Preface

MOMO: Multi-Objective Metaheuristic Optimization

INTRODUCTION

Optimization techniques have played an important role in engineering and business domains where many complex problems we face in real-world are mathematically modeled as a parametric-form objective function and the optimal parameter setting for obtaining the best objective value is sought for. Recently, metaheuristic computing has shown significant competence against classic optimization methods. The first two volumes of the International Journal of Applied Metaheuristic Computing (IJAMC) have disclosed many theoretic breakthroughs and successful applications in metaheuristic computing for optimization with one single objective. We particularly solicit in future volumes of IJAMC more research articles addressing Multi-Objective Optimization Problems (MOOP) using metaheuristic computation.

Multi-Objective Metaheuristic Optimization (MOMO) is increasingly important due to two observations. First, many real-world problems involve multiple objectives that are conflicting with one another. Typical examples are cost-benefit analysis, engine performance and fuel consumption, weight of a mechanical part, and its strength, to name just a few. Secondly, literature findings have shown that MOMO is more beneficial than classic multi-objective optimization approaches. MOMO is able to find a set of non-dominated solutions in a single run, while classic multi-objective optimization approaches, such as weighted-sum, e-constraint programming, and compromise programming, need repetitive runs with various parameter settings.

Fred Glover (1986) first coined the term metaheuristic, which employs a master heuristic to guide the search course of a low-level heuristic to look beyond the local optimality. The master heuristic ranges from phylogenetic evolution, sociocognition, gestalt psychology, social insects foraging, to strategic level problem-solving rules. From a broader perspective, metaheuristic approaches can be classified as evolutionary algorithms (EA) and adaptive memory programming (AMP) techniques. The EA focuses on intelligent mechanisms inspired by natural metaphors, while the AMP relies on strategic level rules and memory manipulation. MOMO based on EA or AMP has received great attention from the MOOP community, but there are no or only few attempts that were devoted to MOMO using a hybrid framework of EA and AMP.

The previous volume of this series book has disclosed that researchers and practitioners intend to identify the primitive components contained in metaheuristic computing and to develop the so-called hybrid metaheuristics towards more effective metaheuristic computing (Yin, 2012). We particularly emphasize on the innovation named Cyber-heuristic, which combines useful notions found in EA and AMP and creates a more effective form of metaheuristic computing. This preface introduces two MOMO methods that illustrate the strengths of Cyber-heuristic.
The remainder of this preface is organized as follows. Section 2 reviews principal classic multi-objective optimization approaches. In Section 3 we introduce important features of MOEAs. Section 4 presents two notions for establishing a MOMO method that synergizes the strengths of EA and AMP. Finally, conclusions are made in Section 5.

CLASSIC MULTI-OBJECTIVE OPTIMIZATION

A widely accepted notion for multi-objective optimization is to search for the Pareto-optimal solutions which are not dominated by any other solution. A solution \( x \) dominates another solution \( y \) if \( x \) is strictly better than \( y \) in at least one objective and \( x \) is no worse than \( y \) in the others. Formally, given \( k \) minimization objective functions, \( f_i(x), \ i = 1, 2, \ldots, k \), solution \( x \) dominates solution \( y \), denoted as \( x \succ y \), if \( f_i(x) \leq f_i(y), \ \forall \ i = 1, 2, \ldots, k \) and \( f_j(x) < f_j(y), \ \exists \ j \in \{1, 2, \ldots, k\} \).

Figure 1 illustrates a simple case for a bi-objective optimization problem. Solution \( x \) dominates solution \( y \) because \( f_1(x) < f_1(y) \) and \( f_2(x) < f_2(y) \) (see Figure 1(a)). Therefore, those solutions whose plots locate in the dark region are dominated by solution \( x \). The plots of objective values for all Pareto-optimal (non-dominated) solutions form a Pareto front in the objective space as shown in Figure 1(b). We aim to find representative members of Pareto-optimal solutions whose plots are desirable to be equally distanced on the Pareto front.

