on various heuristic mechanisms.

Generally speaking, an *island model* [5] of a GA implies the parallel work of several algorithms. A parallel implementation of GAs has shown not just an ability to preserve genetic diversity, since each island can potentially follow a different search trajectory, but also could be applied to separable problems. The initial number of individuals M is spread across L subpopulations: $M_i=M/L$, $i=1,\ldots,L$. At each T-th generation algorithms exchange the best solutions (*migration*). There are two parameters: *migration size*, the number of candidates for migration, and *migration interval*, the number of generations between migrations. Moreover, it is necessary to define the island model topology, in other words, the scheme of migration. We use the fully connected topology that means each algorithm shares its best solutions with all other algorithms included in the island model. The multi-agent model is expected to preserve a higher level of genetic diversity. The benefits of the particular algorithm could be advantageous in different stages of optimization. In this study the Non-Sorting Genetic Algorithm II (NSGA-II) [6], the Preference-Inspired Co-Evolutionary Algorithm with goal vectors (PICEA-g) [7], and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [8] were used to be involved as parallel working islands (Figure 1).

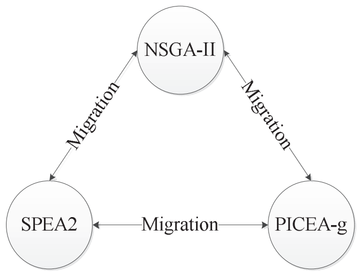

Fig. 1. The island model implemented

The next subsection provides a concise description of the algorithms included in the cooperation and their essential features.

2.2 Brief Description of Island Model Components

Several decades ago Goldberg suggested the usage of the Pareto-dominance idea as the main principle of fitness assignment in any evolutionary algorithm [9]. Since that time this strategy has proved its effectiveness and substituted other alternative proposals. Therefore, the chosen methods (NSGA-II, SPEA2, and PICEA-g) are based on the Pareto-dominance idea. However, there are various ways of its implementation [10]: some algorithms use the dominance rank (the amount of individuals by which the candidate-solution is dominated); in others the dominance depth is evaluated (this implies the division of a population into several fronts and determination of the front which an individual belongs to); the dominance count might also be taken into consideration (in other words, the amount of points dominated by a certain individual), and so on. Thus, the algorithms involved in the island model accomplish diverse fitness assignment strategies based on the Pareto-dominance idea (Table 1).

Table 1. 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

Besides, diversity preservation techniques are incorporated in most of the MOGAs to maintain variety within Pareto Set and Front approximations. There are also several ways to implement these techniques [11]. Kernel methods estimate the density with a Kernel function, which takes the distance to another point as an argument. Nearest neighbour techniques are based on the assessment of the distance between a given point and its k-th nearest neighbour. And histograms present another class of density estimators that use a hypergrid to calculate neighbourhoods. In most cases, these approaches define the distance between points in the objective space.

Moreover, there is the problem of losing effective individuals during the optimization process due to stochastic effects, and to solve this problem the idea of elitism has been suggested. Generally, there are two ways to implement it. The first strategy to cope with the problem is to combine the parent population with the offspring and then to apply a deterministic selection procedure taking into account fitness values of individuals from the mating pool. Another strategy is based on the usage of a secondary population which is called archive to copy there promising solutions at each generation. Actually, one of these techniques might be implemented: in NSGA-II the first strategy is used, whereas in SPEA2 the second one is applied, but they might also be combined (as a case in point, PICEA-g).

In Table 1 a brief description of the MOGAs involved in the cooperative algorithm and their main features are summarized.

In the approach developed we have tried to engage various heuristic concepts to implement fitness assignment strategies, diversity preservation techniques, and the elitism idea. On the one hand, it is supposed to lead to an increase in the algorithm reliability. On the other hand, due to the parallel structure of the island model computational time might be decreased.

The next section includes a description of the test problems which have been used to investigate the effectiveness of the approach proposed.

3 Test Problems

To investigate the effectiveness of the approach proposed in comparison with its components, we have engaged a set of high-dimensional test problems designed by the international scientific community to compare the effectiveness of developed algorithmic schemes (the CEC 2009 competition [12]). There are problems with discrete and continuous, convex and non-convex Pareto Sets and Fronts.

In this study we use a number of these test instances which are unconstrained two- and three-objective optimization problems with real variables.

In the CEC 2009 competition the metric IGD was used to estimate the quality of obtained Pareto Front approximations:

$$IGD(A, P^*) = \frac{\sum_{v \epsilon P^*} d(v,A)}{|P^*|}, \tag{1}$$

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.

These continuous multi-objective optimization test problems have been proposed in the past 25 years. In the CEC 2009 competition they were gathered to investigate the algorithms developed. Although in this study we do not compare our proposal with the winners of this competition, it is fair to notice that for most of the test problems the effectiveness of the approach developed is higher than the effectiveness of some methods from the list of winners (the list of thirteen best algorithms).

The next section provides a description of the experiments conducted, the results obtained and a brief discussion of them.

4 Experiments and Results

Firstly, conventional algorithmic schemes were applied to solve the problems introduced. All algorithms were provided with the same amount of resources: according to the rules of the CEC 2009 competition, 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.

In the experiments conducted the following settings were defined: binary tournament selection, uniform recombination and 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.

Secondly, a similar experiment was conducted for the developed cooperative multi-objective algorithm. 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), and the migration interval was 25 generations. Again all results were averaged over 25 runs.

The main criterion which was used to compare the effectiveness of the algorithm proposed with the performance of its components was the IGD metric. However, we also measured computational time required in each case. The results obtained are presented in Table 2.

