ISSN 1407-7493 **DATORZINĀTNE** **COMPUTER SCIENCE** **2009-7493**

INFORMATION TECHNOLOGY AND MANAGEMENT SCIENCE
INFORMĀCIJAS TEHNOLOĢIJA UN VADĪBAS ZINĀTNE

HYBRIDISATION OF EVOLUTIONARY ALGORITHMS FOR SOLVING MULTI-OBJECTIVE SIMULATION OPTIMISATION PROBLEMS

EVOLŪCIJAS ALGORITMU HIBRIDIZĀCIJA DAUDZMĒRĶU IMITĀCIJAS MODELĒŠANĀ BĀZĒTĀS OPTIMIZĀCIJAS PROBLĒMU RISINĀŠANAI

Liana Napalkova holds her MSc degree in Computer Science from Riga Technical University, Latvia (2006).

Currently, she is a researcher and PhD student at RTU Department of Modelling and Simulation, and participates in research projects in the logistics field. Her interests focus on the use of simulation-based optimisation techniques for providing competitive advantages in multi-echelon supply chain cyclic planning.

Liana Napalkova is a member of the Society for Modeling and Simulation (SCS Student Chapter) and member of Latvian Simulation Society (LSS).

Keywords: multi-objective simulation optimisation, evolutionary algorithms, hybrid algorithms

Abstract – The paper presents a taxonomic analysis of existing hybrid multi-objective evolutionary algorithms aimed at solving multi-objective simulation optimisation problems. For that, the properties of evolutionary algorithms and the requirements made to solving the problem considered are determined. Finally, a combination of the properties, which allows one to increase the approximation accuracy of the Pareto-optimal front at relatively low computational costs, is revealed.

Introduction

Multi-objective optimisation problems are very common due to the multi-criteria nature of most real-world problems. Design of complex software systems, production scheduling, supply chain planning are such applications, where two or more, possibly conflicting and/or incommensurable objective functions have to be optimised simultaneously.

Traditional optimisation approach involves the use of analytical models represented in the form of algebraic, integral or differential relations. Although it allows investigators to find optimal solutions during a short time, they often over-simplify modelled systems and, thus, may produce infeasible solutions for a practice.

Nowadays, a promising way to solve real-world optimisation problems concerns the application of simulation optimisation approach [1]. In this case, simulation technique is used to capture all the complexities and dynamics of the modelled system, whereas optimisation techniques are aimed at finding optimal or near-optimal solutions. Thus, it is possible to estimate a system performance without a rigid structure of analytical model. This feature of simulation optimisation serves as the main advantage over traditional optimisation approach.

However, to be compatible with simulation optimisation approach, the algorithms should be able to systematically investigate a space of candidate solutions without requiring a “closed form” (i.e. analytical expression) of the problem in presence of simulation noise and high computational costs of experiments. Tabu search (TS), simulated annealing (SA) and evolutionary algorithms (EAs) are some representatives of this class of algorithms. They have been implemented in major simulation optimisation software, such as SimRunner® and OptQuest®. All of these algorithms require converting multi-objective problem into single-objective analogue. Then, the essential difficulty consists in determining a priori information concerning weight coefficients, thresholds and goal targets. Besides, the predefined thresholds and goal targets may be unobtainable under certain conditions. It should be also, mentioned that only a single solution of Pareto-optimal set can be found, which is not always the most preferable trade-off. To obtain more than one solution, multiple optimisation runs with different weight coefficients should be performed [2]. This trick, however, can be complicated by the non-convexity of Pareto-optimal front.

Alternatively, the multi-objective EAs apply the concepts of Pareto-optimality and dominance relation for searching Pareto-optimal solutions. They can evolve multiple parallel solutions instead of a single one that allows generating a set of non-dominated solutions at each iteration. On the other hand, EAs are able to preserve a diverse set of non-dominated solutions using specific mechanisms. In general, EAs are able to perform a search in a high dimensional space with different ranges for decision variables. Moreover, they have proved to be independent on strong problem structure, such as, for example, convexity and discontinuity of the objective function.

