Available online at www.sciencedirect.com
ScienceDirect
Int. J. Production Economics 113 (2008) 359–369
international journal of production economics
www.elsevier.com/locate/ijpe

Inventory replenishment with interdependent ordering costs: An evolutionary algorithm solution

Anne L. Olsen*

Department of Computer Science and Quantitative Methods, College of Business Administration, Winthrop University, 322 Thurmond Building, Rock Hill, SC 29733, USA

Received 1 September 2006; accepted 3 September 2007
Available online 6 October 2007

Abstract

Interdependence of minor ordering costs occurs in joint replenishment of inventory when the cost of placing an item in an order depends on which other items are also in the order. In this paper, we propose an evolutionary algorithm (EA) for joint replenishment of inventory which allows for the interdependence of minor ordering costs. Since solutions to the joint replenishment problem (JRP) can be represented by integer decision variables, this makes the JRP a good candidate for an EA. The results of testing 2430 randomly generated problems show that our algorithm provides close to optimal results for some problems.

Keywords: Joint replenishment; Evolutionary algorithm; Interdependence; Minor ordering costs

1. Introduction

The problem of determining an inventory replenishment policy where multiple items are ordered from a single supplier is known as the joint replenishment problem (JRP). The JRP takes into account the cost of holding items in inventory, the fixed cost of preparing an order (major ordering cost), a variable cost associated with each item in the order (minor ordering cost), and constant demand for items. The savings realized by joint replenishment can be significant due to savings associated with the major ordering cost.

The JRP is typically formulated as follows: determine an inventory replenishment policy that minimizes the total cost (TC) of replenishing multiple items from a single supplier. The problem assumes that the minor ordering costs of all items included in an order are independent of each other. This means that the minor ordering cost of an item is not affected by other items included in the same order. In practical applications, this assumption does not always hold since minor ordering costs may depend on which other items are included in the order. In 2003 Olsen outlined an evolutionary algorithm (EA) to solve the problem of inventory replenishment under interdependent ordering costs for 10 items. Our algorithm is based on that work. We consider two types of dependencies: when the minor ordering cost associated with an item increases when it is placed in the same order as some other item(s) and when items are constrained from being placed in the same order.

*Tel.: +1 803 323 2597; fax: +1 803 323 3960.
E-mail address: olsena@winthrop.edu

0925-5273/$ - see front matter
doi:10.1016/j.ijpe.2007.09.004

We begin with examples of conditions under which the assumption of independence of minor ordering costs fails. Consider the case of ordering food items. When a store needs to replenish both canned goods and refrigerated items, these items are usually shipped separately. If the canned goods were shipped with the refrigerated items in the same truck, the minor ordering cost of the canned goods would increase to reflect the higher cost of shipping under refrigeration. We call this added cost a penalty. Other examples exist in which certain items may be prohibited or constrained from being included in the same order. This is the case with lithium batteries which may not be shipped by air and chemical products which must be isolated from food products. In both examples, costs associated with joint replenishment depend on which items are included in the same order.

The problem of finding a joint replenishment policy under conditions of interdependent minor ordering costs requires that all possible groupings of items be investigated. Since there are 2^{n-1} ways to place n items into two groups, the time it takes to find an optimal solution grows exponentially with n. The problem is actually much more difficult since all ways to group items into three groups, four groups, etc. must be considered. This means that for large n it may take years, even for modern computers, to find a solution. Algorithms based on evolutionary computation are useful for searching large solution spaces and finding good solutions to this type of problem (Eiben and Smith, 2003; Michalewicz, 1992). These evolutionary methods include genetic algorithms (GAs), evolution strategies, EAs, and evolutionary programs. In this paper, we propose an EA to find a near-optimal solution to the JRP under the assumption of interdependent minor ordering costs. All other assumptions of the JRP remain as stated in the classic problem. Since EAs work very well for large problems where enumeration is not feasible, we expect that an EA should also work well for the JRP under conditions of interdependence of minor ordering costs. We continue in Section 2 with a brief review of the JRP. In Section 3 we present a summary of the evolutionary computational approach to finding solutions to problems and propose an EA (EA_dep) to find a solution to the JRP under conditions of interdependence. Our testing methods and results are described in Section 4. Section 5 concludes with discussion and suggestions for future research.

2. The problem and related literature

2.1. Joint replenishment under independence

The goal of the JRP is to minimize the TC of inventory replenishment. The notations used are:

n—number of items ordered from a single supplier,
i—item number, $1 \leqslant i \leqslant n$,
D_i—demand rate for item i, in units/year,
h_i—holding (or inventory) cost for item i, in dollars per unit per year,
S—fixed cost of ordering, in dollars,
s_i—variable cost if ordering item i, in dollars,
T—basic cycle time, and
TC—total cost per year where TC is the sum of the total ordering costs and the total holding costs for the year.

The joint replenishment strategy involves coordinating orders so that the fixed (major) cost of ordering is shared by multiple items. The problem is to find an optimal (or close to optimal) replenishment policy. This requires determining how items will be grouped into single orders.

There are two methods covered in the literature to find these groupings, indirect grouping and direct grouping. The indirect grouping method uses a basic cycle time, T, with a replenishment order placed every cycle. Not all items are necessarily ordered every cycle. Thus, for every item, i, an integer, k_i, needs to be computed with the following meaning: if $k_i = 1$, the item is ordered every cycle, if $k_i = 2$, the item is ordered every other cycle, etc. To determine T, the k_i's must be known and vice versa. Early solutions to this problem include those published by Shu (1971), Goyal (1973, 1974), Silver (1976), Goyal and Belton (1979), and Kaspi and Rosenblatt (1983).

Kaspi and Rosenblatt (1991) proposed an algorithm called RAND which improves their 1983 algorithm by determining minimum and maximum values for T, $T_{\min}$, and $T_{\max}$. A set of initial values of T are then determined from this range. For each of these initial values of T, the 1983 algorithm is used to find values for the k_i's. An iterative process improves the values of the k_i's until they converge. The TC is calculated by

$$\mathrm{TC} = \left[2\left(S + \sum_i (s_i/k_i)\right)\left(\sum_i h_i k_i D_i\right)\right]^{1/2}. \qquad (1)$$