3

MOEA Local Search and Coevolution

> When two opposite points of view are expressed with equal intensity, the truth does not necessarily lie exactly halfway between them. It is possible for one side to be simply wrong.
>
> Richard Dawkins

3.1 Introduction

In order to make multiobjective evolutionary algorithms (MOEAs) more beneficial to real-world applications, local search structures have been proposed to drive the search towards the Pareto front more effectively and efficiently. A number of generic local search techniques have been proposed along with problem domain specific methods. These approaches are discussed in this chapter with thoughts on integrating new innovative local search with MOEAs. Another emerging area of MOEA research is applying coevolutionary techniques. Relatively few researchers have explored the idea of combining coevolution with MOEAs. This chapter presents various researchers' algorithmic processes for Coevolutionary MOEAs (CMOEA) with each researcher's efforts summarized, categorized, and analyzed. Some potential concept and future applications of MOEA coevolution are also suggested. Exercises, discussion questions, and possible research directions for MOEA local search and coevolution are presented at the end of the chapter.

3.2 MOEA Local Search Techniques

It turns out that in many multiobjective optimization problems (MOPs), points in PF_{known} are clustered in various regions of objective space. Thus, it may be possible to computationally direct points in such regions as well as isolated points closer to PF_{true} using clever mechanisms that exploit certain

properties of the search space. For example, using one or a few of the objectives, it may be possible to adopt a **local search** technique to move a point closer to PF_{true} (i.e., better approximate the Pareto front with PF_{known}). And in addition, based upon MOEA goals, the use of local search (LS) may generate a better distribution of points on PF_{known}. Of course, the LS process starts in decision space or solution space with points in this space mapping to objective space.

Specific local search decision space approaches for consideration would be depth-first search (hill-climbing) [1407], simulated annealing [861], and Tabu search [572]. Since we are combining (hybridizing) global search MOEAs with local search techniques, they are generally defined as *hybrid* or *memetic* MOEAs.[1]

Algorithm 15 represents a generic memetic MOEA with the inclusion of the local search (LS) process noted. The specific position of the LS within a standard MOEA cycle can vary depending upon design (i.e., conducting a LS at every generation or at the end of a certain number of epochs). In general, LS techniques employ decision space neighborhoods whose selected points generate vectors in the objective space (phenotype). Note that doing local search in the phenotype domain is impractical since mapping from nonlinear objective functions back to unique decision variable values is generally impossible.

Balancing global MOEA search with local search for specific MOPs is critical to achieving good results. If fitness function computation in real-world MOPs takes a considerable amount of CPU time, there exist computational tradeoffs between local and global search. Thus, in the design and implementation of a MOEA-LS,[2] specific questions arise relating to LS effectiveness and efficiency:

- How often should the LS be applied based upon a probability, P_{LS}?
- On which k solutions should LS be used given a neighborhood $N(\mathbf{x})$ where $\mathbf{x}$ is a current solution?
- How long should LS be run defined by a time period T?
- How efficient does LS need to be versus effectiveness?

[1] Pablo Moscato [1133] introduced the concept of "memetic algorithm" to denote the use of local search heuristics with a population-based strategy. The term "memetic" has its roots in the word "meme", which was first introduced by Richard Dawkins in his classical book "The Selfish Gene" [340]. Dawkins defines a meme as the "unit of imitation" in cultural transmission. Therefore, a **memetic algorithm** can be seen as an approach that tries to mimic cultural evolution rather than biological evolution (like evolutionary algorithms). The main difference has to do with the way in which information is transmitted. Whereas genes are passed intact, memes are typically adapted by the individual who transmits them. For more information on memetic algorithms, see [661].

[2] The terms multiobjective memetic algorithm and MOEA-LS, are used interchangeably in this chapter.