In spite of important advantages of multi-objective EAs, a simple combination with the simulation model may not provide efficient results because of high computational costs and the simulation noise, which influence the objective function estimates and performance of EA operators. Moreover, multi-objective EAs are often unable to simultaneously ensure good approximation and diversity of the Pareto-optimal front. The reason is that either search intensification or diversification is targeted. In such conditions, a perspective direction is to combine the desirable properties of EAs and other methods to mitigate their individual weaknesses, which is called the hybridisation [3].

Classifying the Hybrid Multi-Objective Evolutionary Algorithms

Over the last years, an interest in the hybridisation of multi-objective evolutionary algorithms has risen considerably among researchers. Although combinations of evolutionary algorithms and other methods have provided more powerful search algorithms, their efficiency in optimising outputs from simulation models has not yet been sufficiently investigated.

Therefore, the purpose of the hybridisation is to provide a solution to the following two issues:

1. How to increase the approximation accuracy of the Pareto-optimal front?
2. How to reduce total number of simulation optimisation iterations?

To accomplish this purpose, a taxonomic analysis of known hybrid multi-objective evolutionary algorithms is performed. The resulting classification tree is shown in Figure 1. It includes the following properties of the algorithms:

- cooperation strategy;
- cooperation level;
- search strategy;
- execution order;
- Pareto-optimal front generation type;
- mixture type.

The properties such as cooperation strategy, cooperation level are mixture type are recalled from existing taxonomies of single-objective metaheuristics [3,4], while others are proposed by the author to fit the multi-objective domain.

The cooperation strategy determines the way in which the algorithms are combined. It is possible to distinguish *collaborative* and *integrative* strategies. In the collaborative strategy, the algorithms work independently on each other by exchanging the information during the search process. On the contrary, the integrative strategy assumes that one algorithm is embedded in the other.

