Chapter 3.6

DIMMA: A Design and Implementation Methodology for Metaheuristic Algorithms – A Perspective from Software Development

Masoud Yaghini
Iran University of Science and Technology, Iran

Mohammad Rahim Akhavan Kazemzadeh
Iran University of Science and Technology, Iran

ABSTRACT

Metaheuristic algorithms will gain more and more popularity in the future as optimization problems are increasing in size and complexity. In order to record experiences and allow project to be replicated, a standard process as a methodology for designing and implementing metaheuristic algorithms is necessary. To the best of the authors' knowledge, no methodology has been proposed in literature for this purpose. This paper presents a Design and Implementation Methodology for Metaheuristic Algorithms, named DIMMA. The proposed methodology consists of three main phases and each phase has several steps in which activities that must be carried out are clearly defined in this paper. In addition, design and implementation of tabu search metaheuristic for travelling salesman problem is done as a case study to illustrate applicability of DIMMA.

DOI: 10.4018/978-1-61350-456-7.ch3.6

1. INTRODUCTION

Optimization problems, which occur in real world applications, are sometimes NP-hard. In the case of NP-hard problems, exact algorithms need, in the worst case, exponential time to find the optimum. Metaheuristics or *modern heuristics* deal with these problems by introducing systematic rules to escape from local optima. Metaheuristics are applicable to a wide range of optimization problems (Doreo et al., 2006; Morago, DePuy, & Whitehouse, 2006). Some popular population-based metaheuristic methods are genetic algorithm (Goldberg, 1989) and ant colony optimization (Dorigo & Stützle, 2004) in which collective intelligence play the important role (Wang, 2010). Tabu search (Glover & Laguna, 1997) and simulated annealing (Kirkpatrick, Gelatt, & Vecchi, 1983) are the two popular single-solution based metaheuristics that improve a single solution in an iterative algorithm.

With growing scale and complexity of optimization problems, metaheuristics will gain more and more popular. According to significant growth in using metaheuristics as optimization tools, there must be a standard methodology for design and implementing them. Such a methodology is used for recording experience and allows projects to be replicated. Moreover, this standard methodology can be a comfort factor for new adopters with little metaheuristic experience, and can show the guidelines to everyone who want to design and implement metaheuristics.

To the best of our knowledge, no methodology has been proposed in literature for design and implementation metaheuristic algorithms. There are many software frameworks in the literature for metaheuristics (Voss & Woodruff, 2002; Fink et al., 1999), in which framework means reusable programming codes and components for metaheuristics (Talbi, 2009). Hence, the meaning of frameworks in these references is different from our proposed methodology. Although there are several tutorials as lectures on how to design meheuristics (Thierens, 2008), they are sometimes for special metaheuristic and do not consider this process as a whole.

The proposed methodology in this paper, a *Design and Implementation Methodology for Metaheuristic Algorithms (DIMMA)*, shows guidelines to everyone who wants to design and implement a metaheuristic algorithm. Webster's collegiate dictionary defines methodology as "a body of methods, rules, and postulate employed by a discipline" or "the analysis of the principles or procedures of inquiry in a particular field" (Merriam-Webster, 1997).

DIMMA includes several phases, steps, disciplines, and principles to design and implement a specific metaheuristic for a given optimization problem. In other words, DIMMA refers to the methodology that is used to standardize process of design and implementing a metaheuristic algorithm. In Sections 2-5 we explain the architecture of DIMMA and its phases and steps, In Section 6 we followed by a description of each step of DIMMA using design and implementation of Tabu Search (TS) metaheuristic for Travelling Salesman Problem (TSP) as a case study.

2. ARCHITECTURE OF DIMMA

The architecture of DIMMA has been inspired from *Rational Unified Process* (RUP) which is a methodology for software engineering (Kroll & Krutchten, 2003). DIMMA has two dimensions including *dynamic* and *discipline* dimension (Figure 1). Dynamic dimension is the horizontal dimension, which includes phases of the methodology: *initiation*, *blueprint*, and *construction*. Discipline dimension is the vertical dimension that shows the disciplines, which logically group the steps, activities, and artifacts.

DIMMA has three sequential phases that each of them has several steps (Figure 2). In each step, we define several activities, which must be done to complete the steps. These phases are as follows:

Related Content