Decision makers are clearly comfortable in seeking Pareto-optimal solutions because if the final solution is not Pareto-optimal it can be improved in at least one objective without deteriorating the solution quality in other objectives. However, it is sometimes difficult to find the true Pareto-optimal solutions due to the high complexity of the problem nature, and an approximate Pareto front is instead sought for. The quality of this front is measured in two aspects: (1) The convergence metric indicates how closely the approximate Pareto front is converging to the true Pareto front, and (2) the diversity metric is in favour of the approximate Pareto front whose plots are most evenly spread. The classic multi-objective optimization approaches include weighted-sum scheme (Chang et al., 2008; Ismail et al., 2011),

Figure 1. Dominance relationship and Pareto front in a bi-objective optimization case
goal programming (Schniederjans, 1995) and ε-constraint programming (Yokoyama, 1996). However, all of these approaches decompose the given multi-objective problem into a number of single-objective sub-problems, and multiple executions of the program are needed to obtain the final approximate Pareto front. In the following subsections, the major classic multi-objective optimization approaches are presented.

**Weighted-Sum Scheme**

The weighted-sum scheme transforms an MOOP into an SOOP (single-objective optimization problem). A linear aggregating function $\sum_{i=1}^{m} w_i f_i(x)$, where $w_i$ is the weight for the $i$th objective, is employed to combine the $m$ objectives as a weighted sum. Many executions with various settings of the weight values are needed to identify multiple non-dominated solutions on the front. Figure 2 shows an example. A given setting of weight values defines a tangent to the front, and the plot at the intersection is the optimal weighted sum corresponding to this weight setting.

The weighted-sum method, however, has the following drawbacks. (1) Multiple settings with evenly distributed weight values are needed to perform repetitive executions to locate representative plots on the front. (2) Some non-dominated solutions may have never been found using any setting of weights if the Pareto front is concave (as shown in Figure 2). (3) The weight value for each objective is difficult to determine, even the decision makers cannot precisely state and quantify the importance degree of each objective.

**ε-Constraint Programming**

The ε-constraint programming approach retains one objective and transforms the remaining objectives to constraints. Formally, the MOOP is converted to the following form:

$$\min_{X \in D} f_i(X)$$

subject to

$$f_j(X) < \varepsilon_j \quad j = 1, 2, ..., m; \quad j \neq i$$

*Figure 2. Weighted-sum scheme with various settings of weight values*
It can be seen in Figure 3 that $\varepsilon$-constraint programming can identify a point on the Pareto front by minimizing objective $f_2$ with respect to the constraint that $f_1 < \varepsilon_1$. With multiple executions of various constraint settings which gradually decrease the $\varepsilon$ value, multiple points on the Pareto front can be located.

**Compromise Programming**

The compromise programming technique uses the ideal point $y$ as a reference point to estimate how closely the objective plot approaches to the Pareto front. The ideal point is a pseudo plot whose individual objective value is obtained by deriving the optimal value for this single objective. The compromise programming intends to find the point that has the minimum value for the weighted-sum distance, 

$$\sum_{i=1}^{m} w_i |f_i(x) - f_i(y)|,$$

or the Chebyshev distance, 

$$\max_i \left( w_i |f_i(x) - f_i(y)| \right).$$

As shown in Figure 4, the objective plot for solution $x$ is optimal with respect to the weighted-sum distance with equal weights.

**Figure 3. Illustration of $\varepsilon$-constraint programming**

![Illustration of $\varepsilon$-constraint programming](image)

**Figure 4. Illustration of compromise programming**

![Illustration of compromise programming](image)
MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Nature-inspired metaheuristics employing various forms of natural metaphors are easy to describe and implement, and have been intensively investigated by the evolutionary algorithm (EA) community. Some of the prevailing EAs are evolutionary strategy, genetic algorithm (Holland, 1975), ant colony optimization (Dorigo, 1992), and particle swarm optimization (Kennedy & Eberhart, 1995). A great number of works have been conducted to develop Multi-Objective Evolutionary Algorithms (MOEAs) that extends EAs for SOOP to the context of MOOP. This section will review the features of MOEAs.