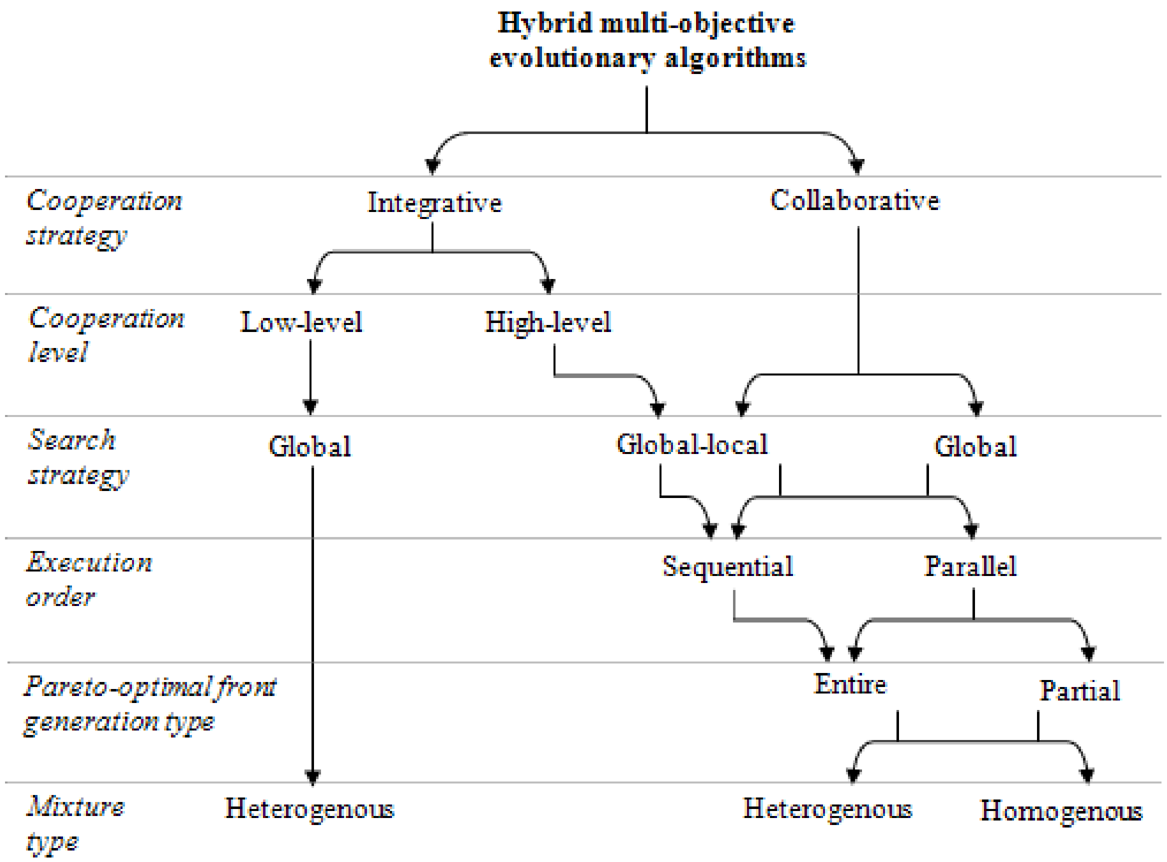

Fig. 1. Classification tree of hybrid multi-objective evolutionary algorithms

In particular in the integrative strategy, a further question is regarding the strength at which particular algorithms are combined. It is defined by the cooperation level that can be either can be either *low* or *high*. Low-level cooperation assumes that the original algorithms strongly depend on each other, because their individual components are exchanged or shared. In case of high-level cooperation, the main algorithm cooperates with embedded algorithms based on an interface, but for all that their individual identities are retained.

Another property is the search strategy of the Pareto-optimal front, which can be either *global* or *global-local* (mixed). The global search strategy implies that the algorithm is provided with mechanisms for escaping local optima of particular objective functions in the possible presence of multiple local optima. The mixed search strategy combines global and local search to, for example, more carefully investigate the neighbourhoods of non-dominated solutions and try to find their local improvements.

The execution order indicates a sequence in which the algorithms are applied. There are known *parallel* and *sequential* executions. In case of parallel execution, multiple algorithms simultaneously investigate the search space with the aim to provide better diversity of the Pareto-optimal front. To facilitate better results, non-dominated solutions found by each algorithm are shared, for instance, by placing them into general-use array. Sequential execution assumes that one algorithm is strictly performed after the other. Typically, the preceding algorithm performs a global search, whereas the succeeding algorithm is more focused on a local search. Although sequential executions based on the pure global search strategy is theoretically available, there is a lack of any examples in the literature.

From another point of view, it is possible to distinguish two types of Pareto-optimal front generations – entire and partial. In case of entire generation, the algorithms work on the whole search space trying to find the approximate Pareto-optimal solutions. Investigation of the entire search space of multi-modal objective functions implies the use of global search algorithms, whereas unimodal functions are compatible with pure local search algorithms. Alternatively, the front can be collected as a puzzle by decomposing the entire search space into sub-spaces and linking them to particular algorithms.

Finally, the last property that defines which algorithms are hybridised is the mixture type - heterogenous or homogenous. In heterogenous mixture, the multi-objective evolutionary algorithms are combined with the other methods, for instance, simulated annealing or tabu search. In homogenous mixture, only evolutionary algorithms are allowed.

An emphasis of the further analysis is made on searching a link between these properties and the above-mentioned issues.

A Unified View on the Hybrid Multi-Objective Evolutionary Algorithms

In the literature, several hybrid multi-objective evolutionary algorithms can be found.

The genetic algorithm running on the internet (GAIN) [5] and parallel single front genetic algorithm (PSFGA) [6] perform genetic search in parallel way. They both use the so called master-slave approach (see Figure 2) that is a variant of the parallel execution order. In this instance, the control of genetic operators is divided between computer processors. A single processor called as master takes control of the selection for mating, whereas the other processors called as slaves execute fitness evaluation, crossover and mutation operators independently. The communication between processors is usually set-up through the internet.

In order to speed-up the optimisation process, it is important to select a critical number of processors. In case of appropriate selection, these algorithms may outperform classical multi-objective EAs in terms of computational costs. However, if at least one of slaves has a lower processing speed than the others, then waiting for it may take a long time, which could negatively reflect on computational costs. Thus, because of the implementation complexity, there is a risk of getting worse results that it was previously expected.

