Applied Soft Computing xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

New version of a multi-algorithm genetically adaptive for multiobjective optimization

Q1 Wali Khan [a,*], Abdel Salhi [b], M. Asif Jan [a], R. Adeeb Khanum [c]

[a] Department of Mathematics, Kohat University of Science & Technology, Pakistan
[b] Department of Mathematical Sciences, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
Q2 [c] Jinnah College for Women, University of Peshawar, Pakistan

A R T I C L E I N F O

Article history:
Received 8 March 2014
Received in revised form 13 May 2015
Accepted 24 August 2015
Available online xxx

Keywords:
Multi-objective optimization
Evolutionary algorithms
Pareto front
Multiobjective memetic algorithm
(MOMAs)

A B S T R A C T

Evolution is an indispensable process in life. Biologists have found this basic concept and scientists have modeled it on computers algorithmically known as evolutionary algorithms (EAs). Multiobjective EAs (MOEAs) are well established population based techniques for solving various search and optimization problems. MOEAs can employ different evolutionary operators to evolve population for approximating a set of optimal solutions in single run to problem at hand. Different evolutionary operators suite different problems. The use of multiple operators with self-adaptive manner can further improve the performance of existing MOEAs. This paper suggest new version of a multi-algorithm genetically adaptive for multi-objective (AMALGAM) by employing differential evolution (DE), particle swam optimization (PSO) and genetic algorithm (GA) for population evolution during the whole course of optimization. We examine the performance of new version of AMALGAM experimentally over two different test suites, the ZDT test problems and test instances designed recently for the special session of MOEAs competition in Congress of Evolutionary Computing 2009 (CEC'09). The suggested algorithm have provided better approximated results on most test problems in terms of inverted generational distance (IGD) as metric indicator.

1. Introduction

Q5 Multi-objective evolutionary optimization is a subject of intense interest in all fields of Sciences, Engineering, Economics, Logistics and others. Multiobjective optimization problems (MOPs) more than one conflicting objective functions and they have many real-world application [1,30]. A general MOP can mathematically be formulated as under:

$$\begin{aligned} &\text{minimize} \quad F(x) = (f_1(x), \ldots, f_m(x))^T \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{1}$$

where Ω is the decision variable space, $x = (x_1, x_2, \ldots, x_n)^T$ is an individual/solution and x_i, $i = 1, \ldots, n$ are their decision variables, $F(x) : \Omega \to R^m$ is consist of m real valued objective functions and R^m is called the objective space.

If Ω is closed and connected region in R^n and all the objective function in (1) are continuous of x, we called it continuous MOP. Furthermore, if $m \geq 3$, then problem (1) is said to be many objective problems. In single objective optimization, the main focus is on the decision space while in multi-objective optimization, we are mainly focus the objective space because objective values are used in defining the optimality [20].

In practical applications of optimization, it is very common, the objective functions of the MOP are in conflict with one another or mostly incommensurable. One needs a set of optimal solutions to solve these problems. Multiobjective evolutionary algorithms (MOEAs) are highly effective and powerful techniques and can find a set optimal solutions in single simulation run to MOPs due population based nature unlike traditional mathematical programming.

In the past two decade, since the inception of vector evaluated GA (VEGA) [24], different types of MOEAs have been suggested [5,32,31,7,21,9,2,3]. All they are mainly emphasizing on three conflicting goals: firstly, the final approximated Pareto front (PF) should be as close as possible to the true PF, secondly, final set of Pareto optimal solutions should be uniformly distributive and diverse over the true PF of the problem (1) and thirdly the approximated PF should capture the whole spectrum of the true PF. Different fitness assignment procedures, elitism or diversity promoting strategies are found in existing literature of evolutionary computing (EC). Pareto dominance concept MOEAs are

Q3 * Corresponding author. Tel.: +92 3469335809.
E-mail address: mashwanigr8@gmail.com (W. Khan).

http://dx.doi.org/10.1016/j.asoc.2015.08.059
1568-4946/

very common for solving MOPs [12,4]. To promote diversity in their population, most of these algorithms are utilizing different diversity techniques such as fitness sharing, niching approach, Kernel approach, nearest neighbor approach, histogram technique, crowding or clustering, relaxed form of dominance and restricted mating [3]. Among them, a fast non-dominated sorting algorithm (NSGA-II) [5], SPEA2: improving the strength Pareto evolutionary algorithm [31], the Pareto archive evolution strategy (PAES) [11], multiobjective genetic algorithm (MOGA) [8], and niched Pareto genetic algorithm (NPGA) [9] are long-familiar and well known approaches. They have shown good behaviors several comparative analysis.

Multiobjective memetic algorithms (MOMAs) is newly and attractive area of research in the existing literature of EC. They are algorithms inspired by the models of adaptation found in nature. This paradigm are habituating genetic algorithm globally in combination with various local search heuristics. They are also known as Baldwinian evolutionary algorithms (EA), Lamarckian EAs, cultural algorithms, or genetic local search and hybrid MOEAs [22]. Hybrid MOEAs develop with aim to overcome the shortcomings of stand-alone MOEAs [19].