Recently, MOEA has been introduced as a viable technique to tackle multi-objective optimization problems. Two notable techniques, solution ranking and density estimation, were introduced to obtain high-quality convergence and diversity performances. The solution ranking technique gives each solution a score and is thus able to perform survival of the fittest to reach good convergence based on the rank of competing solutions. The density estimation technique measures the degree of crowding between the plot points in the objective space in order to guide the evolution with good diversity control. Another interesting notion for diversity control is via objective decomposition, which decomposes a multi-objective optimization problem into a number of sub-problems and optimizes them simultaneously. Each sub-problem is defined by a weighted-sum of objectives and the weight vector is well separated from the weight vectors for other sub-problems. Thus decomposition-based MOEA can generate a set of evenly distributed points on the front.

Solution Ranking

In order to maneuver the evolution, the quality of the solutions should be quantitatively measured and differentiated. One of the useful criteria is the dominance relationship among solutions. Here, we present the non-dominated sorting and Pareto strength broadly used in the literature.

The non-dominated sorting technique was originally proposed in NSGA (Deb et al., 2002). It gives the highest score to the solutions non-dominated by any other solutions in the population. The solutions with the highest score are then removed from the population, and the strategy proceeds to give the second-highest score to the non-dominated solutions in the population. The process is repeated until every solution in the population has been assigned a score.

The notion of Pareto strength was presented in SPEA (Zitzler et al., 2001). Each found solution is given a raw Pareto strength value defined as the ratio of the number of the remaining solutions that are dominated by this solution. However, this raw value is preferring points on the front central region because these points usually can dominate more points in the objective space than the points located near the end regions of the front. Therefore, the evaluation is conducted in a reverse way, i.e., the quality fitness of a solution is the sum of raw Pareto strength value for all the solutions that dominate this solution.

Density Estimation

The density estimation technique measures the crowding degree between the points found in the objective space in order to guide the evolution with good diversity control. SPEA2 uses the k-distance which is the distance to the k-th nearest neighbour to estimate the density. The k-distance is more reliable than the shortest distance which is easily biased by uneven point distributions. Some statistics on the k-distance value, such as the mean and the maximum values, can be derived for designing the diversity strategy.
As defined in NSGA-II, the crowding distance for a point is the average distance of two points on either side of this point along each of the objectives. The crowding distance is compromise between the k-distance and the shortest distance and does not require any density parameters. When the competing solutions have the same ranking based on dominance relationship, the solution with larger crowding distance is prevailing.

The grid-based technique has been used in the Pareto Archived Evolution Strategy (PAES) (Knowles & Corne, 2000), Fard et al. (2011) and the Multi-Objective Particle Swarm Optimization (MOPSO) (Coello Coello et al., 2004). The objective space is divided into regions called grids, and the number of points located in each grid cell is used as an estimate for the density. When there is a need for replacing a solution in the grid archive, a solution in the densest grid is selected at random for removal.

**Decomposition-Based MOEA**

Another interesting notion for diversity control is via objective decomposition. The MOEA/D algorithm (Zhang & Li, 2007) decomposes a multiobjective optimization problem into a number of sub-problems and optimizes them simultaneously. As shown in Figure 5, each sub-problem is defined by a weight vector $\lambda$ and the weighted-sum of objectives is considered as the global optimization goal of this sub-problem. The weight vectors are generated with uniformly distributed slopes and the weight vector of each sub-problem is well separated from the weight vectors for the other sub-problems.