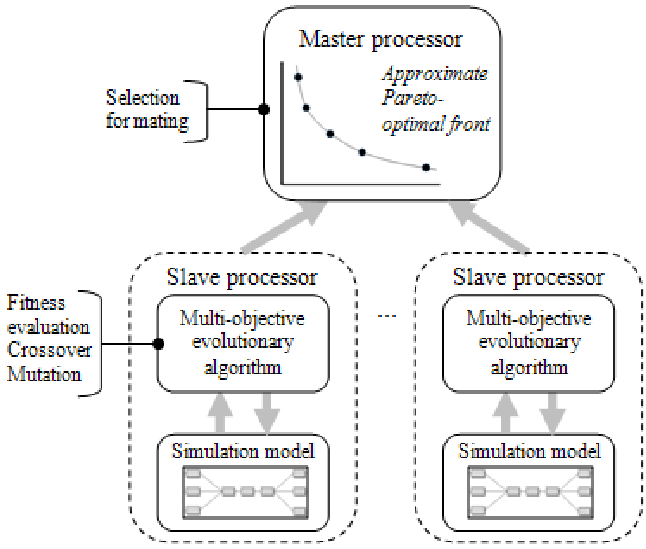

Fig. 2. Parallel execution order based on the master-slave approach

At the heart of the next three algorithms, divided range multi-objective genetic algorithm (DRMOGA) [7], parallel strength Pareto multi-objective evolutionary algorithm (PSPMEA) [8] and parallel multi-objective evolutionary algorithm with a hypergraph represented population structure (pMOHypEA) [9], lies the island approach (see Figure 3). This approach is based on a sort of the parallel execution order. In particular, a population of candidate solutions is divided into sub-populations called as islands, which are associated with particular objective functions or certain ranges of the Pareto-optimal front. In each island, multi-objective EA or another heuristic method is performed for several iterations, and non-dominated solutions are copied to the external archive. Periodically, each island randomly selects some solutions from its population and sends them to the neighbouring island defined by a ring topology. This process is called as migration. After migration, the worst solutions are replaced by the immigrants according to the ranking in each island, and the optimisation is continued. Finally, all Pareto-optimal solutions are collected from the islands to obtain the Pareto-optimal front.

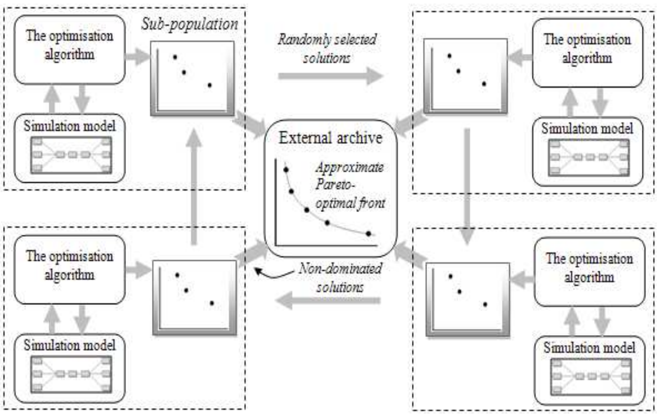

Fig. 3. Parallel execution order based on the island approach

The positive aspect of the island approach is that the population structure and hardware level architecture do not have to match. The major drawback is that the optimisation results are affected by multiple parameters, such as number of sub-populations, number and type of migrating solutions, migration frequency and topology. Inappropriate setting of at least one parameter can drastically deteriorate the quality of solutions.

The described approaches and related algorithms illustrate the entire (the master-slave approach) and partial (the island approach) generation of the Pareto-optimal front based on the collaborative cooperation of homogenous algorithms.

The memetic algorithms (MA) or genetic local search (GLS) algorithms, such as the simple multi-objective genetic local search (S-MOGLS) algorithm [10] and the memetic Pareto-archived evolution strategy (M-PAES) [11], are based on the idea of hybridising genetic operators of multi-objective EA with one of the local search methods. The role of the local search can be played by hill climbing, tabu search, simulated annealing, etc. Methods of this class apply the local search to every candidate solution generated by EA. Then, this improved solution is competing with the population for survival to the next population. The MA and GLS algorithms are representatives of the global-local search based on the high-level integrative cooperation and sequential execution order (see Figure 4).

On this way, Ishibuchi and Kaige [10] proposed the S-MOGLS algorithm, which probabilistically applies the local search to candidate solutions found by NSGA-II. In using the local search, multiple objective functions are aggregated by introducing weight coefficients that are randomly generated.