A multi-algorithm genetically adaptive multi-objective (AMALGAM) is recently developed for solving both multiobjective optimization problems [25] and single optimization problems [26]. It employ multiple search operators for its population evolution. The search operators used including the particle swarm optimizer (PSO) [6], differential evolution (DE) [23] and NSGA-II [5] and allocates resources dynamically to each search operators based on their individual performances. It does not involve any decomposition as like MOEA/D: multiobjective evolutionary algorithm based on decomposition [27]. MOEA/D decomposes the approximated PF of the given MOP into a number of different single objective optimization subproblems and then optimize all these subproblems simultaneously by using generic evolutionary algorithm. MOEA/D [27] have performed very and have had many hybrid versions [10,13,15,14,18,16,19]. Our main objective is further improve the algorithmic performance of ALMAGAM by employing an alternative dynamic resource allocation scheme in its framework to tackle recently developed CEC'09 test instances [29] and commonly used ZDT test problems [33] in various comparative analysis.

The rest of this paper is organized as follows. Section 2 outlines the framework of new version of multi-algorithm genetically adaptive multi-objective (AMALGAM). Section 3 presents experimental results provided by new AMALGAM on both CEC'09 [29] and five ZDT test problems [33]. Section 4 devoted to discussion on experimental results. Section 5 finally concludes this paper with possible future plan.

2. New version of a multi-algorithm genetically adaptive for multiobjective optimization problems

Algorithm 1 outlines the framework of new version of AMALGAM. In Sept 1, a population P with size N is generated uniformly and randomly in the search space of the given MOP. We then evaluate the fitness values of each member of the population P together with their crowding distance calculation and categorize in different layers using fast non-dominating sorting technique of NSGA-II [5]. After this an Algorithm 1 utilizes k search operators to work on subpopulations N_1, N_2, N_3 in order to generate an offspring population Q of size N. We have used three search operators including DE, PSO, GA in the evolutionary process of our algorithm. Resources of each individual search operator are updated dynamically based on their individual performance according to the procedure as explained in Section 2.1.

Algorithm 1. New version of AMALGAM for MOPs.

Input: **1:** MOP: the multiobjective optimization problem;
2: N: the population size and other main parameters;
3: $\mathbf{F}_{eval}$: maximum function evaluations;
Output: $\{x^1, \ldots, x^N\}$ and $\{F(x^1), \ldots, F(x^N)\}$;
Step 1: Generate an initial population P of size N uniformly and randomly.
Step 2: Calculate the F-function values of each member of the P population.
Step 3: Assign rank to each member of P using fast non-dominating procedure.
Step 4: Assign sub-populations $P=\{P_1, P_2, \ldots, P_k\}$ to k operators for creating an offspring population $Q=\{Q_1, Q_2, \ldots, Q_k\}$ of size N.
Step 5: Calculate F-function values of Q offspring population.
Step 6: Assign rank to each member of Q using fast non-dominating procedure.
Step 7: Combine the new and old population P and Q, $R=P\cup Q$.
Step 8: Select population P of size N from population R of size $2N$ based on their ranks and crowding distances for next generation.
Step 9: Update N best individuals among C population with high ranks and crowding density.
Step 10: Update $P=\{P_1, P_2, \ldots, P_k\}$ (Explanation can be found in Section 2.1) based on the individual performances of each search operator.

2.1. Alternative adaptive resources allocation scheme

- We calculate the number of solutions that successfully enter to the next generation during the evolutionary process of new version of AMALGAM. A successful solution got awarded 1 and unsuccessful 0. A more successful operator gets more resources in the form of subpopulation to be operate on as compared to others.
- Let δ_1, δ_2, δ_3 are total number of non-dominated solutions produced by DE, PSO, GA which are successfully enter to next generation are convert into normalized form to develop probability formula (3)

$$P_k = \frac{\zeta_k}{\sum_{k=1}^{3}\zeta_k}, \quad \text{where } \zeta_k = \frac{\delta_k}{\sum_{k=1}^{3}\delta_k} \tag{2}$$

$$P_k = \alpha P_{k-1} \times N + (1-\alpha) P_k \times N \tag{3}$$

where P_k is the current and P_{k-1} is the previous probability of successes of the k search operators. More importantly, the above mentioned dynamic resources allocation did not switch on at every generation of proposed algorithm. It can switch at every multiple of 5th generation in ZDT test problems while for CEC'09 test instances at every multiple of 10th generation.

3. Parameters setting and experimental results

Experiments carried out on test functions with two and three objectives. Parameters for solving both ZDT test [33] and CEC'09 [29] are explained in Sections 3.1 and 3.2, respectively.

3.1. Parameter settings for ZDT problems

- $N=100$: population size for 2-objective test instances.
- $F=0.5$: scaling factor of the DE.
- $CR=0.5$: crossover probability for DE.
- w is the inertia factor which lies in [0.8, 1.2].
- c_1 and c_2 are the two acceleration constant or acceleration coefficients that usually lies between 1 and 4.
- $u_r \in [-1, 1]$ is a continuous uniform random number.
- $w = 0.5 + rand/2$: inertia factor which lies in [0.8, 1.2].
- $c_1 = c_2 = 1.5$: acceleration constant or acceleration coefficients that usually lies between 1 and 4.
- $\xi = 1$.
- $F_{eval} = 25,000$: maximum function evaluations.

Please cite this article in press as: W. Khan, et al., New version of a multi-algorithm genetically adaptive for multiobjective optimization, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.08.059