Figure 6 shows the flow chart of the MOEA/D algorithm. The External Population (EP) is a memory space to store the non-dominated solutions that are found during the whole execution. At the initialization step, a population of solutions is randomly generated and each solution is designated to a weight vector that defines a particular sub-problem. The weight vectors are systematically generated such that their slop values are uniformly distributed. Each solution is seeking to optimize a sub-problem with the designated weight vector. For each solution, two neighbors of this solution are chosen randomly and

---

**Figure 5. Illustration of MOEA/D decomposition**

![Figure 5. Illustration of MOEA/D decomposition](image)
they are used to produce an offspring. Then, a local search heuristic is performed to improve the offspring. Finally, the offspring competes with the solution and its neighbors for the goal of the designated sub-problem. The three processing steps for all the sub-problems are iterated until a stopping criterion is reached. Thus, MOEA/D can generate a set of evenly distributed points on the front.

**AMP-BASED MOEA**

The adaptive memory programming (AMP) metaheuristic approaches takes a strategic level problem solving rules with memory manipulation. With varying forms of memory structure, purposeful strategies can be devised to enhance the balance between intensification and diversification types of search. Tabu search (Glover, 1989), scatter search (Laguna&Marti, 2003), GRASP (Feo&Resende, 1995) are notable strategic level problem solving metaheuristics among others. Few attempts for developing multi-objective versions of strategic level problem solving metaheuristics have been contemplated, such as MOTS (Kulturel-Konak et al., 2006) and AbYSS (Nebro et al., 2008). However, the distinct features of
adaptive memory and responsive strategy may be overlooked and there is a promising research area for seeking effective ways of combining AMP with EA to create a powerful MOMO algorithm.

As previously noted, the advantages of AMP have not been fully explored in assistance of MOEA. This section proposes two notions in this vein and presents preliminary results which are very promising. The first notion marries AMP with multi-objective particle swarm optimization (MOPSO), and the second notion uses AMP to enhance the performance of MOEA/D.

**AMP-Based MOPSO**

We previously introduced the Cyber Swarm Algorithm (Yin et al., 2010) which enhances PSO by emphasizing to generate new solutions using high-level problem-solving strategies. These strategies may involve complex neighborhood concepts, memory structure, and adaptive search principles, mainly drawn from the field of Tabu Search. The adjective “Cyber” indicates the connection between the nature-inspired PSO and the high-level problem-solving strategies provided by the Scatter Search/Path Relinking (SS/PR) suite (Glover, 1998). Omran (2011) edited a special issue on Scatter Search and Path Relinking methods which addresses the contributions of SS/PR template for swarm intelligence. Hence, we further propose an extension of Cyber Swarm Algorithm (CSA) for solving multi-objective optimization problems.

The proposed method, named MOCSA, adds new features to CSA for generating non-dominated solutions with good convergence and diversity performances. MOCSA consists of four memory components and responsive strategies. The swarm memory component is the working memory where the population of swarm particles evolves to improve their solution quality based on guided moving by reference to strategically selected solution guides. The individual memory reserves a separate space for each particle and stores the pseudo non-dominated solutions by reference to all the solutions found by the designated particle only. Note that the pseudo non-dominated solutions could be dominated by the solutions found by other particles, but we propose to store the pseudo non-dominated solutions because our preliminary results show that these solutions contain important diversity information along the individual search trajectory and they assist in finding influential solution guides that are otherwise overlooked by just using global non-dominated solutions. The global memory tallies the non-dominated solutions that are not dominated by any other solutions found by all the particles. The solutions stored in the global memory will be output as the approximate Pareto-optimal solutions as the program terminates. Finally, the reference memory selects the most influential solutions based on convergence and diversity measures. Moreover, to arouse the power of MOCSA when the search loses its efficacy, two responsive strategies are performed upon the detection of critical events which disclose the stagnation of the search power. It should be noted that the manipulation of memory in MOCSA is very different from that used in the original CSA. MOCSA determines the ranking of solutions by the dominance power and diversity relationship in the multi-objective space while the CSA considers the single-objective fitness and the diversity in the solution space. The selection of solution guides is also different in the two versions. In CSA, the best solution leader can be uniquely identified due to the single-objective context. In MOCSA, however, there exist multiple non-dominated solutions in each level of memory and alternative strategies may be applied.