More sophisticated algorithm that doesn’t use the aggregation function is M-PAES. It consists of two phases, such as local search phase at which a classical PAES is applied and recombination phase. To maintain a finite set of non-dominated solutions found, a global archive is introduced, whereas a local archive is used as the comparison set in each of the local search phases.

All of these algorithms benefit from the overall global perspective of EAs and good convergence of local search methods to a local optimal solution. However, their essential drawback is that running the local search after each generation of EA is computationally expensive in case of solving MOSO problems.

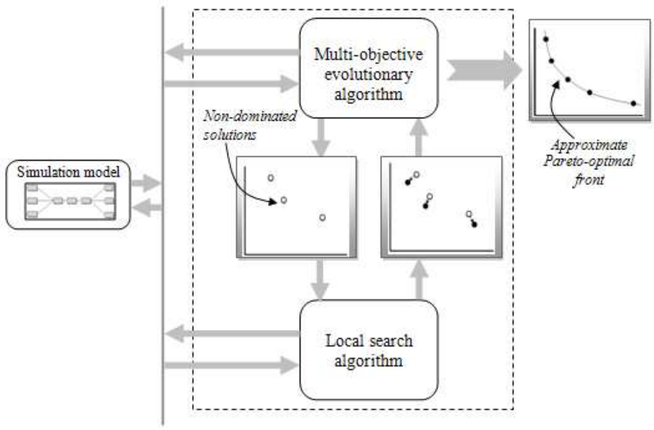

Fig. 4. High-level integrative cooperation with the sequential execution order

Global-local search based on the collaborative cooperation and the two-phase sequential execution order has been implemented in the hybrid EA developed by Talbi et al [12] and in the hybrid NSGA-II of Deb and Goel [13]. The main idea behind this algorithm is to use multi-objective EA for obtaining a diversified Pareto-optimal front at the first phase and to improve its approximation by the local search method at the second phase (see Figure 5). Therefore, such kind of algorithms is also called as two-phase.

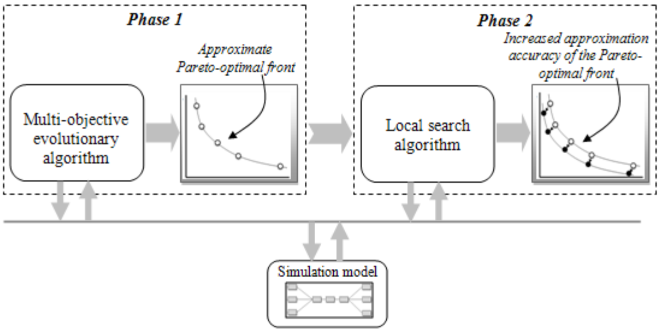

Fig. 5. Collaborative cooperation with the two-phase sequential execution order

The proposal of Talbi was to run EA for a fixed number of generations and then compute the neighbourhood for each Pareto-optimal solution by using simple local search method. Non-dominated solutions found are included in the Pareto-optimal front and their neighbourhoods are recomputed.

Deb and Goel [13] suggested combining NSGA-II and the local search method, where the later requires aggregating multiple objective functions into a weighted sum. At that the weights are computed for each solution based on its location in the Pareto-optimal front.

This class of algorithms is less computationally expensive than the previous one, because it applies local search only after completing the genetic search. In general, the efficiency of the sequential execution order based algorithms lies in fact that they don't require multiple processors and can be easily implemented on a single processor. Besides, they allow one to sufficiently improve the approximation of the Pareto-optimal front.

Recent developments in the investigated field concern low-level integrative cooperation of multi-objective EAs and techniques originating in artificial intelligence. Among such developments, one can mention the application of fuzzy logic rules to (i) the dynamical adjustment of the crossover and mutation rates in the NSGA-II algorithm [14], (ii) selection of more preferable solutions from the Pareto-optimal set based on their degrees of fuzzy optimality [15], (iii) incorporation of fuzzy ranking scheme, in which dominance degrees are measured by using membership functions [16]. It should be noticed that authors of these researches emphasise that the fuzzification is reasonable in case of more than three objectives.

