Chapter 8
Applications in Operations Research

This chapter is devoted to three representative applications of PSO in operations research. Similarly to the previous chapters, our attention is focused on the presentation of essential aspects rather than reviewing the existing literature. Thus, we present methodologies for formulation of the optimization problem, which is not always trivial, as well as for the efficient treatment of special problem requirements that cannot be handled directly by PSO. Under this prism, we report applications from the fields of scheduling, inventory optimization and game theory. Recent results are also reported per case to provide an idea of the efficiency of PSO.

INTRODUCTION

Operations research (or *operational research*) (OR) is a scientific field that deals with the detection of optimal solutions in complex problems. Optimization itself constitutes a branch of OR, probably the most important one, since the majority of OR problems end up as optimization tasks. However, this is not the only interesting branch of OR. Probability theory, queuing systems, game theory, graph theory, simulation and management sciences, are also intimately related or constitute branches of OR. The procedure of solving an OR problem requires both the selection and application of the most appropriate algorithm for the specific task. Thus, the user interferes crucially in different aspects of the procedure, ranging from solution representation to constraints handling (Hillier & Lieberman, 2005; Winston, 2003).

Algorithms with the plasticity, adaptability and efficiency of evolutionary and swarm intelligence approaches can be more than useful in applications with the aforementioned characteristics. For this reason, OR has always constituted a prosperous field of applications for evolutionary computation and swarm intelligence methods, usually providing problems characterized by peculiarities such as mixed

DOI: 10.4018/978-1-61520-666-7.ch008

variable types, varying dimensionality and computationally expensive function evaluations that are obtained through simulation procedures (Hillier & Lieberman, 2005; Winston, 2003).

PSO has been increasingly used in OR applications in the past few years. We present three such applications, from the fields of scheduling, management sciences and game theory, to illustrate the workings of PSO and offer a taster of its performance. For each application, elements that require user interference are analyzed along with the necessary techniques that render PSO applicable to such problems.

SCHEDULING PROBLEMS

In general, *scheduling* refers to the allocation of resources to tasks. This problem type is met very often in real world applications and it has proved to be NP-hard (Johnson & Garey, 1979; Lenstra *et al.*, 1977; Pinedo, 1995). The main objective in scheduling is the assignment of jobs (tasks) to a single or many machines so that several operational criteria are met. These criteria are usually modeled as the minimization of one or more objective functions.

The *single machine total weighted tardiness* (SMTWT) problem is a very challenging scheduling task, with many applications (Pinedo, 1995). Its objective is the sequential processing of n jobs on a single machine, so that its total tardiness is minimized. To put it formally, let $j = 1, 2,\ldots, n$, denote a job; p_j be its processing time; d_j be its due date; and w_j be a weighting factor. A *job sequence* is an n-dimensional ordered vector, $s = (s_1, s_2,\ldots, s_n)^T$, where s_j is an integer denoting the processing order of the j-th job, $j = 1, 2,\ldots, n$. For example, $s_1 = 3$, denotes that job 1 is scheduled as the third job in the processing order, while, $s_2 = 1$ denotes that job 2 is scheduled as the first job in the processing order. All jobs are assumed to be available for processing at time zero.

Let C_j be the completion time of job j in a job sequence s. Since s constitutes a permutation of the vector $(1, 2,\ldots, n)^T$, and time counting starts from zero, C_j can be computed as:

$$C_j = \sum_{\substack{\text{All } i \text{ such} \\ \text{that } s_i \leq s_j}} p_i .$$

Then, the *tardiness*, T_j, of job j is defined as:

$$T_j = \max\{0, C_j - d_j\}.$$

The objective of the SMTWT problem is the determination of a job sequence that minimizes the *total weighted tardiness*, which is defined as:

$$T = \sum_{j=1}^{n} w_j T_j . \tag{1}$$

Branch-and-bound algorithms constitute a standard methodology for tackling SMTWT problems. Unfortunately, they fail to solve large-scale instances of the problem (Abdul-Razaq *et al.*, 1990). In such cases, specially designed heuristic approaches, such as the *earliest due date* and *apparent urgency*, as well as more common optimization algorithms such as simulated annealing, tabu search, genetic algorithms and