The first experiment revealed that there was no one MOGA which demonstrated the highest effectiveness (in the sense of the IGD metric) for all of the test problems. The best results provided with NSGA-II, SPEA2, and PICEA-g separately are highlighted with in bold.

Table 2. 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**	196.060	0.107	42.327	0.010	236.677	0.068	56.566	**Outperforms the best value**
UF2	0.061	181.520	**0.060**	84.538	0.078	262.089	0.056	64.837	Corresponds to the **best** value
UF3	**0.191**	181.150	0.222	36.781	0.326	237.594	0.202	55.952	Corresponds to the **best** value
UF4	**0.055**	182.233	0.0570	75.837	0.083	243.208	0.058	60.271	Corresponds to the **best** value
UF5	**0.426**	181.509	0.498	33.844	0.518	240.198	0.338	56.391	**Outperforms the best value**
UF6	0.335	183.085	0.346	34.997	**0.319**	237.906	0.254	56.008	**Outperforms the best value**
UF7	**0.085**	181.039	0.091	75.556	0.125	245.891	0.084	60.269	**Outperforms** the best value
UF8	0.269	190.269	**0.191**	166.056	0.259	253.813	0.259	87.240	Corresponds to the **second** value
UF9	0.319	191.105	**0.290**	107.157	0.407	406.996	0.314	78.532	Corresponds to the **best** value
UF10	0.626	186.267	**0.421**	118.744	0.534	290.870	0.533	75.119	Corresponds to the **best** value

Then we compared these best IGD values obtained by MOGAs with the results of the cooperative algorithm. A t-test (with the significance level p=0.01) was used to expose the significant difference in the pairs of IGD values. As a result, it turned out that in seven cases there was no difference between the best results provided with MOGAs separately and the IGD values obtained with the cooperation of these MOGAs (in Table 2 'Corresponds to the best value' indicates these cases). Furthermore, the cooperative method outperformed the best MOGA twice (in Table 2 it is labeled 'Outperforms the best value') and only once its effectiveness corresponded to the second (in the sense of the IGD values) MOGA (this case is marked 'Corresponds to the second value'). This implies that our proposal is an effective alternative to the random choice of the appropriate MOGA for the problem considered.

Also the parallel implementation allows us to save computational time: the average number of seconds spent with conventional MOGAs (NSGA-II, SPEA2, and PICEA-g) is 176, whereas the cooperative algorithm requires 65 seconds. On average, it works faster than the fastest MOGA (PICEA-g) and much faster than two others. Certainly, these results depend on different characteristics of the computer used, but it might be roughly assessed that the computational time has been decreased essentially.

5 Conclusion

In this paper, we have proposed the multi-agent heuristic procedure to solve multi-objective optimization problems which does not require additional experiments to expose the most appropriate algorithm for the problem considered. This cooperative technique might be effectively used instead of any of its component. Moreover, the parallel work of island model components allows us to decrease the computational time significantly. For most of the test problems the effectiveness of the developed algorithmic scheme is comparable with (or even better than) the performance of its component which provides the best results separately.

The algorithm developed has already been applied to select informative features from data bases (two criteria were introduced – the Intra- and Inter-class distances). Also it has been successfully used to design neural network models taking into account two criteria (the computational complexity and the accuracy). All these applications will be presented in the next paper.

Thus, it might be concluded that due to advances in the algorithm proposed it might be effectively used in the *Machine Learning* field.

Acknowledgements. Research is performed with the financial support of the Ministry of Education and Science of the Russian Federation within the State Assignment for the Siberian State Aerospace University, project 2.1889.2014/K.

References

1. Khritonenko, D., Semenkin, E.: Application of artificial neural network ensembles for city ecology forecasting using air chemical composition information. In: Proceedings of the International Conference on Environment Engineering and Computer Application (ICEECA2014), Hong Kong, China, (2014) – In press
2. Stanovov, V., Semenkin, E.: Hybrid self-configuring evolutionary algorithm for automated design of fuzzy logic rule base. In: 11th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 317–321 (2014)
3. Semenkina, M., Semenkin, E.: Hybrid self-configuring evolutionary algorithm for automated design of fuzzy classifier. In: Tan, Y., Shi, Y., Coello, C.A. (eds.) ICSI 2014, Part I. LNCS, vol. 8794, pp. 310–317. Springer, Heidelberg (2014)

4. Freitas, A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Spinger-Verlag, Berlin (2002)
5. Whitley, D., Rana, S., Heckendorn, R.: Island model genetic algorithms and linearly separable problems. In: Corne, C., Shapiro, J.L. (eds.) Evolutionary Computing. LNCS, vol. 1305, pp. 109–125. Springer, Heidelberg (1997)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
7. Wang, R.: Preference-Inspired Co-evolutionary Algorithms. A thesis submitted in partial fulfillment for the degree of the Doctor of Philosophy, University of Sheffield, p. 231 (2013)
8. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems EUROGEN 2001 **3242**(103), 95–100 (2002)
9. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-wesley (1989)
10. Zitzler, E., Laumanns, M., Bleuler, S.: A tutorial on evolutionary multiobjective optimization. In: Gandibleux, X., (ed.) Metaheuristics for Multiobjective Optimisation. Lecture Notes in Economics and Mathematical Systems, vol. 535. Springer (2004)
11. Silverman, B.: Density estimation for statistics and data analysis. Chapman and Hall, London (1986)
12. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., Tiwari, S.: Multi-objective optimization test instances for the CEC 2009 special session and competition. University of Essex and Nanyang Technological University, Tech. Rep. CES-487 (2008)