A few attempts have been made to couple computational neuroscience with multi-objective EAs (see Figure 6). The most promising attempt consisted in replacing computationally expensive simulation models with artificial neural networks (ANNs). The ANNs play the role of so called metamodels that are used to accurately approximate the black-box objective functions over the range of interest.

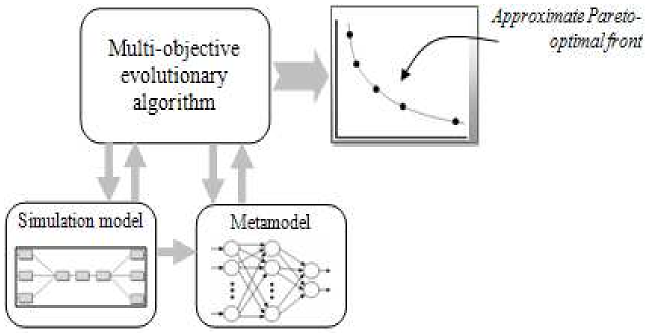

Fig. 6. Low-level integrative cooperation

On this way, Karakasis and Giannakoglou proposed a metamodel-assisted multi-objective evolutionary algorithm [16], wherein the fitness of chromosomes is pre-evaluated by using a radial basis function network in order to filter the poorly performing chromosomes and to direct only non-dominated ones to the exact evaluation. The role undertaken by the metamodels in multi-objective optimisation, where a front of non-dominated solutions is sought, is considerably harder than in single-objective optimisation. The reason lies in the fact that the population remains spread over a relatively extended area of the search domain throughout the evolution [196]. Moreover, metamodels require training and, for this purpose, an adequate set of already evaluated solutions must be available. Then the difficulty associated with ANN-based metamodels is that a large training data set (usually more than 100 data points) must be collected before its application.

This class of algorithms is utilised by commercial optimisation tools, such as SimRunner®, OptQuest®, which are add-on compatible with discrete-event simulation tools, such as Arena®, Crystal Ball®, ProModel®, SIMUL8®. Such tools automatically seek a course of actions to optimise performance of a simulated system by means of intelligent searching for an optimal solution. They use one or more common metaheuristics as its primary search method. For instance, SimRunner® combines genetic algorithm and evolution strategy, whereas OptQuest® relies on using tabu search and scatter search. In both tools, ANN-based metamodels are applied to substitute time-consuming simulation optimisation experiments in order to evaluate a model performance. The major drawback associated with commercial optimisation tools is that multiple objective functions must be aggregated into a single-objective function via assigning weighting coefficients to each objective function. As a result, multiple runs of simulation optimisation algorithm should be performed, which significantly increases the time needed for generating the Pareto-optimal front.

The analysis revealed that in condition of simulation optimisation collaborative cooperation outperforms the integrative cooperation as it has more appeal for providing lower computational costs. Although parallel execution order can ensure even lower computational costs than sequential one, this is paid by high implementation complexity and risk of inappropriate settings. Thus, the combination of the following properties may allow one to increase the approximation accuracy and diversity of the Pareto-optimal front at relatively low computational costs:

- collaborative cooperation of algorithms;
- global-local search strategy;
- sequential execution order;
- entire generation of the Pareto-optimal front;
- heterogenous mixture of algorithms.

Conclusions

The contribution of this paper can be summarised as the taxonomic analysis of the hybrid multi-objective evolutionary algorithms with the aim to reveal necessary features for solving multi-objective simulation optimisation problems. The achieved results should underlie the development of the methodology specialised on solving these problems.