The pseudo code of the MOCSA is summarized in Figure 7. In the initialization (Step 1) the initial values for particle positions, velocities, and experience memory are given. In the evolutionary iterations (Step 2), the guided moving, memory update, and responsive strategies are performed. In Step 2.1, each
Figure 7. Pseudo codes for the Multi-Objective CSA (MOCSA)

1 Initialization
   1.1 Randomly generate U particle solutions, \( P_{i} = \{p_{ij}\}, \) \( 0 \leq i < U, 0 \leq j < d \)
   1.2 Randomly generate U velocity vectors, \( V_{i} = \{v_{ij}\}, \) \( 1 \leq i \leq U, 0 \leq j < d \)
   1.3 Evaluate multiple fitness values for each particle. Update experience (individual, global, and reference) memory

2 Repeat until a stopping criterion is met
   2.1 For each particle \( P_{i}, i = 1, \ldots, U, \) Do
      
      2.1.1 Guided moving with selected solution guides:
      
      \[ v_{ij}^{m+1} \leftarrow K \left( v_{ij} + (\phi_{1} + \phi_{2} + \phi_{3}) \left( \frac{\omega_{1}\phi_{1}p_{best}_{ij} + \omega_{2}\phi_{2}g_{best}_{ij} + \omega_{3}\phi_{3}RefSol[m]}{\omega_{4}\phi_{4} + \omega_{5}\phi_{5} + \omega_{6}\phi_{6}} - P_{g} \right) \right) \]

      2.1.2 Update experience memory if necessary

      2.2 Convergence PR strategy: If the global memory has not been updated for \( t_{1} \) iterations, restart every particle by exploiting the region between the particle’s two closest gbest (say gbest\(_{j}\) and gbest\(_{k}\)) by

      \[ P_{i} \leftarrow \text{Convergence\_PR}(\text{gbest}_{j}, \text{gbest}_{k}), \; i = 1, \ldots, U \]

      Diversity PR strategy: Else if a particular particle’s individual memory has not been updated for \( t_{2} \) iterations, restart this particle by

      \[ P_{i} \leftarrow \text{Diversity\_PR}(\text{pbest}_{i}, P_{i}) \]

   2.3 Update experience memory if necessary

3 Output feasible non-dominated solutions in the global memory

The advantages of the MOCSA algorithm compared with previous MOEA algorithms for improving convergence and diversity performance are as follows. (1) MOCSA adopts multi-level memory to facilitate solution ranking and density estimation mechanisms. The experience updating is conducted in the order of swarm, individual, global and reference memory. The best experience obtained in a low level memory is used for updating of the next level memory. Thus, the multi-level memory realizes the solution ranking in guiding the evolution. Moreover, as each type of memory performs a different updating strategy which will preserve unique features of its maintained solutions, sustaining good diversity
Figure 8. Illustration of parental updating and within-section updating by using the weighted-sum scheme
in the whole population. (2) Reference set is an adaptive memory which stores the most influential solutions by reference to convergence and diversity estimates. This feature is very effective in guiding the revolution towards the true Pareto front. (3) The solution guides are selected according to competitions related to both performance measures. These solution guides are used systematically in combination with a reference solution selected in turn from the reference memory, imposing a dynamic social network for fostering a new particle. (4) Two responsive strategies triggered by critical events are particularly developed for improving the convergence and diversity performance. MOEA algorithms usually suffer the barrier of premature convergence. The responsive strategies employed in MOCSA are useful in detecting these critical events and redirect the search towards uncharted regions.

**AMP-Based MOEA/D**

This section intends to propose two useful notions from SS/PR domain to improve MOEA/D. Without loss of generality, in the remainder of the paper the decomposition technique for MOEA/D refers to the weighted-sum approach unless stated otherwise. To improve the convergence performance obtained by MOEA/D, we present the within-section updating by referring to the slope of the line connecting the origin and each population member. The produced solution will be used for updating of the population members of which the slope value is closest to that of this solution. Secondly, we present the cross-section update by density estimation for improving the diversity performance of the produced solutions.