References

1. Gosavi A. Simulation Optimisation: Parametric Optimisation Techniques and Reinforcement Learning. – Norvell: Kluwer Academic Publishers, 2003, 551 p.
2. Deb K. Evolutionary algorithms for multi-criterion optimisation in engineering design. Proceedings of Evolutionary Algorithms in Engineering and Computer Science. – London: John Wiley and Sons, Ltd., 1999. 135-161 p.
3. Jourdan L., Basseur M., Talbi E.G. Hybridizing exact methods and metaheuristics: A taxonomy. European Journal of Operational Research. – Vol. 199 (2009), 620-629 p.
4. Talbi E.G. A Taxonomy of Hybrid Metaheuristics. Journal of Heuristics. – Hingham: Kluwer Academic Publishers, 2002. 541-564.
5. Stanley T. and Mudge T. A parallel genetic algorithm for multiobjective microprocessor design. Proceedings of the Sixth International Conference on Genetic Algorithms. - San Francisco: Morgan Kaufmann Publishers, 1995. 597-607 p.
6. de Toro F., Ortega J., Ros E., Mota S., Paechter B., Martin J. PSFGA: Parallel processing and evolutionary computation for multiobjective optimisation. Parallel Computing. – Vol. 30, No. 6 (2004), 721-739 p.
7. Hiroyasu T., Miki M., Watanabe S. Divided range genetic algorithms in multiobjective optimization problems. Proceedings of International Workshop on Emergent Synthesis. – Kobe, 1999. 57-66 p.
8. Xiong S., Li F. Parallel strength Pareto multiobjective evolutionary algorithm. Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies. – Chengdu, 2003. 681-683 p.
9. Mehnen J., Michelitsch T., Schmitt K., Kohlen T. pMOHypEA: Parallel Evolutionary Multiobjective Optimization using Hypergraphs. Technical Report Reihe CI-189/04, SFB 531, ISSN 1433-3325, University of Dortmund, 2004.
10. Ishibuchi H., Kaige S. Implementation of Simple Multiobjective Memetic Algorithms and Its Application to Knapsack Problems. International Journal of Hybrid Intelligent Systems. - Vol. 1, No.1 (2004), 22-35 p.
11. Knowles J.D., Corne D.W. M-PAES: a memetic algorithm for multiobjective optimization. Proceedings of the 2000 Congress on Evolutionary Computation. - 2000. [Vol. 1], 325-332 p.
12. Talbi E.-G., Rahoual M., Mabed M.H., Dhaenens C. A Hybrid Evolutionary Approach for Multicriteria Optimization Problems: Application to the Flow Shop. Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science. - Springer-Verlag, 2001. 416-428 p.
13. Deb K., Goel T. Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science. - Springer-Verlag, 2001. 67-81 p.
14. Lau H.C.W., Chan T.M., Tsui W.T., Chan F.T.S., Ho G. T.S., Choy K L. A fuzzy guided multi-objective evolutionary algorithm model for solving transportation problem. Expert Systems with Applications. - Vol. 36, No. 4 (2009), 8255-8268 p.

15. Farina M., Amato P. Fuzzy optimality and evolutionary multiobjective optimization. Second International conference on evolutionary multi-criterion optimization, Lecture Notes in Computer Science. – Faro: Springer, 2003. [Vol.2632], 58–72 p.
16. Karakasis, M.K., Giannakoglou, K.C. Metamodel-Assisted Multi-Objective Evolutionary Optimization, EUROGEN 2005.

Liāna Napalkova. Evolūcijas algoritmu hibridizācija daudzmērķu imitācijas modelēšanā bāzētās optimizācijas problēmu risināšanai

Rakstā aplūkota evolūcijas algoritmu hibridizācijas problēma daudzmērķu imitācijas modelēšanā bāzēto optimizācijas uzdevumu risināšanai. Ir dota šo algoritmu klasifikācija ar mērķi noskaidrot raksturīgas īpašības, kas ļauj uzlabot konverģenci uz Pareto-optimālo frontu un samazināt kopējo iterāciju skaitu. Pamatojoties uz klasifikācijas rezultātiem, tiek piedāvāts izmantot divposmu secīgu optimizāciju, kas apvieno globālās un lokālās pārmeklēšanas stratēģijas, daudzmērķu imitācijas modelēšanā bāzētās optimizācijas metodoloģijas izstrādei.

Лиана Напалкова. Гибридизация эволюционных алгоритмов для решения задач многоцелевой оптимизации на основе имитации

Статья посвящена проблеме гибридизации эволюционных алгоритмов для решения задач многоцелевой оптимизации на основе имитационного моделирования. Приводится классификация данных алгоритмов с целью выявления характерных деталей, способствующих улучшению сходимости алгоритмов к Парето-оптимальному фронту и уменьшению вычислительных затрат связанных с имитицией. На основе результатов классификации предлагается использовать двухэтапную последовательную оптимизацию и объединение глобального и локального поиска для дальнейшей разработки методологии многоцелевой оптимизации на основе имитации.