**Within-Section Updating**

In standard MOEA/D, the produced solution will be used for updating of the neighboring solutions of its parents (see Figure 8(a)). However, our preliminary experiment showed that the produced solution may have better improving capability if the reference is not restricted to its parents. We contemplate that the slop of the line connecting the origin and the objective plot is a good reference to select appropriate candidate solutions for updating, and that the convergence performance of the final plot will be enhanced if the success rate of the within-section updating is higher (see Figure 8(b)).

The notion of within-section updating is also applicable to other decomposition techniques. For example, Figure 9 illustrates the improvement for MOEA/D by using the ε-constraint programming. The produced solution is used for updating of the previous solutions that are located in the same section of ε-constraint values. Our experimental result showed that the ε-constraint programming with the within-section updating technique is more effective than the weighted-sum scheme with the within-section updating technique.

**Cross-Section Updating**

The diversity control mechanism used in standard MOEA/D is solely based on weighted-sum decomposition. Although the solution updating is performed according to each weight vector in turn, the plots will be unevenly distributed on the front if the improvement success rate for each sub-problem varies significantly. We propose to use cross-section updating by path-relinking technique to enhance the diversity degree of the plots.

The cross-section updating proceeds as follows. The objective space is divided to a number of equal-size sections. We estimate the section density by the number of plots locating in this section zone. The
secion zone $i$ is bounded by the two radii with slop values $\sigma_i$ and $\sigma_{i+1}$. The section with the lowest density is targeted for cross-section updating for improving the front diversity. The path-relinking is performed by taking an external-archived solution from each of the two neighboring sections of the lowest-density section. One solution is referred to as the initiating solution, the other solution is used as the guiding solution. Path-relinking undertakes to explore the trajectory space by constructing a path that transforms the initiating solution into the guiding solution by generating a succession of moves that introduce attributes from the guiding solution into the initiating solution (see Figure 10(a)). The cross-section updating technique can be also applied to improve the diversity performance of $\varepsilon$-constaint programming as shown in Figure 10(b), but the section is referring to the zone within two consecutive $\varepsilon$-constaint values. Our preliminary experiment showed that path-relinking technique can identify new non-dominated solutions in the lowest-density section and the cross-section updating is useful in improving the diversity performance of the final solution front.

CONCLUSION

Many real-world problems involve multiple optimization objectives that are conflicting to one another. A compromise to seek a trade-off among these objectives is to find a set of non-dominated solutions that define a Pareto front. However, it is usually too computationally prohibitive to find these non-dominated solutions. Instead, the approximate Pareto front is sought for and the quality of this front is measured in terms of convergence and diversity. The convergence performance metric indicates how closely the approximate Pareto front is converging to the true Pareto front. The diversity performance metric is in favour of the approximate Pareto front whose plots are most evenly spread.
Figure 10. Illustration of cross-section updating
This chapter reviews the major classic multi-objective optimization approaches including the weighted-sum scheme, $\varepsilon$-constraint programming, and compromise programming. The salient features of Multi-Objective Evolutionary Algorithms (MOEAs) are then presented. Finally, we propose two novel Adaptive Memory Programming (AMP)-based MOEA methods. The first method named MOCSA combines PSO with SS and creates more benefits that are not obtainable by the previous version. The second method improves MOEA/D by using two notions. The within-section updating drives the search towards the true Pareto front more effectively than the parental-relationship updating that is employed by the original MOEA/D. The cross-section updating technique detects the section with the lowest density value, and applies path-relinking procedure to construct a path between its two neighboring sections. The cross-section updating is able to produce more non-dominated solutions in the lowest-density section, and the diversity degree of the solution front is thus improved. The chapter has shown the great potentials of AMP for improving MOEA methods and we invite more future studies aiming to this promising research domain.

Peng-Yeng Yin
National Chi Nan University, Taiwan

REFERENCES


